An updated version of this publication is available at
http://urn.fi/URN:ISBN:978-952-302-702-2

EHES Manual

PART C.
EUROPEAN LEVEL COLLABORATION

Hanna Tolonen (editor)

Data sharing rules

Evaluation

Quality assurance

Indicators

Reporting

Data management

European level coordination

Directions 2013_003

Edited by

Hanna Tolonen

EHES MANUAL

PART C.
European level collaboration

Helsinki 2013

The EHES Pilot Project received funding from the European Commission/DG Sanco. The views experessed here are those of the authors and they do not represent the Commission's official position.

Publisher National Institute for Health and Welfare

PO Box 30, FI-00271 Helsinki, FINLAND

http://www.thl.fi

NATIONAL INSTITUTE
FOR HEALTH AND WELFARE

ISBN (pdf) 978-952-245-846-9

ISSN (pdf) 2323-4172

http://urn.fi/URN:ISBN:978-952-245-846-9

Helsinki, Finland 2013

Reference:

Tolonen H (Ed.) EHES Manual. Part C. European level collaboratoin. National Institute for Health and Welfare. 2013. Directions 2013_003. URN:ISBN:978-952-245-846-9, URL: http://urn.fi/URN:ISBN:978-952-245-846-9

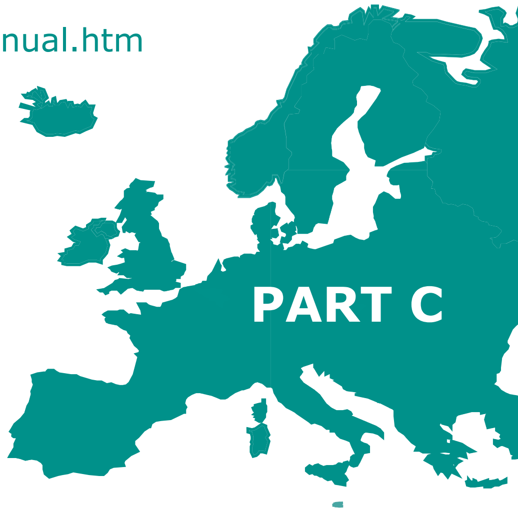
Layout: Hanna Tolonen

Cover: Hanna Tolonen

Graphics:

Chapter 2: Ari Haukijärvi

Contents



Introduction

The European Health Examination Survey (EHES) Manual provides guidelines and specifies the requirements for the implementation of standardized national health examination surveys (HES) in the European countries. Recommendations based on past experiences from national and international surveys were prepared by the Feasibility of a European Health examination Survey (FEHES) Project (Tolonen 2008). The EHES manual builds on these recommendations and on further experience obtained during the EHES Pilot Project in 2009-2012. The EHES Manual has three parts:

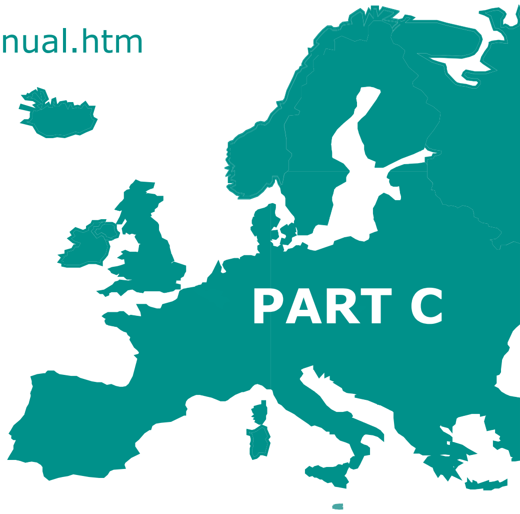
A. Planning and preparation of the survey
B. Fieldwork procedures
C. European level coordination

The EHES Manual is maintained by the EHES Reference Centre, and the plan is to update it with further clarifications and additional relevant topics. The latest version of the EHES Manual is available in the Internet at www.ehes.info.

This is Part C of the EHES Manual. It considers the principles for European level sharing of the EHES data, the procedures for European level evaluation, analysis and reporting of the national HESs, and the structure for the European level coordination of EHES.

References

- Tolonen H, Koponen P, Aromaa A, et al. (Eds.) Recommendations for the Health Examination Surveys in Europe. B21/2008, Publications of the National Public Health Institute, Helsinki 2008. Available at http://urn.fi/URN:ISBN:978-951-740-838-7. Accessed on 22 January 2013



1. Data sharing rules

Kari Kuulasmaa[1]

[1] National Institute for Health and Welfare (THL), Helsinki, Finland

The principles and rules for sharing and use of EHES data will be defined separately for the different organizational settings and funding contracts of EHES. However, whenever data are transferred within the EHES framework, the conditions specified the respective Data Transfer Agreement (DTA) need to be followed until a change of the conditions is mutually agreed between the contracting parties.

The following Principles and Rules and a template for the DTA were drafted during the EHES Pilot Project in year 2011. Their purpose is to serve as an example when the preparing the relevant documents for EHES in the future:

- Draft principles and rules for sharing and use of the EHES data for future use (see Part C, Section 1.1)
- Draft template for EHES Data Transfer Agreement for future use (see Part C, Section 1.2)

These following versions were used for the data collected in the EHES Joint Action in (2000-2011):

- Principles and rules for sharing and use of the EHES Pilot Project data (see Part C, Section 1.3)
- Template for EHES Data Transfer Agreement for use in the EHES Pilot Project (see Part C, Section 1.4)

1.1 Draft principles and rules for sharing and use of the EHES data for future use

This document was prepared by the EHES Reference Centre and discussed by the partners of the of the EHES Joint Action in year 2011. The purpose of this draft is to serve as an example when preparing the principles and rules for sharing and use of EHES data in future EHES activities.

1.1.1 Introduction

The European Health Examination Survey (EHES) is a collaboration between countries to collect nationally representative health data which are comparable between countries and over time. The data are collected, analyzed and reported nationally. The EHES Reference Centre (EHES RC) is responsible for the European level coordination, standardization, evaluation and reporting. Surveys which join EHES have the obligation to share specified data (EHES Data) with the EHES RC so that this can fulfill its tasks. The EHES data from the different countries also has a major potential for further research, possibly conducted by research groups other than the national data providers and the EHES RC. The purpose of this chapter is to define the principles and rules for the sharing and use of the EHES data.

Any transfer of EHES data between organizations will be covered by Data Transfer Agreements (DTA). The DTAs between the national data providers and the EHES RC are survey specific. Data transfer agreements between EHES RC and third parties can be made only after consultation with the national data providers, and must be in concordance with the DTAs between the data providers and the EHES RC.

1.1.2 General statement

The European Health Examination Survey (EHES) data sharing policy is designed to encourage the use of the data widely for public health benefit while maintaining the legitimate interests of the parties who collected the data and the confidentiality of the participants. Another objective of the data sharing policy is to increase the capacity of the contributing countries to analyze the data on the different health aspects covered by EHES.

1.1.3 Scope of the data sharing policy - EHES Data

The policy concerns the data which have been collected in the framework of EHES using the joint standardization of the data collection. If countries have other health examination survey (HES) data which are comparable with the EHES data, they are encouraged to share the data, on voluntary basis.

The **EHES Data** covered by this data sharing policy includes all data that are necessary for the assessment, analysis and interpretation of the results of the survey:

- individual level data on the survey measurements;

- data on the sampling procedures, sampling units and sampling weights;
- data on eligibility and participation status of those selected to the sample;
- detailed data on the survey procedures; and
- external quality assessment data generated for EHES.

EHES Data include anonymized data on individual persons. The persons whom the data represent cannot be identified from the EHES data.

1.1.4 Ownership of and obligation to share the data

The ownership of the EHES data from each country stays within the country. It is up to each country to decide on further details of the ownership. Each country is encouraged to use their data widely for public benefit. The only limitation is that the use of the data must be ethically acceptable and follow national and international rules and principles of data confidentiality and protection.

There is an obligation for each country to transfer a copy of their EHES Data to the **EHES Reference Centre (EHES RC)**, which has been assigned the responsibility of the European level coordination and standardization of EHES. An exception are possible data which are not allowed to be transferred by national laws and conditions set e.g. by national data protection authorities. The EHES RC will use the data for quality assessment, analysis and reporting, following the principles outlined below. If the EHES RC is shared by more than one organization, the organization receiving the data can share the data with the other organizations which need it to carry out the tasks specified in Section 1.1.6.1 below.

Before the transfer of the data from a country to the EHES RC takes place, the organization hosting the EHES RC will sign a **Data Transfer Agreement (DTA)** with the **Data Provider,** which is the organization which has the authority to provide the national data. The Data Transfer Agreement is in line with this document ("Data sharing rules"), which is appended to the agreement. When necessary, the Data Transfer Agreement can include additional restrictions. The data transfer agreements are survey specific. Each new survey in a country requires a new Data Transfer Agreement.

The EHES Reference Centre can further share the data with other responsible organizations for research on specified topics. The rules and principles for this are outlined in Section 1.1.7 below.

1.1.5 Data Security and Confidentiality

Each Data Provider is responsible for not including information in the EHES Data transferred to the EHES RC which would enable the identification of the person. Furthermore, only age in full years of the person and the year of birth will be provided, but not the date and month of birth. The names of the primary sampling units of the persons will not be provided.

Each country can keep the person identifiers provided it is in agreement with the national ethics and data confidentiality principles and approvals. The country must store them in such a way that the anonymity of the data transferred to the EHES RC is not endangered. The person identifiers are not EHES Data.

The EHES RC has the responsibility for the security and confidentiality of the transferred data.

If EHES RC moves from one organization to another, the EHES database will be transferred to the new organization after this has signed a Data Transfer Agreement with each Data Provider. If the EHES RC ceases from existing, the EHES Data will be sent back to the Data Providers. Alternatively, if agreed with the Data Providers, the data can be destroyed (using a procedure to be approved by the Data Providers) or archived. The organizations which have received EHES Data are responsible for the data security and confidentiality of the data they hold even if the organization no longer hosts the EHES RC.

1.1.6. Publication policy

This publication policy concerns the EHES Data transferred to the EHES RC.

1.1.6.1 Data assessment and basic reporting

The EHES RC can use these data, without further approval, for:

- assessment and documentation of the quality and country-specific characteristics of the data;
- calculating and reporting of health indicators by country; and
- evaluation and development of survey methods.

The health indicators covered here will be specified and agreed with the Data Providers during the EHES Pilot Project (see Part C, Chapter 4). Other indicators can be added after consultation with the Data Providers.

The results can be published in an appropriate dissemination platform. Before any such publication, the Data Providers should approve the results for their country to ensure that the data are presented and interpreted correctly. The Data Providers should respond within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the results were circulated; failure to reply within this time will be taken to mean approval.

Any such publication will have to acknowledge EHES as the data source and have a hyperlink to a web page showing the sites and key personnel of the national HESs in each country as well as other relevant acknowledgements. The Data Providers should provide this information when the results are circulated for approval.

In general, these analyses should be conducted and the results published without delay after the data are available.

1.1.6.2 Additional analysis and research using the data

Proposals are invited from research groups for the analysis of specific research questions using the EHES Data. The Publication Proposal should specify:

- The Proposer and his/her affiliation;
- Purpose of the analysis;
- Specification of data needed for the analysis;
- Place of the analysis;
- Tentative **Manuscript Group**, i.e. those who analyze the data and prepare the manuscript;
- Suggested timeline for the analysis and the publication plan. (As a general rule, a complete manuscript should be ready for submission for publication within a year after the survey data are made available for the analysis.)

The proposals will be approved by the EHES Publications Committee (EHES PC, see Section 1.1.8).

When the EHES PC agrees with a Publication Proposal, this will be circulated to the Data Providers of countries whose data would be used for the analysis. The procedure will follow either alternative A or alternative B, depending on what each Data Provider has agreed in its DTA with the EHES RC:

- **Alternative A - Opting in:** The data can be used for the analysis only if there is a written approval for this from the Data Provider. If the Data Provider wants its data to be used for the analysis, it should indicate this within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the proposal was circulated. The Data Provider is expected to acknowledge the request within 14 days. (It is the responsibility of each Data Provider to ensure that the EHES RC has their up-to-date e-mail addresses.) The Data Provider can propose additional members to the Manuscript Group.
- **Alternative B - Opting out:** The data can be used for the analysis unless the Data Provider indicates the opposite. If the Data Provider does not want its data to be used for the analysis, it should indicate this within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the proposal was circulated. The Data Provider is expected to acknowledge the request within 14 days. (It is the responsibility of each Data Provider to ensure that the EHES RC has their up-to-date e-mail addresses.) The data provider can propose additional members to the Manuscript Group.

As different countries carry out their national HESs at different times, the EHES PC may negotiate with the proposer about the optimal time of starting the analysis and/or publication of the results. It is usually in the interest of the proposer to include as large number of countries as possible, but there may be a legitimate interest for a country to publish its national results first. In all cases, two years after the completion of the HES fieldwork should be considered a sufficient time period for the national analysis.

In case of competing proposals, the EHES PC may give priority to proposal from groups which are contributing EHES surveys for the analysis. It may also suggest collaboration between the proposing groups.

Before any publication of results of the analysis, the Data Providers should approve the results for their country to ensure that the data are presented and interpreted correctly.

The authorship of such publications should follow the principles for Authorship and Contributorship specified in the Uniform Requirements for Manuscripts Submitted to Biomedical Journals by the International Committee of Medical Journal Editors (http://www.icmje.org/index.html).

1.1.7 Sharing data with research groups

There will be a possibility for research groups to get an analysis data set outside the EHES RC. The Data Request for this should accompany or follow the Publication Proposal (see above).

Data will be provided only to a legal body, after signing a **Data Transfer Agreement (DTA)** between the organization hosting the EHES RC and the recipient organization. If the requesting organization is not a

- university or other higher education organisation established by Community law or by the law of a Member State of the European Union;
- organisation or institution for scientific research established under Community law or under the law of a **Member State;**
- national statistical institute of a Member State; or
- another non-profit agency, organisation or institution, which has received the opinion of the Committee on statistical confidentiality, as specified in Article 3 of Regulation (EC) 831/2002 (http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2002:133:0007:0009:EN:PDF),

the recipient needs to seek for a written Data Transfer Agreement directly with the Data Provider in the country.

The Data Transfer Agreements specify the purpose of the use of the data (see Publication Proposal above), the Principles and rules for sharing and use of the EHES data (i.e. this document), responsibility of the recipient on the data security and confidentiality, recipient's responsibility on the documentation of the analysis and what will happen to the data after the analysis has been completed or a specified deadline and any other conditions requested by the Data Providers.

When the EHES PC agrees with a Data Request, this will be circulated by e-mail to the Data Providers of the countries whose data are requested. The procedure will follow either alternative A or alternative B, depending on what each Data Provider has agreed in its DTA with the EHES RC:

- **Alternative A - Opting in:** The data can be shared only if there is a prior written approval for this from the Data Provider. If the Data Provider agrees to share its data, possibly with additional conditions, it should indicate this within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the request was circulated. The Data Provider is expected to acknowledge the request within 14 days. (It is the responsibility of each Data Provider to ensure that the EHES RC has their up-to-date e-mail addresses.)
- **Alternative B - Opting out:** The data can be shared unless the Data Provider indicates the opposite. If the Data Provider refuses to share its data or requests additional conditions for the data sharing, it should indicate this within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the request was circulated. The Data Provider is expected to acknowledge the request within 14 days. (It is the responsibility of each Data Provider to ensure that the EHES RC has their up-to-date e-mail addresses.)

1.1.8 Coordination and decision making

Each Data Provider should appoint a Principal Investigator (PI) who represents the country in EHES and has the authority to approve participation of the country in data analyses and sharing of the country's data with research groups. The PI is the focal contact point of the EHES RC and the EHES PC on publication and data sharing issues.

The EHES Publication and data sharing issues will be coordinated by the EHES RC.

An **EHES Publications Committee (EHES PC)** will be set up jointly by the Data Providers. Its tasks are:

- to approve publication proposals of the EHES data;
- to approve requests to share EHES data with research groups;
- to oversee the EHES reporting and publication process and make possible recommendations for further developing these.

EHES PC will constitute of:

- three PIs from countries contributing to EHES; and
- a representative of EHES RC (ex officio).

The PI members will be elected by the PIs of the countries which have provided data to the EHES RC by the time of the election. The membership of PIs will rotate between countries in such a way that every year one member (the longest serving), will be replaced. (details to be worked out)

The EHES PC will select its Chair from among the three PIs. A total of three members shall form a quorum. Each member of the EHES PC shall be eligible to vote.

1.2 Draft template for EHES Data Transfer Agreement for future use

This is a draft template for the transfer of EHES data from a Data Provider to the EHES Reference Centre. The data transfer agreement is complemented by the document Draft principles and rules for sharing and use of the EHES data for future use.

1.2.1 Data Transfer Agreement for the transfer of EHES data from a national survey to EHES Reference Centre

The European Health Examination Survey (EHES) is a collaboration for standardizing national health examination surveys. The purpose of EHES is to provide data for national and Europe wide planning and evaluation of health policies, health promotion and research. EHES is coordinated by the EHES Reference Centre (EHES RC) *(add here the name of the organization)*. The EHES RC collects data from the national surveys for quality assessment, reporting, data analysis and development of survey methods.

1.2.1.1 Contracting parties

This Data Transfer Agreement (DTA) is between

(add here the name and address of the organization hosting the EHES RC)

(hereafter called "*(XXX)*")

and

(add here the name and address of the organization)

(hereafter called the “Data Provider”).

1.2.1.2 Data covered

This DTA covers the transfer of the EHES data from the

(*specify the survey here, e.g. 2001 EHES Pilot data in Portugal)*

1.2.1.3 Terms and conditions of this agreement

1. Ownership of the data is not transferred to <XXX>.
2. The Data Provider certifies that:
   - it has the authority to transfer the data; and
   - the ethical and other approvals required for the transfer and use of the data for the purposes specified below are in place.
3. The data are transferred without person identifiers. If the Data Provider can link the data to person identifiers, it stores the link in such a way that the anonymity of the data transferred to the EHES RC is not endangered.
4. The EHES RC has the responsibility for the security and confidentiality of the transferred data.
5. The EHES RC can use these data, without further approval, for:
   - assessment and documentation of the quality and country-specific characteristics of the data;
   - reporting of health indicators by country; and
   - evaluation and development of survey methods.

The results can be published in an appropriate dissemination platform. Before any such publication, the Data Provider should approve the results for its country to ensure that the data are presented and interpreted correctly.

Any such publication will have to acknowledge EHES as the data source and have a hyperlink to a web page showing the sites and key personnel of the national HESs in each country as well as other relevant acknowledgements.

6. *(This is relevant only if the tasks of the EHES RC are shared by different organizations) (XXX)* can share the data with *(Specify the organization)*, subject to a separate data transfer agreement between *(XXX)* and *(Specify the organization)*, for the purpose and under the conditions specified in Item 5 above.

7. Additional analysis and research using the data will follow the rules and principles specified in Section 1.1.6.2, Alternative *(select A or B)*, of Section 1.1. Sharing the data with research groups will follow the rules and principles specified in Section 1.1.7, Alternative *(select A or B)*, of Section 1.1.
8. For any transfer of the data from *(XXX)*, other than specified above, a written agreement of the Data Provider will be needed.
9. When *(XXX)* ceases from hosting the EHES RC, the EHES Data will be sent back to the Data Provider and the copy at *(XXX)* will be destroyed. Alternatively, if agreed in writing with the Data Provider, the data can be transferred to another organization and/or archived. *(XXX)* is responsible for the data security and confidentiality of the data it holds even if it no longer hosts the EHES RC.
10. This DTA is complemented by text from Section 1.1: "Principles and rules for sharing and use of the EHES Pilot Project data". In case the terms of this DTA are in conflict with the principles and rules specified in Attachment 1, the terms of this DTA shall prevail.

1.2.1.4 Agreement signatures

For and on behalf of *(add here the name of the organization hosting the EHES RC)*

Date:

Authorized signature:

Print name:

Position in organization:

For and on behalf of *(add here the name of the organization of the Data Provider)*

Date:

Authorized signature:

Print name:

Position in organization:

1.3 Principles and rules for sharing and use of the EHES Pilot Project data

These are the principles and rules for sharing and use of the EHES data from the EHES Pilot Project in 2010-2011. This document will be appended to the Data Transfer Agreements for the pilot data between the Data Providers and the EHES Reference Centre.

1.3.1 Introduction

The European Health Examination Survey (EHES) is a collaboration between countries to collect nationally representative health data which are comparable between countries and over time. The data are collected, analyzed and reported nationally. The EHES Reference Centre (EHES RC) is responsible for the European level coordination, standardization, evaluation and reporting. Surveys which join EHES have the obligation to share specified data (EHES Data) with the EHES RC so that this can fulfill its tasks. The EHES data from the different countries also has a major potential for further research, possibly conducted by research groups other than the national data providers and the EHES RC. The purpose of this chapter is to define the principles and rules for the sharing and use of the EHES data from the EHES Pilot Project.

Any transfer of EHES data between organizations will be covered by Data Transfer Agreements (DTA). The DTAs between the national data providers and the EHES RC are survey specific. Data transfer agreements between EHES RC and third parties can be made only after consultation with the national data providers, and must be in concordance with the DTAs between the data providers and the EHES RC.

1.3.2 General statement

The European Health Examination Survey (EHES) data sharing policy is designed to encourage the use of the data widely for public health benefit while maintaining the legitimate interests of the parties who collected the data and the confidentiality of the participants. Another objective of the data sharing policy is to increase the capacity of the contributing countries to analyze the data on the different health aspects covered by EHES.

1.3.3 Scope of the data sharing policy - EHES Data

The policy concerns the data which have been collected in the EHES Joint Action in years 2010-2011. If countries have other health examination survey (HES) data which are comparable with the EHES data, they are encouraged to share the data, on voluntary basis.

The **EHES Data** covered by this data sharing policy includes all data that are necessary for the assessment, analysis and interpretation of the results of the survey:

- individual level data on the survey measurements;
- data on the sampling procedures, sampling units and sampling weights;
- data on eligibility and participation status of those selected to the sample;
- detailed data on the survey procedures; and
- external quality assessment data generated for EHES.

EHES Data include anonymized data on individual persons. The persons whom the data represent cannot be identified from the EHES data.

1.3.4 Ownership of and obligation to share the data

The ownership of the EHES data from each country stays within the country. It is up to each country to decide on further details of the ownership. Each country is encouraged to use their data widely for public benefit. The only limitation is that the use of the data must be ethically acceptable and follow national and international rules and principles of data confidentiality and protection.

There is an obligation for each country to transfer a copy of their EHES Data to the **EHES Reference Centre (EHES RC)**, which has been assigned the responsibility of the European level coordination and standardization of EHES. An exception are possible data which are not allowed to be transferred by national laws and conditions set e.g. by national data protection authorities. The EHES RC will use the data for quality assessment, analysis and reporting, following the principles outlined below. If the EHES RC is shared by more than one organization, the organization receiving the data can share the data with the other organizations which need it to carry out the tasks specified in Section 1.3.6.1

below. For the EHES Pilot Project in 2009-2012 the EHES RC is shared between the National Institute for Health and Welfare, THL, of Finland, Statistics Norway and Istituto Superiore di Sanità, Italy. The countries transfer their data to THL which has the right to share them with Statistics Norway for the assessment of the sampling procedures and development of imputation and estimation procedures.

Before the transfer of the data from a country to the EHES RC takes place, the organization hosting the EHES RC will sign a **Data Transfer Agreement (DTA)** with the **Data Provider**, which is the organization which has the authority to provide the national data. The Data Transfer Agreement is in line with this document ("Data sharing rules"), which is appended to the agreement. When necessary, the Data Transfer Agreement can include additional restrictions. The data transfer agreements are survey specific. Each new survey in a country requires a new Data Transfer Agreement.

The EHES Reference Centre can further share the data with other responsible organizations for research on specified topics. The rules and principles for this are outlined in Section 1.3.7 below.

1.3.5 Data Security and Confidentiality

Each Data Provider is responsible for not including information in the EHES Data transferred to the EHES RC which would enable the identification of the person. Furthermore, only age in full years of the person and the year of birth will be provided, but not the date and month of birth. The names of the primary sampling units of the persons will not be provided.

Each country can keep the person identifiers provided it is in agreement with the national ethics and data confidentiality principles and approvals. The country must store them in such a way that the anonymity of the data transferred to the EHES RC is not endangered. The person identifiers are not EHES Data.

The EHES RC has the responsibility for the security and confidentiality of the transferred data.

If EHES RC moves from one organization to another, the EHES database will be transferred to the new organization after this has signed a Data Transfer Agreement with each Data Provider. If the EHES RC ceases from existing, the EHES Data will be sent back to the Data Providers. Alternatively, if agreed with the Data Providers, the data can be destroyed (using a procedure to be approved by the Data Providers) or archived. The organizations which have received EHES Data are responsible for the data security and confidentiality of the data they hold even if the organization no longer hosts the EHES RC.

1.3.6 Publication policy

This publication policy concerns the EHES Data transferred to the EHES RC.

1.3.6.1 Data assessment and basic reporting

The EHES RC can use these data, without further approval, for:

- assessment and documentation of the quality and country-specific characteristics of the data;
- calculating and reporting of health indicators by country; and
- evaluation and development of survey methods.

The health indicators covered here will be specified and agreed with the Data Providers during the EHES Pilot Project (see Part C, Chapter 4). Other indicators can be added after consultation with the Data Providers.

The results can be published in an appropriate dissemination platform. Before any such publication, the Data Providers should approve the results for their country to ensure that the data are presented and interpreted correctly. The Data Providers should respond within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the results were circulated; failure to reply within this time will be taken to mean approval.

Any such publication will have to acknowledge EHES as the data source and have a hyperlink to a web page showing the sites and key personnel of the national HESs in each country as well as other relevant acknowledgements. The Data Providers should provide this information when the results are circulated for approval.

In general, these analyses should be conducted and the results published without delay after the data are available.

1.3.6.2 Additional analysis and research using the data

Any proposals for additional analyses (i.e. those not specified in Section 1.3.6.1) of the EHES Data will be circulated to the Data Providers of countries whose data would be used for the analysis. The procedure will follow either alternative A or alternative B, depending on what each Data Provider has agreed in its DTA with the EHES RC:

- **Alternative A - Opting in:** The data can be used for the analysis only if there is a written approval for this from the Data Provider. If the Data Provider wants its data to be used for the analysis, it should indicate this within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the proposal was circulated. The Data Provider is expected to acknowledge the request within 14 days. (It is the responsibility of each Data Provider to ensure that the EHES RC has their up-to-date e-mail addresses.) The Data Provider can propose additional members to the Manuscript Group.
- **Alternative B - Opting out:** The data can be used for the analysis unless the Data Provider indicates the opposite. If the Data Provider does not want its data to be used for the analysis, it should indicate this within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the proposal was circulated. The Data Provider is expected to acknowledge the request within 14 days. (It is the responsibility of each Data Provider to ensure that the EHES RC has their up-to-date e-mail addresses.) The data provider can propose additional members to the Manuscript Group.

Before any publication of results of the analysis, the Data Providers should approve the results for their country to ensure that the data are presented and interpreted correctly.

The authorship of such publications should follow the principles for Authorship and Contributorship specified in the Uniform Requirements for Manuscripts Submitted to Biomedical Journals by the International Committee of Medical Journal Editors (http://www.icmje.org/index.html).

1.3.7. Sharing data with research groups

There will be a possibility for research groups to get an analysis data set outside the EHES RC. The **Data Request** for this should accompany or follow the publication proposal (see Section 1.3.6.2 above).

Data will be provided only to a legal body, after signing a **Data Transfer Agreement (DTA)** between the organization hosting the EHES RC and the recipient organization. If the requesting organization is not a

- university or other higher education organisation established by Community law or by the law of a **Member State** of the European Union;
- organisation or institution for scientific research established under Community law or under the law of a Member State;
- national statistical institute of a Member State; or
- another non-profit agency, organisation or institution, which has received the opinion of the Committee on statistical confidentiality, as specified in Article 3 of Regulation (EC) 831/2002 (http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2002:133:0007:0009:EN:PDF)

the recipient needs to seek for a written Data Transfer Agreement directly with the Data Provider in the country.

The Data Transfer Agreements specify the purpose of the use of the data, the Principles and rules for sharing and use of the EHES data (i.e. this document), responsibility of the recipient on the data security and confidentiality, recipient's responsibility on the documentation of the analysis and what will happen to the data after the analysis has been completed or a specified deadline and any other conditions requested by the Data Providers.

The Data Requests will be circulated by e-mail to the Data Providers of the countries whose data are requested. The procedure will follow either alternative A or alternative B, depending on what each Data Provider has agreed in its DTA with the EHES RC:

- **Alternative A - Opting in:** The data can be shared only if there is a prior written approval for this from the Data Provider. If the Data Provider agrees to share its data, possibly with additional conditions, it should indicate this within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the request was circulated. The Data Provider is expected to acknowledge the request within 14 days. (It is the responsibility of each Data Provider to ensure that the EHES RC has their up-to-date e-mail addresses.)
- **Alternative B - Opting out:** The data can be shared unless the Data Provider indicates the opposite. If the Data Provider refuses to share its data or requests additional conditions for the data sharing, it should in-

dicate this within 28 days (56 days but not later than 30 September if the proposal is sent between 1st June and 31 August) after the request was circulated. The Data Provider is expected to acknowledge the request within 14 days. (It is the responsibility of each Data Provider to ensure that the EHES RC has their up-to-date e-mail addresses.)

1.3.8 Coordination and decision making

Each Data Provider should appoint a **Principal Investigator (PI)** who represents the country in EHES and has the authority to approve participation of the country in data analyses and sharing of the country's data with research groups. The PI is the focal contact point of the EHES RC on publication and data sharing issues.

The EHES Publication and data sharing issues will be coordinated by the EHES RC.

1.4 Template for EHES Data Transfer Agreement for use in the EHES Pilot Project

This is a template for use for the transfer of EHES pilot data to THL. If the Data Provider wants any modifications or additional conditions, it should contact the EHES Reference Centre at THL. The data transfer agreement is complemented by the document Principles and rules for sharing and use of the EHES Pilot Project data.

1.4.1 Data Transfer Agreement for the transfer of EHES data from a national survey to EHES Reference Centre

The European Health Examination Survey (**EHES**) is a collaboration for standardizing national health examination surveys. The purpose of EHES is to provide data for national and Europe wide planning and evaluation of health policies, health promotion and research. EHES is coordinated by the EHES Reference Centre (EHES RC) at the National Institute for Health and Welfare, Helsinki, Finland (THL). The EHES RC collects data from the national surveys for quality assessment, reporting, data analysis and development of survey methods.

1.4.1.1 Contracting parties

This Data Transfer Agreement (DTA) is between

National Institute for Health and Welfare (THL)
Mannerheimintie 166
00271 Helsinki
Finland

(hereafter called “THL”)

and

(add here the name and address of the organization)

(hereafter called the “Data Provider”).

1.4.1.2 Data covered

This DTA covers the transfer of the EHES data from the

(specify the survey here, e.g. 2001 EHES Pilot data in Portugal)

1.4.1.3 Terms and conditions of this agreement

1. Ownership of the data is not transferred to THL.
2. The Data Provider certifies that:
   - it has the authority to transfer the data; and
   - the ethical and other approvals required for the transfer and use of the data for the purposes specified below are in place.
3. The data are transferred without person identifiers. If the Data Provider can link the data to person identifiers, it stores the link in such a way that the anonymity of the data transferred to the EHES RC is not endangered.
4. The EHES RC has the responsibility for the security and confidentiality of the transferred data.
5. The EHES RC can use these data, without further approval, for:
   - assessment and documentation of the quality and country-specific characteristics of the data;
   - reporting of health indicators by country; and
   - evaluation and development of survey methods.

The results can be published in an appropriate dissemination platform. Before any such publication, the Data Provider should approve the results for its country to ensure that the data are presented and interpreted correctly.

Any such publication will have to acknowledge EHES as the data source and have a hyperlink to a web page showing the sites and key personnel of the national HESs in each country as well as other relevant acknowledgements.

6. THL can share the data with Statistics Norway, subject to a separate data transfer agreement between THL and Statistics Norway, for the purpose and under the conditions specified in Item 5 above.
7. Additional analysis and research using the data will follow the rules and principles specified in Section 1.3.6.2, Alternative *(select A or B)*, of Section 1.3. Sharing the data with research groups will follow the rules and principles specified in Section 1.3.7, Alternative A, of Section 1.3.
8. For any transfer of the data from THL, other than specified above, a written agreement of the Data Provider will be needed.
9. When THL ceases from hosting the EHES RC, the EHES Data will be sent back to the Data Provider and the copy at THL will be destroyed. Alternatively, if agreed in writing with the Data Provider, the data can be transferred to another organization and/or archived. THL is responsible for the data security and confidentiality of the data it holds even if it no longer hosts the EHES RC.
10. This DTA is complemented by Attachment 1: "Principles and rules for sharing and use of the EHES Pilot Project data". In case the terms of this DTA are in conflict with the principles and rules specified in Attachment 1, the terms of this DTA shall prevail.

1.4.1.4 Agreement signatures

For and on behalf of the National Institute for Health and Welfare (THL)

Date:

Authorized signature:

Print name:

Position in organization:

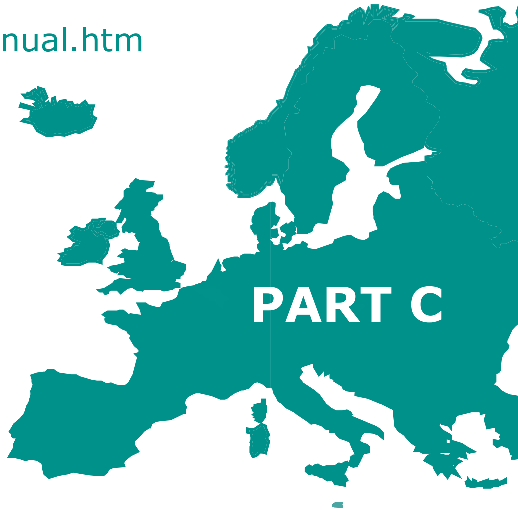
For and on behalf of *(add here the name of the organization of the Data Provider)*

Date:

Authorized signature:

Print name:

Position in organization:



2. EHES RC data management

Ari Haukijärvi[1]

[1]National Institute for Health and Welfare (THL), Helsinki, Finland

2.1 Overview

Collecting anonymous individual level data from EHES countries into the centralized database at EHES Reference Centre (RC) is necessary for quality assessment of the data and for assessing the success of the standardization and documentation of country-specific characteristics of the data. It also facilitates joint analysis and reporting. The database at the Reference Centre is established therefore to serve as a central repository for national data on

- survey procedures,
- sampling,
- eligibility and
- anonymous individual level data on the survey measurements.

2.1.1 Main areas and use cases

The data management in EHES Reference Centre is focused on the following main areas:

a) Data transfer and import

- storing, retrieving and updating data on the national survey procedures
- uploading the survey data to the Reference Centre
- importing the survey data to the central database

b) **Checking of the data**

- checking of the survey data locally in each country before uploading the data to the Reference Centre
- checking of the received survey data in the Reference Centre
- generating reports of the data checks

c) Generating reports

- exporting data for analysis and reports
- producing tables and figures for the survey evaluation and data quality assessment

The communication of the data from countries to EHES Reference Centre as well as the functions of the Reference Centre's data management are shown as *use cases* in the context diagram below. These are discussed in detail in Sections 2.2 to 2.5.

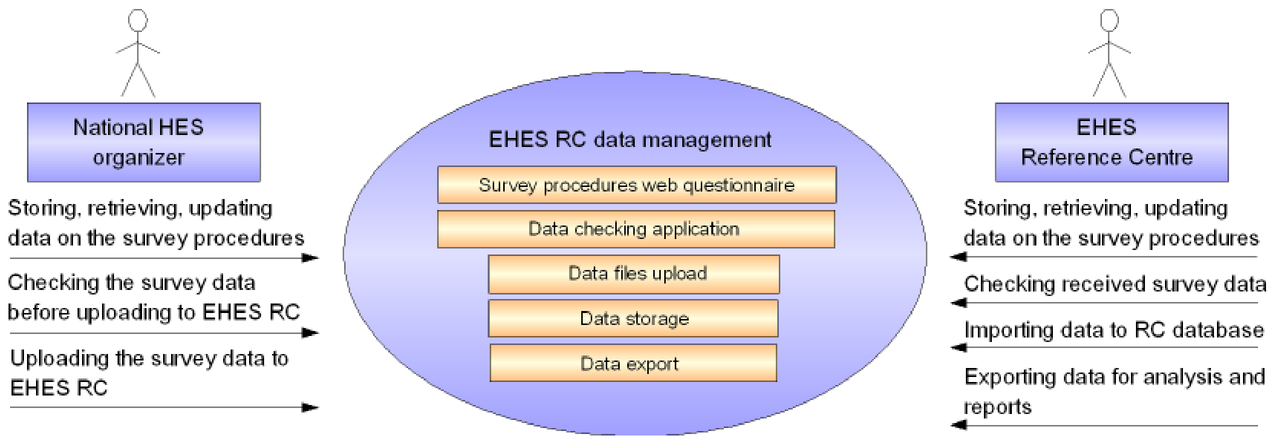


Figure 2.1. The use cases and functions of the RC data management

2.1.2 RC data management system

2.1.2.1 Databases

The data management system at EHES Reference Centre includes the following databases, which are implemented on Oracle relational database management system platform:

- Database for the survey procedures data
- Database for the survey data
- Database for the derived variables and QA data

2.1.2.2 Applications

2.1.2.2.1 Web applications

The web applications - the questionnaire for survey procedures and data files upload interface - are structured as three-tiered www-applications using JSP (Java Server Pages) technology to create dynamic web contents and SSL/TLS support to allow secure encrypted connections to the server. The system is designed to be compliant with W3C standards and demands browser support for SSL protocol. Client computers communicate with RC application server by using web browser as HTTPS user agent.

2.1.2.2.2 Other applications

The applications for data checking and database import of the survey data are implemented in Java programming language. EHES data checking application is a stand-alone Java software application using JWS (Java Web Start) technology and is deployed from RC application server to a local use over the network. All reports and QA tables are generated from database by statistical R software.

An overview of the Reference Centre's data management system is depicted in Figure 2.2.

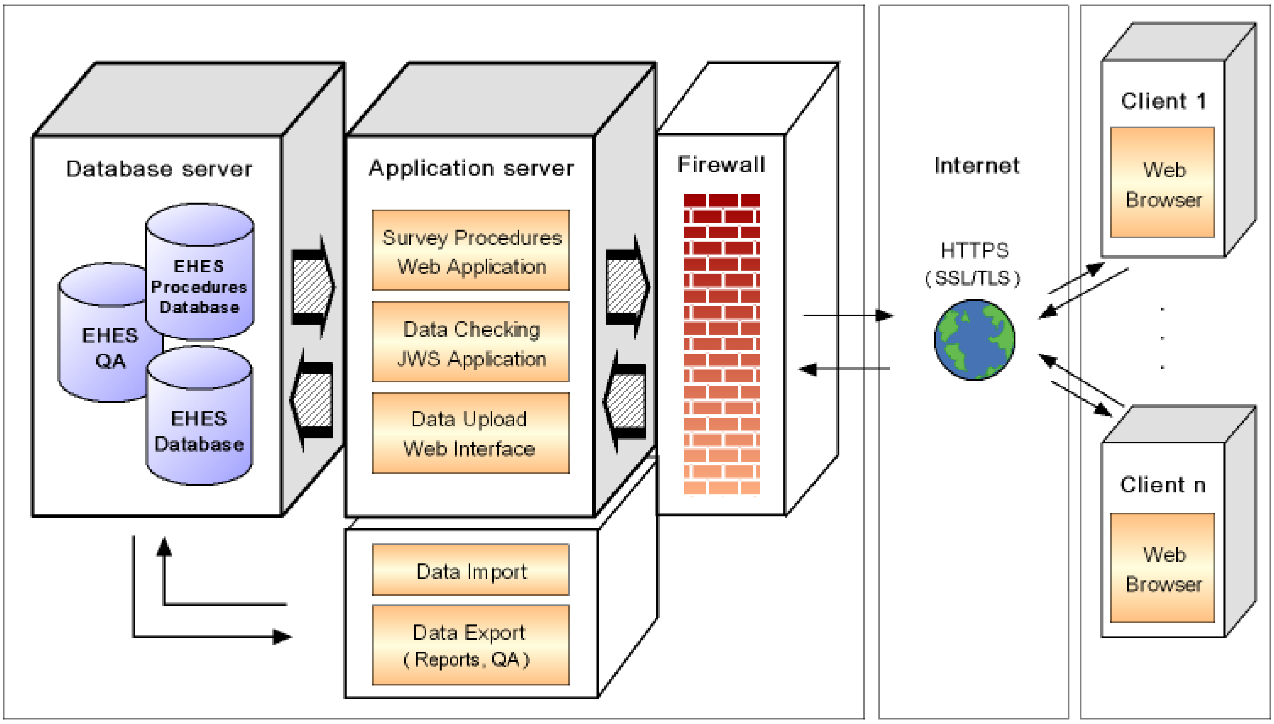


Figure 2.2. An overview of the the RC data management system

2.2 Survey procedures web questionnaire

The survey procedures web questionnaire exists for storing, retrieving and updating data on the national survey procedures. It is a tool for the members of the EHES team to fill-in information about the following topics of the national HESs:

- The period of the survey
- Fieldwork staff - members and training
- Target population and sampling
- Recruitment
- Communication - the plan and using mass media
- Data management
- Order of the measurements and timing of the survey
- Questionnaire administration
- Details on height, weight, waist and blood pressure measurements
- Blood sample collection
- Preparation of plasma/serum samples
- Non-responder data collection
- Quality control

Entering data on national survey procedures to the central database allows comparison of the survey procedures in different manuals and comparison between national manuals and site visit observations.

The survey procedures questionnaire is launched via web browser at EHES Info website (http://www.ehes.info/rc/datatools/datatools.htm). Access to the questionnaire is limited by user login and password which are unique for each EHES country. The questionnaire data are saved into RC database and can be updated later. See the user guide for EHES survey procedures questionnaire (http://www.ehes.info/rc/datatools/EHES_SurveyProceduresDatabase.pdf).

2.3 Transferring survey data to the RC

The central database at EHES Reference Centre is established to store anonymous individual level data on EHES core measurements. The principles and rules for the transfer of survey data from EHES countries are described below in this chapter.

Before any data is transferred to the EHES RC a data transfer agreement (DTA) will be made between EHES RC and country. This is a document signed by both the representative of the country and the RC, as described in Part C, Chapter 1. The data exported from national HES databases should be transferred in a fixed format using tools provided by the Reference Centre.

Transferring survey data to the Reference Centre involves the following steps:

1. Preparation of the data files in EHES countries according to the specification of data items (defined in Section 2.3.1)
2. Checking of the data files in the countries
3. Uploading the data files to the RC
4. Checking of the received data in the RC
5. Importing the received data into the RC database

2.3.1 Format of data transfer

The data transfer format for countries to transfer data on sampling, eligibility, questionnaire and core measurements to the Reference Centre is described in the appendixes by each category:

- Appendix 2a. Data transfer format - Sampling
- Appendix 2b. Data transfer format - Eligibility and participation
- Appendix 2c. Data transfer format - Questionnaire data
    - Background module
    - Health status module
    - Health care module
    - Heath determinants module
    - Background variables module
- Appendix 2d. Data transfer format - Measurements data
    - Blood pressure
    - Height
    - Weight
    - Waist circumference
- Appendix 2e. Data transfer format - Laboratory data

FAQ of the data transfer (Appendix 2f.)

If local questions or data items for measurements are not identical to EHES recommendations, an algorithm how EHES data items are derived from local data items should be delivered to the Reference Centre as a separate document via email.

2.3.2 Data checking

The Reference Centre provides an application to check the data before uploading data to the Reference Centre. The application complies with the specification of data items in Chapter 2.3.1 and allows checking the data variables locally for accuracy and consistency. The data check is formal (e.g. checking that given data conforms to a certain value range) and does not involve data analyses.

The data checking application can be launced at EHES Info website (http://www.ehes.info/rc/datatools/datatools.htm). It is a standalone Java software application using Java Web Start feature and can be deployed from the Reference Centre to a local use over the network. See the user guide for EHES data checking application (http://www.ehes.info/rc/datatools/EHES%20Data%20Check.pdf).

2.3.3 Uploading data files to the RC

Survey data files can be uploaded to EHES Reference Centre via EHES file upload website (https://www3.thl.fi/EHESUpload/), which allows secure transfer of the data. The website is password protected and users are authenticated with the same user ID and password as with the EHES survey procedures questionnaire (see Section 2.2).

2.4 Survey database

The uploaded data files are saved into the Reference Centre's database. Thereafter the data files are checked with the data checking application and an acknowledgement or a request for data correction will be sent via email to the responsible person in the country. When the data are ready, the data from the files are imported into the central survey database at the Reference Centre.

The scheme of the survey database along with the description of database tables and components is included in Appendix 2g: Survey database scheme.

2.5 Derived variables and QA

In order to provide indicators for quality assessments and basic reporting, the derived variables are calculated from the primary survey data and saved into separate, predefined database tables. The data assessment results are saved in predefined QA tables, which serve as the source of QA reports output. All calculations and data import and export during this phase are implemented using statistical R software. Both the derived variables and QA tables are dynamic, i.e. their statuses will change whenever their variables are recalculated with updated survey data.

The scheme of the derived variables and QA database tables is included in Appendix 2h: Tables for the derived variables and QA data.

Appendix 2a. Data transfer format - Sampling

The purpose of this transfer format is to provide an exact and common format for EHES countries to transfer the data on sampling to the Reference Centre.

Specification of data items

Data items on sampling are specified in the three tables below, in which each data item represents one column of data. To transfer the data on sampling three types of data files will be needed (below as A, B and C) . Note the key items in each dataset (file) and that particular data items serve as filters to indicate whether another data item is relevant (value is expected) or irrelevant (empty field, leave the value empty). The specification here is based on Part A, Chapter 3 sampling procedures definition.

Stage 1

Stratification (file A)

Data in the format specified here should be submitted for each stratum in the survey.

COLUMN NAME	TYPE	LENGTH	DEFINITION
COUNTRY	character string	2	*Country code*. **Key item** that identifies the country of the survey. See Part A, Section 12.2 for details. Missing data are not accepted.
SURVEY	character string	2	*Survey number*. **Key item** that uniquely identifies EHES survey in the country. See Part A, Section 12.2 for details. The Survey number is expressed with two digits. Missing data are not accepted.
STRATUM_ID	character string	max 3	*Stratum identifier code*. **Key item** that uniquely identifies the stratum in the survey. Maximum three digits/characters code. If there was no stratification for primary sampling, code STRATUM_ ID = 01. In this case file A has only one record. Missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
STRATUM_ NAME	character string	max 30	*The name of the stratum.* Common name for the stratum. Maximum 30 characters. NNNNN = Missing data.
STRATUM_SIZE	integer		*The size of the stratum.* Total number of SSUs (*N*) in the stratum. 99999999 = Missing data.
DOMAINS	integer		*The number of age-sex strata (domains) in stage 2 sampling*. If no age-sex stratification is used, code DOMAINS = 1.
ST1_ANT_SSU	decimal		*The anticipated number of SSUs to be selected in the stratum (n).* Round to two decimals. 99999.99 = Missing data.
ST1_NO_PSU	integer		*The number of PSUs in the stratum (Mpsu)*. If the first sampling stage was the sampling of individuals or households (e.g. in a pilot survey), code ST1_NO_PSU = 1. 99999 = Missing data.
ST1_SEL_PSU	integer		*The number of PSUs to be selected in the stratum (m)*. Missing data are not accepted.
ST1_CV	integer		*Was* ST1_SEL_PSU *calculated using cost-variance optimization?* 1 = Yes 2 = No Missing data are not accepted.
ST1_CPSU	integer		**Relevant if ST1_CV = 1.** **Irrelevant = empty field.** *The average cost of establishing a PSU in the stratum (Cpsu)*. If relevant, missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
ST1_CSSU	integer		**Relevant if ST1_CV = 1.** **Irrelevant = empty field.** *The average cost of inviting SSU in the stratum (Cssu).* If relevant, missing data are not accepted.
ST1_COST	integer		**Relevant if ST1_CV = 1.** **Irrelevant = empty field.** *The total cost of carrying out the survey in the actual stratum (Cost).* If relevant, missing data are not accepted.
ST1_VWITHIN	decimal		**Relevant if ST1_CV = 1.** **Irrelevant = empty field.** *The average within PSU variance of the calculation variable (Vwithin).* Round to four decimals. 99999.9999 = Missing data.
ST1_VAMONG	decimal		**Relevant if ST1_CV = 1.** **Irrelevant = empty field.** *The variance of the PSU means for the calculation variable (Vamong).* Round to four decimals. 99999.9999 = Missing data.

Selected PSUs/Domains (file B)

Data in the format specified here should be submitted for each selected PSU/Domain.

COLUMN NAME	TYPE	LENGTH	DEFINITION
COUNTRY	character string	2	*Country code.* **Key item** that identifies the country of the survey. See Part A, Section 12.2 for details. Missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
SURVEY	character string	2	*Survey number*. **Key item** that uniquely identifies EHES survey in the country. See Part A, Section 12.2 for details. The Survey number is expressed with two digits. Missing data are not accepted.
STRATUM_ID	character string	max 3	*Stratum identifier code*. **Key item** that uniquely identifies the stratum in the survey. Maximum three digits/ characters code. Missing data are not accepted.
PSU_SN	integer		*PSU serial number.* **Key item** that uniquely identifies the PSU in the stratum. Replaces the real PSU ID (e.g. postcode, municipality code etc.) that is used nationally to identify the PSU for data collection. It is recommended that PSU serial numbers run across strata since this will distinguish PSUs even without using the stratum identifier code. A link between the PSU serial number and the real PSU ID should be maintained by the national survey organizer only. Missing data are not accepted.
PSU_SIZE	integer		*The number of SSUs in the sampling frame from which the Stage 2 sample in the PSU will be selected (Ni)*. **NOTE:** This number may deviate from the size measure used to calculate ST1_PROB. Missing data are not accepted.
ST1_PROB	decimal		*The stage 1 inclusion probability (π_i) used in sampling*. Missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
DOMAIN_ID	character string	max 10	**Relevant if DOMAINS > 1. Irrelevant = empty field.** *Domain identifier (for each domain in the PSU).* **Key item** that uniquely identifies the age-sex domain for the record within PSU. Maximum 10 characters. If relevant, missing data are not accepted.
DOMAIN_SIZE_PSU	integer		**Relevant if DOMAINS > 1. Irrelevant = empty field.** *The number of people in the age-sex domain* within PSU *(Nid).* If relevant, missing data are not accepted.
DOMAIN_SIZE_STR	integer		**Relevant if DOMAINS > 1. Optional if ST1_CV = 2. Irrelevant = empty field.** *The number of people in the age-sex domain in the statum.* 99999999 = Missing data.
ST2_PROB	decimal		*The stage 2 inclusion probability (φ_i) for the PSU or domain.* If age-sex stratification is used this probability will be different for different age-sex domains. Missing data are not accepted.
ST2_ANT_SSU	decimal		*Anticipated stage 2 sample size within the PSU or domain (nid).* Missing data are not accepted.

Stage 2

Selected persons (file C)

Data in the format specified here should be submitted for each person selected to the sample.

COLUMN NAME	TYPE	LENGTH	DEFINITION
COUNTRY	character string	2	*Country code*. **Key item** that identifies the country of the survey. See Part A, Section 12.2 for details. Missing data are not accepted.
SURVEY	character string	2	*Survey number*. **Key item** that uniquely identifies EHES survey in the country. See Part A, Section 12.2 for details. The Survey number is expressed with two digits. Missing data are not accepted.
SERIAL	character string	12	*Serial number*. **Key item** that uniquely identifies the person selected to the survey sample. See Part A, Section 12.2 for details. Please make sure always to use the same Serial number for the same person. If the Serial number is less than 12 characters, use leading zeros. Missing data are not accepted.
STRATUM_ID	character string	max 3	*Stratum identifier code.* Identifies the stratum in the survey. Maximum three digits/ characters code. Missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
PSU_SN	integer		*PSU serial number.* Identifies the PSU in the stratum. Replaces the real PSU ID (e.g. postcode, municipality code etc.) that is used nationally to identify the PSU for data collection. It is recommended that PSU serial numbers run across strata since this will distinguish PSUs even without using the stratum identifier code. A link between the PSU serial number and the real PSU ID should be maintained by the national survey organizer only. Missing data are not accepted.
DOMAIN_ID	character string	max 10	**Relevant if DOMAINS > 1.** **Irrelevant = empty field.** *Domain identifier (for each domain in the PSU).* Identifies the age-sex domain for the record within PSU. Maximum 10 characters. If relevant, missing data are not accepted.
ST2_SEL_SSU	integer		*Number of SSUs actually selected within the PSU or domain.* Must be calculated from the sample when the Stage 2 sampling has taken place. All who were selected should be counted here. This concerns also those who were later found to be not eligible to the sample. Missing data are not accepted.
HOUSEHOLD_UNIT	integer		*Are addresses /households used as sampling units?* 1 = Yes 2 = No Missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
HOUSEHOLD_SN	character string	max 5	**Relevant if HOUSEHOLD_ UNIT = 1.** **Irrelevant = empty field.** *Address /household code.* The code has to be unique in the PSU. Maximum 5 characters. If relevant, missing data are not accepted.
ST3_SAMPLING	integer		**Relevant if HOUSEHOLD_ UNIT = 1.** **Irrelevant = empty field.** *Was there probability sampling within households?* 1= Yes 2 = No If relevant, missing data are not accepted.
ST3_PROB	decimal		*The stage 3 inclusion probability.* If ST3_SAMPLING = 1 then ST3_PROB = individual propability. If HOUSEHOLD_UNIT = 2 or ST3_SAMPLING = 2 then ST3_ PROB = 1. Missing data are not accepted.
ALL_PROB	decimal		*The overall inclusion probability ($\pi_i\varphi_i$ or $\pi_i\varphi_{id}$) of the person selected to the sample.* ALL_PROB = ST1_PROB x ST2_ PROB x ST3_PROB. Missing data are not accepted.
SAMPLING_ WEIGHT	decimal		*The sampling weight (Wi or Wid) of the person selected to the sample.* SAMPLING_WEIGHT = 1/ALL_ PROB. Missing data are not accepted.

General instructions and examples

The data should be prepared in **ASCII format** using semicolon (;) as a delimiter between columns (i.e. data items), **with the names of the columns in the first row**. Each column name should be given **either in big letters or in small letters** (not both).

To simplify the tracing of data transfers between each country and the EHES Reference Centre, the files should be named as:

EHES_CCXX_YYYYMMDD_N.csv, where

- CC is the country code (e.g. EL for Greece)
- XX is the two digit number identifying the survey
- YYYYMMDD is the date (year, month and day) when the file was created (e.g. 20110915)
- N is the sequence number of the data transfer file during the same day (1...n).

Examples of data columns:

- Stratification

COUNTRY	SURVEY	STRATUM_ID	STRATUM_NAME	STRATUM_SIZE	DOMAINS	ST1_ANT_SSU	ST1_NO_PSU	ST1_SEL_PSU	ST1_CV	ST1_CPSU	ST1_CSSU	ST1_COST	ST1_VWITHIN	ST1_VAMONG
NO	01	01	Østfold	72970	8	400.00	43	2	1	10000	50	40000	1.1531	0.0068
NO	01	02	Akershus	259795	8	1370.44	99	9	1	8000	45	133670	1.0905	0.0091
NO	01	03	Oslo	175955	8	928.17	139	22	1	8500	60	242690	1.0936	0.0908
NO	01	04	Hedmark	65801	8	400.00	34	3	1	7000	55	43000	1.1644	0.0060
NO	01	05	Oppland	68775	8	400.00	42	2	1	8000	50	36000	1.1563	0.0054
NO	01	06	Buskerud	101992	8	538.01	56	4	1	9000	60	68281	1.1610	0.0104
NO	01	07	Vestfold	70045	8	400.00	43	3	1	9500	70	56500	1.1313	0.0078
NO	01	08	Telemark	32045	8	400.00	18	3	1	8500	65	51500	1.1592	0.0070
NO	01	09	Aust-Agder	35719	8	400.00	21	3	1	7500	45	40500	1.1788	0.0100
NO	01	10	Vest-Agder	73670	8	400.00	37	3	1	8000	50	44000	1.1594	0.0099
NO	01	11	Rogaland	182525	8	962.83	89	10	1	10000	55	152956	1.1446	0.0235
NO	01	12	Hordaland	169316	8	893.15	108	10	1	9500	40	130726	1.1529	0.0358
NO	01	14	Sogn og Fjordane	26729	8	400.00	12	2	1	8000	35	30000	1.1471	0.0074
NO	01	15	Møre og Romsdal	63492	8	400.00	39	3	1	8500	50	45500	1.1636	0.0099
NO	01	16	Sør-Trøndelag	112555	8	593.74	59	9	1	7500	55	100155	1.1729	0.0347
NO	01	17	Nord-Trøndelag	47782	8	400.00	22	2	1	6500	50	33000	1.1630	0.0038
NO	01	18	Nordland	63237	8	400.00	35	2	1	8000	45	34000	1.1505	0.0042
NO	01	19	Troms	58134	8	400.00	33	4	1	9000	60	60000	1.1535	0.0183
NO	01	20	Finnmark	25602	8	400.00	15	3	1	9500	75	58500	1.1591	0.0072

- Selected PSUs/Domains

COUNTRY	SURVEY	STRATUM_ID	PSU_SN	PSU_SIZE	ST1_PROB	DOMAIN_ID	DOMAIN_SIZE_PSU	DOMAIN_SIZE_STR	ST2_PROB	ST2_ANT_SSU
NO	01	02	103	3957	0.1371	M25_34	377	26141	0.0478	18.02
NO	01	02	103	3957	0.1371	M35_44	663	39975	0.0313	20.73
NO	01	02	103	3957	0.1371	M45_54	601	35218	0.0355	21.33
NO	01	02	103	3957	0.1371	M55_64	337	28111	0.0445	14.98
NO	01	02	103	3957	0.1371	F25_34	386	27438	0.0455	17.58
NO	01	02	103	3957	0.1371	F35_44	704	40418	0.0309	21.77
NO	01	02	103	3957	0.1371	F45_54	554	33824	0.0369	20.47
NO	01	02	103	3957	0.1371	F55_64	335	28670	0.0436	14.60
NO	01	02	029	4133	0.1432	M25_34	304	26141	0.0458	13.91
NO	01	02	029	4133	0.1432	M35_44	690	39975	0.0299	20.65
NO	01	02	029	4133	0.1432	M45_54	610	35218	0.0340	20.72
NO	01	02	029	4133	0.1432	M55_64	437	28111	0.0426	18.60
NO	01	02	029	4133	0.1432	F25_34	368	27438	0.0436	16.05
NO	01	02	029	4133	0.1432	F35_44	685	40418	0.0296	20.28
NO	01	02	029	4133	0.1432	F45_54	629	33824	0.0354	22.25
NO	01	02	029	4133	0.1432	F55_64	410	28670	0.0417	17.11
NO	01	02	052	1698	0.0588	M25_34	112	26141	0.1114	12.48
NO	01	02	052	1698	0.0588	M35_44	249	39975	0.0729	18.14
NO	01	02	052	1698	0.0588	M45_54	255	35218	0.0827	21.09
NO	01	02	052	1698	0.0588	M55_64	218	28111	0.1036	22.58
NO	01	02	052	1698	0.0588	F25_34	176	27438	0.1061	18.68
NO	01	02	052	1698	0.0588	F35_44	250	40418	0.0721	18.01
NO	01	02	052	1698	0.0588	F45_54	220	33824	0.0861	18.94
NO	01	02	052	1698	0.0588	F55_64	218	28670	0.1016	22.14

- Selected persons

COUNTRY	SURVEY	SERIAL	STRATUM_ID	PSU_SN	DOMAIN_ID	ST2_SEL_SSU	HOUSEHOLD_UNIT	HOUSEHOLD_SN	ST3_SAMPLING	ST3_PROB	ALL_PROB	SAMPLING_WEIGHT
NO	01	000000000261	03	4	M25_34	4	2			1	0.0036	277.189
NO	01	000000000262	03	4	M25_34	4	2			1	0.0036	277.189
NO	01	000000000263	03	4	M25_34	4	2			1	0.0036	277.189
NO	01	000000000264	03	4	M25_34	4	2			1	0.0036	277.189
NO	01	000000000265	03	4	M35_44	5	2			1	0.0045	221.691
NO	01	000000000266	03	4	M35_44	5	2			1	0.0045	221.691
NO	01	000000000267	03	4	M35_44	5	2			1	0.0045	221.691
NO	01	000000000268	03	4	M35_44	5	2			1	0.0045	221.691
NO	01	000000000269	03	4	M35_44	5	2			1	0.0045	221.691
NO	01	000000000270	03	4	M45_54	3	2			1	0.0065	154.031
NO	01	000000000271	03	4	M45_54	3	2			1	0.0065	154.031
NO	01	000000000272	03	4	M45_54	3	2			1	0.0065	154.031
NO	01	000000000273	03	4	M55_64	4	2			1	0.0083	120.753
NO	01	000000000274	03	4	M55_64	4	2			1	0.0083	120.753
NO	01	000000000275	03	4	M55_64	4	2			1	0.0083	120.753
NO	01	000000000276	03	4	M55_64	4	2			1	0.0083	120.753
NO	01	000000000277	03	4	F25_34	6	2			1	0.0036	277.189
NO	01	000000000278	03	4	F25_34	6	2			1	0.0036	277.189
NO	01	000000000279	03	4	F25_34	6	2			1	0.0036	277.189
NO	01	000000000280	03	4	F25_34	6	2			1	0.0036	277.189
NO	01	000000000281	03	4	F25_34	6	2			1	0.0036	277.189
NO	01	000000000282	03	4	F25_34	6	2			1	0.0036	277.189
NO	01	000000000283	03	4	F35_44	4	2			1	0.0051	196.920
NO	01	000000000284	03	4	F35_44	4	2			1	0.0051	196.920
NO	01	000000000285	03	4	F35_44	4	2			1	0.0051	196.920
NO	01	000000000286	03	4	F35_44	4	2			1	0.0051	196.920
NO	01	000000000287	03	4	F45_54	3	2			1	0.0068	146.550

Appendix 2b. Data transfer format - Eligibility and participation

The purpose of this transfer format is to provide an exact and common format for EHES countries to transfer the data on eligibility to the Reference Centre. Data in the format specified here should be submitted for each person selected to the sample. The selected persons should be defined in section Appendix 2a. Data transfer format - sampling data.

Specification of data items

Data items on eligibility are specified in the table below. Within the data transfer file each data item represents one column of data, whereas each row represents a person selected to the sample. The key items for each row are COUNTRY, SURVEY and SERIAL. Note that particular data items (e.g. ELIG, EXAM, QUEST) serve as filters to indicate whether another data item is relevant (value is expected) or irrelevant (empty field, leave the value empty).

COLUMN NAME	TYPE	LENGTH	DEFINITION
COUNTRY	character string	2	*Country code*. **Key item** that identifies the country of the survey. See Part A, Section 12.2 for details. Missing data are not accepted.
SURVEY	character string	2	*Survey number*. **Key item** that uniquely identifies EHES survey in the country. See Part A, Section 12.2 for details. The Survey number is expressed with two digits. Missing data are not accepted.
SERIAL	character string	12	*Serial number*. **Key item** that uniquely identifies the person selected to the survey sample. See Part A, Section 12.2 for details. Please make sure always to use the same Serial number for the same person. If the Serial number is less than 12 characters, use leading zeros. Missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
ELIG	integer		*Eligibility:* 1 = Yes 2 = No 3 = Unresolved The definitions for eligiblity, non-eligiblity and unresolved cases are given in Part A, Section 13.2.1. Missing data are not accepted.
REASON	integer		**Relevant if ELIG = 2.** **Irrelevant = empty field.** *Reason for non-eligibility:* 1 = Died before scheduled examination time 2 = Moved out of the PSU between scheduled time 3 = Erroneus data on sampling frame 4 = Other 5 = Insufficient data If relevant, missing data are not accepted.
EXAM	integer		**Relevant if ELIG = 1.** **Irrelevant = empty field.** *Participated in examination:* 1 = Yes 2 = No If relevant, missing data are not accepted.
EXAM_REF	integer		**Relevant if ELIG = 1 and EXAM = 2.** **Irrelevant = empty field.** *Refused to participate to the examination:* 1 = Yes 2 = No 3 = Insufficient data If relevant, missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
EXAM_NOAPP	integer		**Relevant if ELIG = 1, EXAM = 2 and EXAM_REF ≠ 1.** **Irrelevant = empty field.** *Interested but failed to make appointment for examination.* 1 = Yes 2 = No 3 = Insufficient data If relevant, missing data are not accepted.
QUEST	integer		**Relevant if ELIG = 1.** **Irrelevant = empty field.** *Completed questionnaire. (At least something filled in to the questionnaire. Code 'no' only when questionnaire is not returned or is returned completely empty.)* 1 = Yes 2 = No If relevant, missing data are not accepted.
QUEST_REF	integer		**Relevant if ELIG = 1 and QUEST = 2.** **Irrelevant = empty field.** *Refused to participate to questionnaire:* 1 = Yes 2 = No 3 = Insufficient data If relevant, missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
SEX	integer		*Sex*: 1 = Male 2 = Female 9 = Missing data **Code for missing data is allowed only if** **- ELIG ≠ 1 or** **- ELIG = 1, EXAM = 2 and QUEST = 2** **and the information is not available from the sampling frame or other sources.**
AGE	integer		*Age in full years*. Derive AGE primarily from the date of birth and the date of examination. If examinations for core measurements were carried out on several days, use the date when most of the examinations were carried out. If this is not available, use age from the questionnaire. 999 = Missing data **Code for missing data is allowed only if** **- ELIG ≠ 1 or** **- ELIG = 1, EXAM = 2 and QUEST = 2** **and the information is not available from the sampling frame or other sources.** ***Consent information for participants to the examination:***
CONSENT_ANTR	integer		**Relevant if ELIG = 1 and EXAM=1.** **Irrelevant = empty field.** *a. Is there a consent for anthropometric measurements?* 1 = Yes 2 = No If relevant, missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
CONSENT_BP	integer		**Relevant if ELIG = 1 and EXAM=1.** **Irrelevant = empty field.** *b. Is there a consent for blood pressure measurement?* 1 = Yes 2 = No If relevant, missing data are not accepted.
CONSENT_BS	integer		**Relevant if ELIG = 1 and EXAM=1.** **Irrelevant = empty field.** *c. Is there a consent for taking blood samples?* 1 = Yes 2 = No If relevant, missing data are not accepted.
CONSENT_LIPID	integer		**Relevant if ELIG = 1, EXAM=1 and CONSENT_BS = 1.** **Irrelevant = empty field.** *d. Is there a consent for the measurement of lipids from the blood samples?* 1 = Yes 2 = No If relevant, missing data are not accepted.
CONSENT_GLUC	integer		**Relevant if ELIG = 1, EXAM=1 and CONSENT_BS = 1.** **Irrelevant = empty field.** *e. Is there a consent for the measurement of glucose from the blood samples?* 1 = Yes 2 = No If relevant, missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
CONSENT_ STORE	integer		**Relevant if ELIG = 1, EXAM=1 and CONSENT_BS = 1.** **Irrelevant = empty field.** *f. Is there a consent for long term storage of blood samples for future analysis?* 1 = Yes 2 = No If relevant, missing data are not accepted.
CONSENT_DNA	integer		**Relevant if ELIG = 1, EXAM=1 and CONSENT_BS = 1.** **Irrelevant = empty field.** *g. Is there a consent for the analysis of DNA?* 1 = Yes 2 = No If relevant, missing data are not accepted.

General instructions

The data should be prepared in **ASCII format** using semicolon (;) as a delimiter between columns (i.e. data items), **with the names of the columns in the first row**. Each column name should be given **either in big letters or in small letters** (not both).

To simplify the tracing of data transfers between each country and the EHES Reference Centre, the file should be named as follows:

EHES_CCXX_YYYYMMDD_N.csv, where

- CC is the country code (e.g. EL for Greece)
- XX is the two digit number identifying the survey
- YYYYMMDD is the date (year, month and day) when the file was created (e.g. 20110915)
- N is the sequence number of the data transfer file during the same day (1...n).

Example of data columns:

- Eligibility

COUNTRY	SURVEY	SERIAL	ELIG	REASON	EXAM	EXAM_REF	EXAM_NOAPP	QUEST	QUEST_REF	SEX	AGE	CONSENT_ANTR	CONSENT_BP	CONSENT_BS	CONSENT_LIPID	CONSENT_GLUC	CONSENT_STORE	CONSENT_DNA
NO	01	000000000261	1		1			1		1	25	1	1	1	1	1	2	2
NO	01	000000000262	1		1			1		1	31	1	1	1	1	1	2	2
NO	01	000000000263	1		1			1		1	34	1	1	1	1	1	2	2
NO	01	000000000264	1		2	2	1	2	2	1	34							
NO	01	000000000265	1		2	2	3	2	3	1	35							
NO	01	000000000266	1		1			1		1	41	1	1	1	1	1	2	2
NO	01	000000000267	1		1			1		1	40	1	1	1	1	1	2	2
NO	01	000000000268	1		1			1		1	38	1	1	1	1	1	2	2
NO	01	000000000269	1		1			1		1	44	1	1	1	1	1	2	2
NO	01	000000000270	1		1			1		1	45	1	1	1	1	1	2	2
NO	01	000000000271	2	2						1	51							
NO	01	000000000272	2	3						1	54							
NO	01	000000000273	1		1			1		1	55	1	1	1	1	1	2	2
NO	01	000000000274	1		1			1		1	57	1	1	1	1	1	2	2
NO	01	000000000275	1		2	2	3	2	2	1	60							
NO	01	000000000276	1		1			1		1	64	1	1	1	1	1	2	2
NO	01	000000000277	1		1			1		2	25	1	1	1	1	1	2	2
NO	01	000000000278	1		1			1		2	31	1	1	1	1	1	2	2
NO	01	000000000279	1		1			1		2	30	1	1	1	1	1	2	2
NO	01	000000000280	1		2	2	3	2	1	2	26							
NO	01	000000000281	1		2	2	3	2	3	2	33							
NO	01	000000000282	1		1			1		2	34	1	1	1	1	1	2	2
NO	01	000000000283	1		2	2	3	2	3	2	35							
NO	01	000000000284	1		2	2	3	2	3	2	39							
NO	01	000000000285	1		1			1		2	35	1	1	1	1	1	2	2
NO	01	000000000286	1		1			1		2	44	1	1	1	1	1	2	2
NO	01	000000000287	1		1			1		2	46	1	1	1	1	1	2	2

Appendix 2c. Data transfer format - Questionnaire data

The purpose of this transfer format is to provide an exact and common format for EHES countries to transfer the questionnaire data to the Reference Centre. Data in the format specified here should be submitted for each eligible person who completed the questionnaire, as defined in section Appendix 2b. Data transfer format - Eligibility and participation, i.e. for those with ELIG = 1 and QUEST = 1. The data should not be submitted for those who were non-eligible or whose eligibility status is unresolved.

Please make sure that the key items (COUNTRY, SURVEY and SERIAL) for the person are the same as for the data specified in section Appendix 2b. Data transfer format - Eligibility and participation.

Specification of data items

Data items on the EHES questionnaire are specified in the table below. Within the data transfer file each data item represents one column of data, whereas each row represents a person who completed the questionnaire. The key items for each row are COUNTRY, SURVEY and SERIAL. Note that particular data items serve as filters to indicate whether another data item is relevant (value is expected) or irrelevant (empty field, leave the value empty).

COLUMN NAME	TYPE	LENGTH	DEFINITION
COUNTRY	character string	2	*Country code*. **Key item** that identifies the country of the survey. See Part A, Section 12.2 for details. Missing data are not accepted.
SURVEY	character string	2	*Survey number*. **Key item** that uniquely identifies EHES survey in the country. See Part A, Section 12.2 for details. The Survey number is expressed with two digits. Missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
SERIAL	character string	12	*Serial number*. **Key item** that uniquely identifies the person selected to the survey sample. See Part A, Section 12.2 for details. Please make sure always to use the same Serial number for the same person. If the Serial number is less than 12 characters, use leading zeros. Missing data are not accepted.
			European Background Module:
PERSONS	integer		*How many persons live in the household (including yourself)?* 99 = Missing data
SEX_QUEST	integer		*Sex from the questionnaire:* 1 = Male 2 = Female 9 = Missing data
AGE_QUEST			*Self reported age from the questionnaire.* 999 = Missing data
BIRTH_YEAR	integer		*Year of birth*. Four digit number. 9999 = Missing data
MARIT_STATUS	integer		*What is your legal marital status?* 1 = Single (never married) 2 = Married (including registered partnership) 3 = Widowed and not remarried 4 = Divorced and not remarried (including legally separated and dissolved registered partnership) 9 = Missing data
COUPLE	integer		*Are you living with someone as a couple?* 1 = Yes 2 = No 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
EDUCATION	integer		*What is the highest education leaving certificate, diploma or education degree you have obtained?* 1 = No formal education (or below ISCED1) 2 = Primary education (ISCED1) 3 = Lower secondary education (ISCED2) 4 = Upper secondary education (ISCED3) 5 = Post-secondary but not-tertiary education (ISCED4) 6 = First stage of tertiary education (ISCED5) 7 = Second stage of tertiary education (ISCED6) 9 = Missing data
SCHOOL_YEARS	integer		*How many years have you spent at school or in full-time study?* 99 = Missing data
LABOUR_STATUS	integer		*How would you define your current labour status?* 1 = Working for pay or profit 2 = Unemployed 3 = Pupil, student, further training, unpaid work experience 4 = In retirement or early retirement or has given up business 5 = Permanently disabled 6 = In compulsory military or community service 7 = Fulfilling domestic tasks 8 = Other 9 = Missing data
PROFIT	integer		*Have you ever worked for pay or profit?* 1 = Yes 2 = No 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
EMPLOYMENT	integer		**Relevant if PROFIT = 1. Irrelevant = empty field.** *Are/were you an employee, self-employed or working without payment as a family worker?* 1 = Employee 2 = Self-employed 3 = Family worker 9 = Missing data
			European Health Status Module:
HEALTH	integer		*How is your health in general?* 1 = Very good 2 = Good 3 = Fair 4 = Bad 5 = Very bad 8 = Don't know 9 = Missing data
HEALTH_ PROBLEM	integer		*Do you have any longstanding illness or health problem? (Longstanding means illnesses or health problems which have lasted, or are expected to last, for 6 months or more.)* 1 = Yes 2 = No 8 = Don't know 9 = Missing data
LIMITED	integer		*For at least the past 6 months, to what extent have you been limited because of a health problem in activities people usually do?* 1 = Severely limited 2 = Limited but not severely 3 = Not limited at all 8 = Don't know 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
DIAGN_MI	integer		*Do you have or have you ever had myocardial infarction diagnosed by a medical doctor?* 1 = Yes 2 = No 8 = Don't know 9 = Missing data
DIAGN_CHD	integer		*Do you have or have you ever had coronary heart disease (angina pectoris) diagnosed by a medical doctor?* 1 = Yes 2 = No 8 = Don't know 9 = Missing data
DIAGN_HBP	integer		*Do you have or have you ever had high blood pressure (hypertension) diagnosed by a medical doctor?* 1 = Yes 2 = No 8 = Don't know 9 = Missing data
DIAGN_CHOL	integer		*Do you have or have you ever had elevated blood cholesterol diagnosed by a medical doctor?* 1 = Yes 2 = No 8 = Don't know 9 = Missing data
DIAGN_STROKE	integer		*Do you have or have you ever had stroke (cerebral haemorrhage, cerebral thrombosis) diagnosed by a medical doctor?* 1 = Yes 2 = No 8 = Don't know 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
DIAGN_DIAB	integer		*Do you have or have you ever had diabetes diagnosed by a medical doctor?* 1 = Yes 2 = No 8 = Don't know 9 = Missing data
			European Health Care Module:
MEDICINE	integer		*During the past two weeks, have you used any medicines (including dietary supplements such as herbal medicines or vitamins) that were prescribed or recommended for you by a doctor (for women, include also contraceptive pills or other hormones)?* 1 = Yes 2 = No 9 = Missing data
MEDICINE_HBP	integer		**Relevant if MEDICINE = 1. Irrelevant = empty field.** *Were they medicines for high blood pressure?* 1 = Yes 2 = No 9 = Missing data
MEDICINE_CHOL	integer		**Relevant if MEDICINE = 1. Irrelevant = empty field.** *Were they medicines for lowering the blood cholesterol level?* 1 = Yes 2 = No 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
MEDICINE_DIAB	integer		**Relevant if MEDICINE = 1. Irrelevant = empty field.** *Were they medicines for diabetes?* 1 = Yes 2 = No 9 = Missing data
BPM	integer		*When was your blood pressure last measured by a health professional?* 1 = Within the past 12 moths 2 = 1-5 years ago 3 = Never or more than 5 years ago 8 = Don’t know 9 = Missing data
CHOLM	integer		*When was your blood cholesterol last measured?* 1 = Within the past 12 moths 2 = 1-5 years ago 3 = Never or more than 5 years ago 8 = Don’t know 9 = Missing data
GLCM	integer		*When was your blood sugar (glucose) last measured?* 1 = Within the past 12 moths 2 = 1-5 years ago 3 = Never or more than 5 years ago 8 = Don’t know 9 = Missing data
			European Health Determinants Module:
SHEIGHT	integer/ decimal		*How tall are you without shoes (cm)?* 8 = Don’t know 9 = Missing data
SWEIGHT	integer/ decimal		*How much do you weight without clothes and shoes (kg)?* 8 = Don’t know 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
SMOKE	integer		*Do you smoke at all nowadays?* 1 = Yes, daily 2 = Yes, occasionally 3 = Not at all 9 = Missing data *What tobacco product do you smoke each day:*
SMOKE_CIGSM	integer		**Relevant if SMOKE = 1. Irrelevant = empty field.** *a. Manufactured cigarettes?* 1 = Yes 2 = No 9 = Missing data
SMOKE_CIGSH	integer		**Relevant if SMOKE = 1. Irrelevant = empty field.** *b. Hand-rolled cigarettes?* 1 = Yes 2 = No 9 = Missing data
SMOKE_CIGARS	integer		**Relevant if SMOKE = 1. Irrelevant = empty field.** *c. Cigars?* 1 = Yes 2 = No 9 = Missing data
SMOKE_PIPE	integer		**Relevant if SMOKE = 1. Irrelevant = empty field.** *d. Pipefuls of tobacco?* 1 = Yes 2 = No 9 = Missing data
SMOKE_OTHER	integer		**Relevant if SMOKE = 1. Irrelevant = empty field.** *e. Other?* 1 = Yes 2 = No 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
NUM_CIGSM	integer		*On average, how many cigarettes, cigars or pipefuls do you smoke each day:* **Relevant if SMOKE_CIGSM = 1.** **Irrelevant = empty field.** *a. Manufactured cigarettes?* 999 = Missing data
NUM_CIGSH	integer		**Relevant if SMOKE_CIGSH = 1.** **Irrelevant = empty field.** *b. Hand-rolled cigarettes?* 999 = Missing data
NUM_CIGARS	integer		**Relevant if SMOKE_CIGARS = 1.** **Irrelevant = empty field.** *c. Cigars?* 999 = Missing data
NUM_PIPE	integer		**Relevant if SMOKE_PIPE = 1.** **Irrelevant = empty field.** *d. Pipefuls of tobacco?* 999 = Missing data
NUM_OTHER	integer		**Relevant if SMOKE_OTHER = 1.** **Irrelevant = empty field.** *e. Other?* 999 = Missing data
SMOKE_EVER	integer		**Relevant if SMOKE ≠ 1.** **Irrelevant = empty field.** *Have you ever smoked (cigarettes, cigars, pipes) daily, or almost daily, for at least one year?* 1 = Yes 2 = No 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
SMOKE_YEARS	integer		**Relevant if SMOKE = 1 or SMOKE_EVER = 1. Irrelevant = empty field.** *For how many years have you smoked daily?* *Count all separate periods of smoking daily. If you don't remember the exact number of years, please give an estimate.* 99 = Missing data ***European Background Variables Module:***

COLUMN NAME	TYPE	LENGTH	DEFINITION
TOTAL_INCOME	integer		*Which group represents your household's total net monthly income from all the sources (income from work, unemployment benefits, old-age or survivor's benefits, sickness or disability benefits, family/children related allowances, housing allowances, education-related allowance, other regular benefits) after deductions for income tax, national insurance, etc?* 1 = Below 1st decile 2 = Between 1st decile and 2nd decile 3 = Between 2nd decile and 3rd decile 4 = Between 3rd decile and 4th decile 5 = Between 4th decile and 5th decile 6 = Between 5th decile and 6th decile 7 = Between 6th decile and 7th decile 8 = Between 7th decile and 8th decile 9 = Between 8th decile and 9th decile 10 = Above 9th decile 88 = Don't know 99 = Missing data

General instructions

The data should be prepared in **ASCII format** using semicolon (;) as a delimiter between columns (i.e. data items), **with the names of the columns in the first row**. Each column name should be given **either in big letters or in small letters** (not both).

To simplify the tracing of data transfers between each country and the EHES Reference Centre, the file should be named as follows:

EHES_CCXX_YYYYMMDD_N.csv, where

- CC is the country code (e.g. EL for Greece)
- XX is the two digit number identifying the survey
- YYYYMMDD is the date (year, month and day) when the file was created (e.g. 20110915)
- N is the sequence number of the data transfer file during the same day (1...n).

Appendix 2d. Data transfer format - Measurements data

The purpose of this transfer format is to provide an exact and common format for EHES countries to transfer the data on the core measurements to the Reference Centre. Data in the format specified here should be submitted for each eligible person who participated in examination, as defined in section Appendix 2b. Data transfer format - Eligibility and participation, i.e. for those with ELIG = 1 and EXAM = 1. The data should not be submitted for those who were non-eligible or whose eligibility status is unresolved.

Please make sure that the key items (COUNTRY, SURVEY and SERIAL) for the person are the same as for the data specified in section Appendix 2b. Data transfer format - Eligibility and participation.

Specification of data items

Data items on the core measurements are specified in the table below. Within the data transfer file each data item represents one column of data, whereas each row represents a person who participated in examination. The key items for each row are COUNTRY, SURVEY and SERIAL. Note that particular data items serve as filters to indicate whether another data item is relevant (value is expected) or irrelevant (empty field, leave the value empty).

COLUMN NAME	TYPE	LENGTH	DEFINITION
COUNTRY	character string	2	*Country code*. **Key item** that identifies the country of the survey. See Part A, Section 12.2 for details. Missing data are not accepted.
SURVEY	character string	2	*Survey number*. **Key item** that uniquely identifies EHES survey in the country. See Part A, Section 12.2 for details. The Survey number is expressed with two digits. Missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
SERIAL	character string	12	*Serial number*. **Key item** that uniquely identifies the person selected to the survey sample. See Part A, Section 12.2 for details. Please make sure always to use the same Serial number for the same person. If the Serial number is less than 12 characters, use leading zeros. Missing data are not accepted.
			Blood pressure:

COLUMN NAME	TYPE	LENGTH	DEFINITION
BP_MISSING	integer		*Reason if not measured at all*. **Relevant if all measurement values (BP_SYST and BP_DIAST) are coded as 999. Irrelevant = empty field.** Medical reason preventing blood pressure measurement: 1 = Amputation of both arms 2 = Cast on both arms 3 = Open wounds/sores on both arms 4 = Rash on both arms 5 = Malformation of both arms preventing to place the cuff 6 = Lymph node malfunction in both arms e.g. caused by breast cancer and preventing to place the cuff 10 = Other medical reason 11 = Refusal (includes those who did not give consent for blood pressure measurement: see item CONSENT_BP in section Eligibility an participation (Appendix 2b). If CONSENT_BP = 2 then BP_MISSING = 11.) 88 = Other 99 = Missing data
BP_CODER	character string	(max) 12	*Measurer's identification code*. Maximum 12 characters. NNNNN = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
BP_DATE	character string	8	*Date of the measurement*. Date format YYYYMMDD: year (4 digits), month (2 digits), day (2 digits). Express month and day with a leading zero if needed. If blood pressure was not measured, code the date when the person attended the examinations. If the exact day is not known, code DD=99.
BP_TIME	character string	5	*Time (hour:minute) of the measurement*. Express hour and minute with two digits, use a leading zero if needed. If the minutes are not known, code minute=99. 99:99 = Missing data
BP_ROOMTEMP	integer/ decimal		*Room temperature during blood pressure measurement (°C)*. 99 = Missing data
BP_EXERCISED	integer		*Has participant done vigorous physical exercise one hour before blood pressure measurement?* 1 = Yes 2 = No 9 = Missing data
BP_SMOKED	integer		*Has participant smoked one hour before blood pressure measurement?* 1 = Yes 2 = No 9 = Missing data
BP_EATEN	integer		*Has participant eaten one hour before blood pressure measurement?* 1 = Yes 2 = No 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
BP_DRANK	integer		*Has participant drank anything else than water one hour before blood pressure measurement?* 1 = Yes 2 = No 9 = Missing data
BP_BLADDER	integer		*Has participant emtied his/her bladder one hour before blood pressure measurement?* 1 = Yes 2 = No 9 = Missing data
BP_DEVICE	integer		*Type of the measurement device*: 1 = Mercury sphygmomanometer 2 = Automated device 8 = Other 9 = Missing data
BP_DEVICENO	integer		*Number of the measurement device*. 999 = Missing data
BP_ARM	integer		*Arm used for the measurement*: 1 = Right 2 = Left 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
BP_ARMLEFT	integer		**Relevant if BP_ARM = 2. Irrelevant = empty field.** *Reason for use of left arm*. Medical reason: 1 = Paralized and/or spastic right arm 2 = Amputation of right arm 3 = Cast on right arm 4 = Open wounds/sores on right arm 5 = Rash on right arm 6 = Intravenous access device on right arm 7 = Malformation of right arm preventing to place the cuff 8 = Lymph node malfunction in right arm e.g. caused by beast cancer and preventing to place the cuff 88 = Other 99 = Missing data
BP_ARMC	decimal (0.5)		*Arm circumference in cm*. Round to 0.5 cm. 99.9 = Missing data
BP_CUFF	integer		*Used cuff size*: 1 = Small 2 = Medium 3 = Large 4 = Extra large 9 = Missing data
BP_POSTURE	integer		*Posture of the subject during the measurement*: 1 = Sitting 2 = Supine 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
BP_POSTURE_EX	integer		**Relevant if BP_POSTURE = 2.** **Irrelevant = empty field.** *Reason for supine posture*: 1 = Bed driven 8 = Other 9 = Missing data
BP_PULSE	integer		*Pulse rate (60 seconds)*. 999 = Missing data
BP_PULSEREG	integer		*Was the pulse regular?* 1 = Yes 2 = No 9 = Missing data
BP_SYST1	integer		*Systolic blood pressure (mmHg): 1st measurement*. 999 = missing data
BP_DIAST1	integer		*Diastolic blood pressure (mmHg): 1st measurement*. 999 = missing data
BP_SYST2	integer		*Systolic blood pressure (mmHg): 2nd measurement*. 999 = missing data
BP_DIAST2	integer		*Diastolic blood pressure (mmHg): 2nd measurement*. 999 = missing data
BP_SYST3	integer		*Systolic blood pressure (mmHg): 3rd measurement*. 999 = missing data
BP_DIAST3	integer		*Diastolic blood pressure (mmHg): 3rd measurement*. 999 = missing data
			Height:

COLUMN NAME	TYPE	LENGTH	DEFINITION
HEIGHT_MISSING	integer		*Reason if not measured at all*. **Relevant if HEIGHT = 999.9 .** **Irrelevant = empty field.** Reason codes: 1 = Hairstyle or headdress prevents measurement (not possible to undress) 2 = Wheelchair bound or immobile 3 = Unsteady stand 4 = Height exceeds the upper limit of the stadiometer. Specify the upper limit of the stadiometer. 5 = Refusal (includes those who did not give consent for anthropometric measurements: see item CONSENT_ANTR in section Eligibility and particaption (Appendix 2b). If CONSENT_ANTR = 2 then HEIGHT_MISSING = 5.) 8 = Other 9 = Missing data
HEIGHT_ULIMIT	decimal (0.1)		**Relevant if HEIGHT_ MISSING = 4.** **Irrelevant = empty field.** *Specify the upper limit of the stadiometer in cm*. Round to 0.1 cm. 999.9 = Missing data
HEIGHT_CODER	character string	(max) 12	*Measurer's identification code*. Maximum 12 characters. NNNNN = Missing data
HEIGHT_ DEVICENO	integer		*Number of the measurement device*. 999 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
HEIGHT	decimal (0.1)		*Height in cm*. Round to 0.1 cm. 999.9 = Missing data
			Weight:
WEIGHT_MISSING	integer		*Reason if not measured at all*. **Relevant if WEIGHT = 999.99 .** **Irrelevant = empty field.** Reason codes: 1 = Wheelchair bound or immobile 2 = Unsteady stand 3 = Weight exceeds the upper limit of the scale. Specify the upper limit of the scale. 4 = Refusal (includes those who did not give consent for anthropometric measurements: see item CONSENT_ANTR in section Eligibility and participation (Appendix 2b). If CONSENT_ANTR = 2 then WEIGHT_MISSING = 4.) 8 = Other 9 = Missing data
PREGN	integer		*Is participant pregnant?* 1 = Yes 2 = No 9 = Missing data
PREGNWK	integer		**Relevant if PREGN = 1.** **Irrelevant = empty field.** *Pregnancy week*. 99 = Missing data
WEIGHT_ULIMIT	decimal (0.01)		**Relevant if WEIGHT_ MISSING = 3.** **Irrelevant = empty field.** *Specify the upper limit of the scale in kg*. Round to 0.01 kg. 999.99 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
WEIGHT_CODER	character string	(max) 12	*Measurer's identification code*. Maximum 12 characters. NNNNN = Missing data
WEIGHT_ DEVICENO	integer		*Number of the measurement device*. 999 = Missing data
WEIGHT_CLOTH	integer		*Measurement was done in*: 1 = Light underwear 2 = Without heavy outer garments 8 = Other 9 = Missing data
WEIGHT	decimal (0.01)		*Weight in kg*. Round to 0.01 kg. 999.99 = Missing data
			Waist circumference:
WAIST_MISSING	integer		*Reason if not measured at all*. **Relevant if WAIST = 999.9 .** **Irrelevant = empty field.** Reason codes: 1 = Wheelchair bound of immobile or cannot stand 2 = Unsteady stand 3 = Circumference exceeds the upper limit of the tape. Specify the length of the tape. 4 = Large hernia, stomage bag or other device on measurement area 5 = Refusal (includes those who did not give consent for anthropometric measurements: see item CONSENT_ANTR in section Eligibility and participation (Appendix 2b). If CONSENT_ANTR = 2 then WAIST_MISSING = 5.) 8 = Other 9 = Missing data

•

COLUMN NAME	TYPE	LENGTH	DEFINITION
WAIST_ULIMIT	decimal (0.5)		**Relevant if WAIST_ MISSING = 3.** **Irrelevant = empty field.** *Specify the length of the tape in cm*. Round to 0.5 cm. 999.9 = Missing data
WAIST_CODER	character string	(max) 12	*Measurer's identification code*. Maximum 12 characters. NNNNN = Missing data
WAIST_DEVICENO	integer		*Number of the measurement tape*. 999 = Missing data
WAIST_CLOTH	integer		*Measurement was done over*: 1 = Bare skin 2 = Light underwear 3 = Without thick outer garments 8 = Other 9 = Missing data
WAIST	decimal (0.5)		*Waist circumference in cm*. Round to 0.5 cm. 999.9 = Missing data

General instructions

The data should be prepared in **ASCII format** using semicolon (;) as a delimiter between columns (i.e. data items), **with the names of the columns in the first row**. Each column name should be given **either in big letters or in small letters** (not both).

To simplify the tracing of data transfers between each country and the EHES Reference Centre, the file should be named as follows:

EHES_CCXX_YYYYMMDD_N.csv, where

- CC is the country code (e.g. EL for Greece)
- XX is the two digit number identifying the survey
- YYYYMMDD is the date (year, month and day) when the file was created (e.g. 20110915)

Appendix 2e. Data transfer format - Laboratory data

The purpose of this transfer format is to provide an exact and common format for EHES countries to transfer the data on the laboratory samples to the Reference Centre. Data in the format specified here should be submitted for each eligible person who participated in examination, as defined in section Appendix 2b. Data transfer format - Eligibility and participation, i.e. for those with ELIG = 1 and EXAM = 1. The data should not be submitted for those who were non-eligible or whose eligibility status is unresolved.

Please make sure that the key items (COUNTRY, SURVEY and SERIAL) for the person are the same as for the data specified in section Appendix 2b. Data transfer format - Eligibility and participation.

Specification of data items

Data items on the laboratory samples are specified in the table below. Within the data transfer file each data item represents one column of data, whereas each row represents a person who participated in examination. The key items for each row are COUNTRY, SURVEY and SERIAL. Note that particular data items serve as filters to indicate whether another data item is relevant (value is expected) or irrelevant (empty field, leave the value empty).

COLUMN NAME	TYPE	LENGTH	DEFINITION
COUNTRY	character string	2	*Country code*. **Key item** that identifies the country of the survey. See Part A, Section 12.2 for details. Missing data are not accepted.
SURVEY	character string	2	*Survey number*. **Key item** that uniquely identifies EHES survey in the country. See Part A, Section 12.2 for details. The Survey number is expressed with two digits. Missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
SERIAL	character string	12	*Serial number*. **Key item** that uniquely identifies the person selected to the survey sample. See Part A, Section 12.2 for details. Please make sure always to use the same Serial number for the same person. If the Serial number is less than 12 characters, use leading zeros. Missing data are not accepted.
BS_MISSING	integer		*Reason for missing blood samples*. **Relevant if all measurement values (BS_CHOL, BS_HDL, BS_GLC, BS_HBA1C) are coded as 9999.99**. **Irrelevant = empty field**. Reason codes: 1 = Vein could not be found/ difficult to take a blood sample 2 = Refusal (includes those who did not give consent for blood samples: see item CONSENT_BS in section Eligibility and participation (Appendix 2b). If CONSENT_BS = 2 in the eligibility data, BS_ MISSING = 2.) 8 = Other 9 = Missing data
BS_CODER	character string	(max) 12	*Identification code of person drawing the blood samples*. Maximum 12 characters. NNNNN = Missing data
BS_DATE	character string	8	*Date of the sample collection*. Date format YYYYMMDD: year (4 digits), month (2 digits), day (2 digits). Express month and day with a leading zero if needed. If the exact day is not known, code DD=99. 99999999 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
BS_TIME	character string	5	*Time (hour:minute) of the sample collection*. Express hour and minute with two digits, use a leading zero if needed. If the minutes are not known, code minute=99. 99:99 = Missing data
BS_ROOMTEMP	integer/ decimal		*Room temperature during the sample collection (°C)*. 99 = Missing data
BS_FAST	integer		*Has participant fasted at least 8 hours?* 1 = Yes 2 = No 9 = Missing data
BS_TIME_MEAL	character string	5	*Time (hour:minute) of the last meal or sweetened drink.* Express hour and minute with two digits, use a leading zero if needed. If the minutes are not known, code minute=99. 99:99 = Missing data
BS_POSTURE	integer		*Posture of the subject during the blood collection*: 1 = Sitting 2 = Supine 9 = Missing data
BS_POSTURE_EX	integer		**Relevant if BS_POSTURE = 2.** **Irrelevant = empty field.** *Reason for supine posture*: 1 = Fainting 2 = Bed driven 8 = Other 9 = Missing data
BS_ARM	integer		*Arm used for blood collection (if blood collection failed, code the arm where the last attempt was made)*: 1 = Left 2 = Right 9 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
BS_ARMRIGHT	integer		**Relevant if BS_ARM = 2. Irrelevant = empty field.** *Reason for use of right arm*. Medical reason: 1 = No vein found on left arm 2 = No left arm/ Amputation of left arm 3 = Cast on left arm 8 = Other 9 = Missing data
BS_TUBES	integer		*Number of collected tubes*. 99 = Missing data
BS_HEMOLYSED	integer		*Number of hemolysed sample tubes.* 99 = Missing data
BS_DATE_CTRF	character string	8	*Date of centrifuging the samples*. Date format YYYYMMDD: year (4 digits), month (2 digits), day (2 digits). Express month and day with a leading zero if needed. If the exact day is not known, code DD=99. 99999999 = Missing data
BS_TIME_CTRF	character string	5	*Time (hour:minute) of centrifuging the samples*. Express hour and minute with two digits, use a leading zero if needed. If the minutes are not known, code minute=99. 99:99 = Missing data
BS_DATE_FR20	character string	8	*Date of freezing the samples to at least -20 °C*. Date format YYYYMMDD: year (4 digits), month (2 digits), day (2 digits). Express month and day with a leading zero if needed. If the exact day is not known, code DD=99. 99999999 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
BS_TIME_FR20	character string	5	*Time (hour:minute) of freezing the samples to at least -20 °C*. Express hour and minute with two digits, use a leading zero if needed. If the minutes are not known, code minute=99. 99:99 = Missing data
BS_DATE_FR70	character string	8	*Date of transferring samples to at least -70 °C*. Date format YYYYMMDD: year (4 digits), month (2 digits), day (2 digits). Express month and day with a leading zero if needed. If the exact day is not known, code DD=99. 99999999 = Missing data
BS_DATE_LIPID	character string	8	*Date of lipid analysis*. Date format YYYYMMDD: year (4 digits), month (2 digits), day (2 digits). Express month and day with a leading zero if needed. If the exact day is not known, code DD=99. 99999999 = Missing data
BS_DATE_GLC	character string	8	*Date of glucose analysis*. Date format YYYYMMDD: year (4 digits), month (2 digits), day (2 digits). Express month and day with a leading zero if needed. If the exact day is not known, code DD=99. 99999999 = Missing data
BS_CHOL	decimal/ integer		*Serum total cholesterol*. Use the unit (mmol/L or mg/ dL) in which the value was obtained from the laboratory. If both units are available, mmol/L is preferred. The unit is given in item BS_UNIT_CHOL. 9999.99 = Missing data

COLUMN NAME	TYPE	LENGTH	DEFINITION
BS_UNIT_CHOL	integer		**Relevant if BS_CHOL ≠ 9999.99.** **Irrelevant = empty field.** *Unit of BS_CHOL (total cholesterol).* 1 = mmol/L 2 = mg/dL If relevant, missing data are not accepted.
BS_HDL	decimal/ integer		*Serum HDL-cholesterol*. Use the unit (mmol/L or mg/ dL) in which the value was obtained from the laboratory. If both units are available, mmol/L is preferred. The unit is given in item BS_UNIT_HDL. 9999.99 = Missing data
BS_UNIT_HDL	integer		**Relevant if BS_HDL ≠ 9999.99.** **Irrelevant = empty field.** *Unit of BS_HDL (HDL-cholesterol).* 1 = mmol/L 2 = mg/dL If relevant, missing data are not accepted.
BS_GLC	decimal/ integer		*Plasma glucose*. Use the unit (mmol/L or mg/ dL) in which the value was obtained from the laboratory. If both units are available, mmol/L is preferred. The unit is given in item BS_UNIT_GLC. 9999.99 = Missing data
BS_UNIT_GLC	integer		**Relevant if BS_GLC ≠ 9999.99.** **Irrelevant = empty field.** *Unit of BS_GLC (plasma glucose).* 1 = mmol/L 2 = mg/dL If relevant, missing data are not accepted.

COLUMN NAME	TYPE	LENGTH	DEFINITION
BS_HBA1C	integer/ decimal		*Whole blood glycated hemoglobin*. Use the unit (IFCC: mmol/ mol or DCCT: %) in which the value was obtained from the laboratory. If both units are available, mmol/mol is preferred. The unit is given in item BS_UNIT_HBA1C. 9999.99 = Missing data
BS_UNIT_ HBA1C	integer		**Relevant if BS_HBA1C ≠ 9999.99**. **Irrelevant = empty field**. *Unit of BS_HBA1C (whole blood glycated hemoglobin).* 1 = mmol/mol (IFCC) 2 = % (DCCT) If other units were used, please contact the EHES Reference Centre for instructions. If relevant, missing data are not accepted.

General instructions

The data should be prepared in **ASCII format** using semicolon (;) as a delimiter between columns (i.e. data items), **with the names of the columns in the first row.** Each column name should be given **either in big letters or in small letters (not both)**.

To simplify the tracing of data transfers between each country and the EHES Reference Centre, the file should be named as follows:

EHES_CCXX_YYYYMMDD_N.csv, where

- CC is the country code (e.g. EL for Greece)
- XX is the two digit number identifying the survey
- YYYYMMDD is the date (year, month and day) when the file was created (e.g. 20110915)
- N is the sequence number of the data transfer file during the same day (1...n).

Appendix 2f. FAQ of the data transfer

Q1. Do we need separate datasets for each section i.e. sampling, eligibility, EHES questionnaire, and measurement data?

- **ANSWER:** Yes, this is the clearest way. Especially for the sampling data three separate but related data transfer files are needed (one for stratification, one for the selected PSUs/Domains, and one for the selected persons). It is important is that the key items (with the exception of the first two sampling files these are COUNTRY, SURVEY and SERIAL) are always included.

  The data transfer files should be named as EHES_CCXX_YYYYMMDD_N.csv, where
  - CC is the country code (e.g. EL for Greece)
  - XX is the two digit number identifying the survey
  - YYYYMMDD is the date (year, month and day) when the file was created (e.g. 20110915)
  - N is the sequence number of the data transfer file during the same day (1...n).

Q2. Who is considered an eligible participant?

- **ANSWER:** The concept of eligibility is defined in Part A, Section 13.2.1. (E.g. persons who died or moved out of the primarily sampling unit prior to the scheduled examination are not eligible.) The reason for non-eligibility should be coded with item REASON.

Q3. In datasets for sampling and eligibility all sampled population need to be included. How is it with the other datasets which include the actual survey data? Do we still include the whole sample and just leave the data blank for the questionnaire and examination portions?

- **ANSWER:** No, you will need to include all sampled population only in datasets for sampling and eligibility. For the other datasets (actual survey data) you need to include only those subjects who are eligible and who participated in examination and/or completed the questionnaire. However, it is important that the data on each person's eligiblity status matches the data on examination and questionnaire files. (E.g. if ELIG = 1 and EXAM = 1 in the eligibility data, there should exist a row for the same SERIAL number in the examination data.)

Q4. Regarding the questionnaire dataset some of our questions deviate from the final version of the EHES core questionnaire. Should we code these questions in the dataset as we have coded them and go into detail on deviations in the supporting documents?

- **ANSWER:** The questions should be coded according to the questionnaire definition (Appendix 2c) given here. For the data items which are not one-on-one you need to prepare a document describing how you have derived these data items from your local data. If you have any detailed questions about this, please contact the EHES Reference Centre.

Q5. In cases where we did not include certain questions (for example did not ask about last blood pressure measurement or include genetic testing), should we include data on the question in our datasets and simply code them as missing or should we exclude it all together?

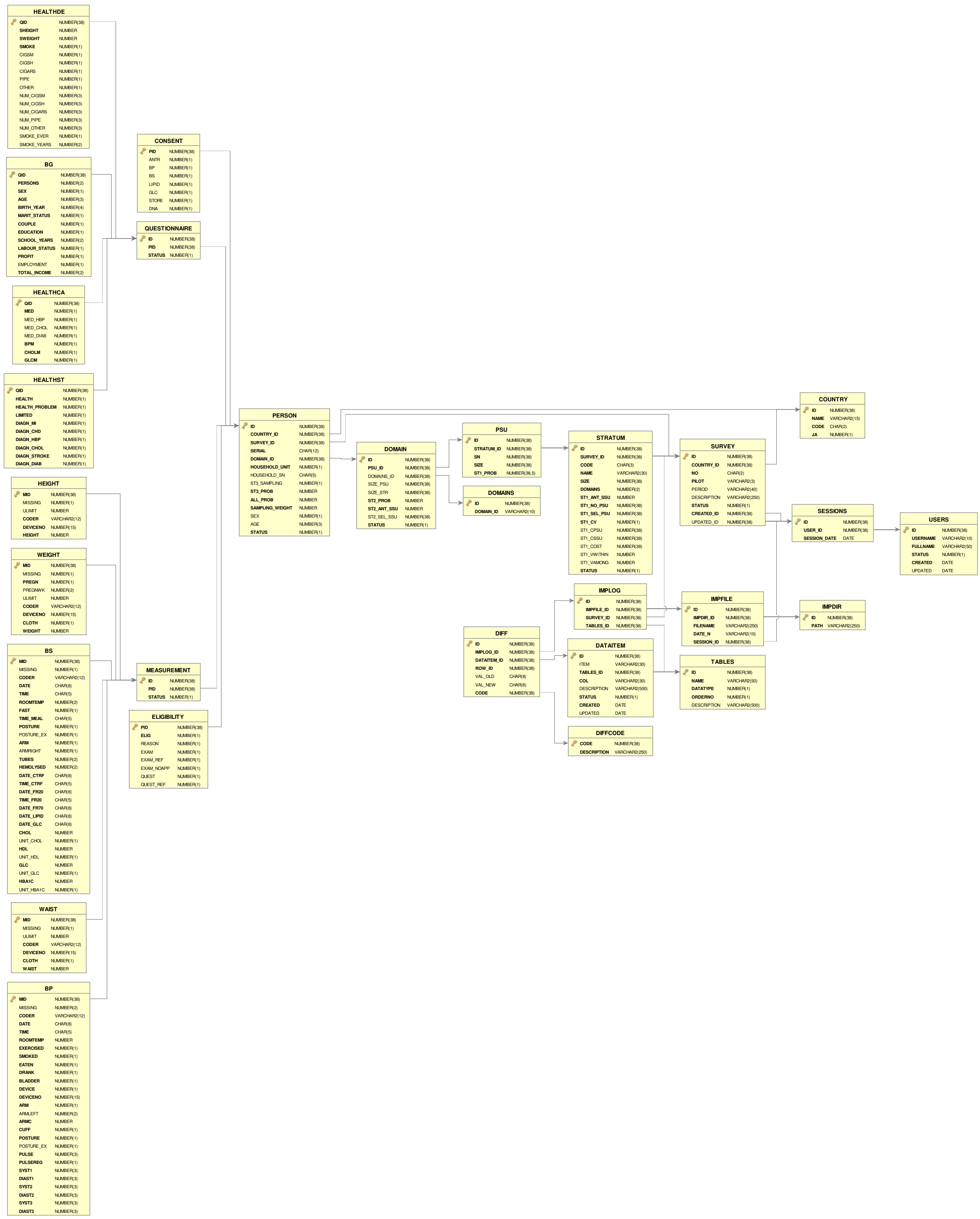
- **ANSWER:** You need to include data on these kind of questions in the datasets and code them as missing data.

Appendix 2g. Survey database scheme

Tables

Table name	Description
COUNTRY	Table for EHES countries.
SURVEY	Table for EHES surveys.
STRATUM	Table for stratification data.
PSU	Table for selected PSUs data.
DOMAIN	Table for selected domains data.
DOMAINS	Table for domain identifiers.
PERSON	Persons selected to the survey sample.
ELIGIBILITY	Table for eligibility data.
CONSENT	Table for consent data.
MEASUREMENT	Measurements root table.
QUESTIONNAIRE	Questionnaires root table.
BP	Table for blood pressure measurements data.
HEIGHT	Table for height measurements data.
WEIGHT	Table for weight measurements data.
WAIST	Table for waist measurements data.
BS	Table for laboratory measurements (blood samples) data.
BG	Table for background data of the questionnaire.
HEALTHST	Table for the questionnaire health status data.
HEALTHCA	Table for the questionnaire health care data.
HEALTHDE	Table for the questionnaire health determinants data.
USERS	Allowed users of the survey database.
SESSIONS	Sessions that modified the state of the database.
IMPDIR	Table for data import directory paths.
IMPFILE	Table for data import files.
IMPLOG	Log table for imported data.
TABLES	Database tables metadata.
DATAITEM	Table for data items metadata.
DIFF	Table for data changes history.
DIFFCODE	Reason for data change/update: code and description.

Graph of the survey database in pdf



Views

View	Description
V_BG	View for the background data of the questionnaire. Used in QA reports.
V_BP	View for the blood pressure data. Used in QA reports.
V_BS	View for the blood sample data. Used in QA reports.
V_DATAITEM	View for the data items and columns metadata.
V_ELIGCONSENT	View for the eligibility and consent data.
V_HEALTHCA	View for the healthcare data of the questionnaire. Used in QA reports.
V_HEALTHDE	View for the health determinants data of the questionnaire. Used in QA reports.
V_HEALTHST	View for the health status data of the questionnaire. Used in QA reports.
V_HEIGHT	View for the height measurements data. Used in QA reports.
V_PERSON	View for the persons selected to the survey sample. Includes stratum and domain data.
V_PERSON_ELIG	View for the basic data on person and eligibility. Used in QA reports.
V_PERSONID	View for the country, survey and serial data of the persons selected to the survey sample. Used in database function F_GET_PERSONID.
V_PSU	selected PSUs data. Used in database function F_GET_PSUID.
V_PSUDOMAIN	View for selected PSUs/Domains data. Used in database function F_GET_DOMAINID.
V_STRATUM	View for stratification data. Used in database function F_GET_STRATUMID.
V_SURVEY	View for EHES surveys. Used in database function F_GET_SURVEYID.
V_USERS	View for the current users of the database. Used in database function F_GET_USERSID.
V_WAIST	View for the waist measurements data. Used in QA reports.
V_WEIGHT	View for the weight measurements data. Used in QA reports.

Functions

Table name	Description
F_GET_COUNTRYID	Gets COUNTRY table id by country code.
F_GET_DIFFCODE	Gets table DIFFCODE code for data change or update by description.
F_GET_DOMAINID	Gets DOMAIN table id by country code, survey number, stratum identifier, psu_sn and domain identifier.
F_GET_DOMAINSID	Gets DOMAINS table id by domain name.
F_GET_IMPDIRID	Gets IMPDIR table id by directory path.
F_GET_IMPLOGID	Gets IMPLOG table id.
F_GET_PERSONID	Gets PERSON table id by country code, survey number and serial.
F_GET_PSUID	Gets PSU table id by country code, survey number, stratum identifier and psu_sn.
F_GET_STRATUMID	Gets STRATUM table id by country code, survey number and stratum identifier code.
F_GET_SURVEYID	Gets SURVEY table id by country code and survey number.
F_GET_TABLESID	Gets TABLES table id by table name.
F_GET_USERSID	Gets USERS table id by user name.

Procedures

Table name	Description
PROC_INSERT_BG	Routine to save data on the questionnaire's background module.
PROC_INSERT_BP	Routine to save data on blood pressure measurements.
PROC_INSERT_BS	Routine to save data on blood samples.
PROC_INSERT_CONSENT	Routine to save data on person's consent.
PROC_INSERT_DIFF	Routine to save changes in survey data update.
PROC_INSERT_DIFFCODE	Routine to save diff codes when updating the survey data.
PROC_INSERT_DOMAIN	Routine to save sampling data on domains.
PROC_INSERT_DOMAINS	Routine to save data on domain names.
PROC_INSERT_ELIGIBILITY	Routine to save data on person's eligibility.

Table name	Description
PROC_INSERT_HEALTHCA	Routine to save data on the questionnaire’s health care module.
PROC_INSERT_HEALTHDE	Routine to save data on the questionnaire’s health determinants module.
PROC_INSERT_HEALTHST	Routine to save data on the questionnaire’s health status module.
PROC_INSERT_HEIGHT	Routine to save data on height measurements.
PROC_INSERT_IMPDIR	Routine to save data import directory metadata.
PROC_INSERT_IMPFILE	Routine to save data import file metadata.
PROC_INSERT_IMPLOG	Routine to save data import log metadata.
PROC_INSERT_MEASUREMENT	Routine to save measurements root data.
PROC_INSERT_PERSON	Routine to save data on persons selected to the survey sample.
PROC_INSERT_PSU	Routine to save data on PSUs.
PROC_INSERT_ QUESTIONNAIRE	Routine to save questionnaires root data.
PROC_INSERT_SESSION	Routine to save data import sessions.
PROC_INSERT_STRATUM	Routine to save data on strata.
PROC_INSERT_WAIST	Routine to save data on waist measurements.
PROC_INSERT_WEIGHT	Routine to save data on weight measurements.

Appendix 2h. Tables for the derived variables and QA data

A. Tables for the derived variables

Table name	Description
DERIVED_AM	Derived variables for anthropometrics.
DERIVED_BG	Derived variables for background items.
DERIVED_BP	Derived variables for blood pressure
DERIVED_BS	Derived variables for laboratory measurements (blood samples).
DERIVED_ELIG	Derived variables for eligibility and consent items.
DERIVED_INCOME	Derived variables for schooling.
DERIVED_ SCHOOLING	Derived variables for schooling.
DERIVED_SMOKING	Derived variables for smoking.

Graph of the derived variables database tables

DERIVED_BS	
PID	INTEGER
COUNTRY	VARCHAR2(255)
SURVEY	VARCHAR2(255)
SERIAL	VARCHAR2(255)
EXAM	VARCHAR2(255)
QUEST	VARCHAR2(255)
SEX	VARCHAR2(255)
BS_CHOL_MMOL	VARCHAR2(255)
BS_HDL_MMOL	VARCHAR2(255)
BS_HBA1C_MMOL	VARCHAR2(255)
BS_FGLC_MMOL	VARCHAR2(255)
BS_NONHDL	VARCHAR2(255)
BS_CHOLHDL_RATIO	VARCHAR2(255)
BS_ECHOL	VARCHAR2(255)
BS_ECHOL6	VARCHAR2(255)
BS_ECHOL7	VARCHAR2(255)
BS_ECHOL8	VARCHAR2(255)
BS_LHDL	VARCHAR2(255)
BS_HRATIO	VARCHAR2(255)
BS_ENONHDL	VARCHAR2(255)
BS_GLYCEMIA	VARCHAR2(255)
BS_PROVDIAB_GLC	VARCHAR2(255)
BS_PROVDIAB_HBA1C	VARCHAR2(255)
BS_DIAB_MSR	VARCHAR2(255)
BS_DIAB_MSR2	VARCHAR2(255)
BS_HCHOL	VARCHAR2(255)
BS_HNONHDL	VARCHAR2(255)
MEDICINE_CHOL1	VARCHAR2(255)
BS_MEDEFF_CHOL	VARCHAR2(255)
MEDICINE_DIAB1	VARCHAR2(255)
BS_MEDEFF_DIAB	VARCHAR2(255)
BS_DIAB_SR	VARCHAR2(255)
BS_DIAB	VARCHAR2(255)
BS_CHOL_AWARE	VARCHAR2(255)
BS_DIAB_AWARE	VARCHAR2(255)
BS_DIAB_SR_AWARE	VARCHAR2(255)
MEDICINE_DIAB_POP	VARCHAR2(255)
MEDICINE_DIAB_ILL	VARCHAR2(255)
MEDICINE_CHOL_POP	VARCHAR2(255)
MEDICINE_CHOL_ILL	VARCHAR2(255)
BS_CHOL_OPTIMAL	VARCHAR2(255)
BS_GLC_OPTIMAL	VARCHAR2(255)

DERIVED_BP	
PID	INTEGER
COUNTRY	VARCHAR2(255)
SURVEY	VARCHAR2(255)
SERIAL	VARCHAR2(255)
EXAM	VARCHAR2(255)
QUEST	VARCHAR2(255)
BP_SYSTM23	VARCHAR2(255)
BP_SYSTM12	VARCHAR2(255)
BP_SYSTM123	VARCHAR2(255)
BP_SYST	VARCHAR2(255)
BP_DIASTM23	VARCHAR2(255)
BP_DIASTM12	VARCHAR2(255)
BP_DIASTM123	VARCHAR2(255)
BP_DIAST	VARCHAR2(255)
MEDICINE_HBP1	VARCHAR2(255)
BP_HBP	VARCHAR2(255)
BP_HBP2	VARCHAR2(255)
BP_HSBP	VARCHAR2(255)
BP_EBP	VARCHAR2(255)
BP_CATEGORY	VARCHAR2(255)
BP_MEDEFF_HBP	VARCHAR2(255)
BP_AWARE	VARCHAR2(255)
BP_AWARE2	VARCHAR2(255)
MEDICINE_HBP_POP	VARCHAR2(255)
MEDICINE_HBP_ILL	VARCHAR2(255)
BP_OPTIMAL	VARCHAR2(255)

DERIVED_AM	
PID	INTEGER
COUNTRY	VARCHAR2(255)
SURVEY	VARCHAR2(255)
SERIAL	VARCHAR2(255)
EXAM	VARCHAR2(255)
QUEST	VARCHAR2(255)
SEX	VARCHAR2(255)
PREGN1	VARCHAR2(255)
BMI	VARCHAR2(255)
WAIST_LARGE	VARCHAR2(255)
BMI_OVERWEIGHT	VARCHAR2(255)
BMI_OBESE	VARCHAR2(255)
BMI_CATEGORY	VARCHAR2(255)
BMI_SR	VARCHAR2(255)
HEIGHT_DIFF	VARCHAR2(255)
WEIGHT_DIFF	VARCHAR2(255)
BMI_DIFF	VARCHAR2(255)
BMI_OVERWEIGHT_SR	VARCHAR2(255)
BMI_OBESE_SR	VARCHAR2(255)
BMI_CATEGORY_SR	VARCHAR2(255)

DERIVED_ELIG	
PID	INTEGER
COUNTRY	VARCHAR2(255)
SURVEY	VARCHAR2(255)
SERIAL	VARCHAR2(255)
ELIG	VARCHAR2(255)
EXAM1	VARCHAR2(255)
REASON1	VARCHAR2(255)
EXAM_REF1	VARCHAR2(255)
EXAM_NOAPP1	VARCHAR2(255)
QUEST1	VARCHAR2(255)
QUEST_REF1	VARCHAR2(255)
CONSENT_ANTR1	VARCHAR2(255)
CONSENT_BP1	VARCHAR2(255)
CONSENT_BS1	VARCHAR2(255)
CONSENT_LIPID1	VARCHAR2(255)
CONSENT_GLC1	VARCHAR2(255)
CONSENT_DNA1	VARCHAR2(255)
CONSENT_STORE1	VARCHAR2(255)
EXAMQUEST	VARCHAR2(255)

DERIVED_SMOKING	
PID	INTEGER
COUNTRY	VARCHAR2(255)
SURVEY	VARCHAR2(255)
SERIAL	VARCHAR2(255)
EXAM	VARCHAR2(255)
QUEST	VARCHAR2(255)
SMOKEY	VARCHAR2(255)
SMOKE_CIGSM1	VARCHAR2(255)
NUM_CIGSM1	VARCHAR2(255)
SMOKE_CIGSH1	VARCHAR2(255)
NUM_CIGSH1	VARCHAR2(255)
SMOKE_CIGARS1	VARCHAR2(255)
NUM_CIGARS1	VARCHAR2(255)
SMOKE_PIPE1	VARCHAR2(255)
NUM_PIPE1	VARCHAR2(255)
SMOKE_OTHER1	VARCHAR2(255)
NUM_OTHER1	VARCHAR2(255)
SMOKE_EVER1	VARCHAR2(255)
SMOKE_YEARS1	VARCHAR2(255)

DERIVED_BG	
PID	INTEGER
COUNTRY	VARCHAR2(255)
SURVEY	VARCHAR2(255)
SERIAL	VARCHAR2(255)
EXAM	VARCHAR2(255)
QUEST	VARCHAR2(255)
EMPLOYMENT1	VARCHAR2(255)
EDUCATION_CAT	VARCHAR2(255)
LABOUR_CAT	VARCHAR2(255)
HEALTH_GOOD	VARCHAR2(255)
LIMITED_YES	VARCHAR2(255)

DERIVED_INCOME	
PID	INTEGER
COUNTRY	VARCHAR2(255)
SURVEY	VARCHAR2(255)
SERIAL	VARCHAR2(255)
EXAM	VARCHAR2(255)
QUEST	VARCHAR2(255)
FINCOME	VARCHAR2(255)
INCOME_CLASS	VARCHAR2(255)

DERIVED_SCHOOLING	
PID	INTEGER
COUNTRY	VARCHAR2(255)
SURVEY	VARCHAR2(255)
SERIAL	VARCHAR2(255)
EXAM	VARCHAR2(255)
QUEST	VARCHAR2(255)
SCHOOL_THIRDS	VARCHAR2(255)

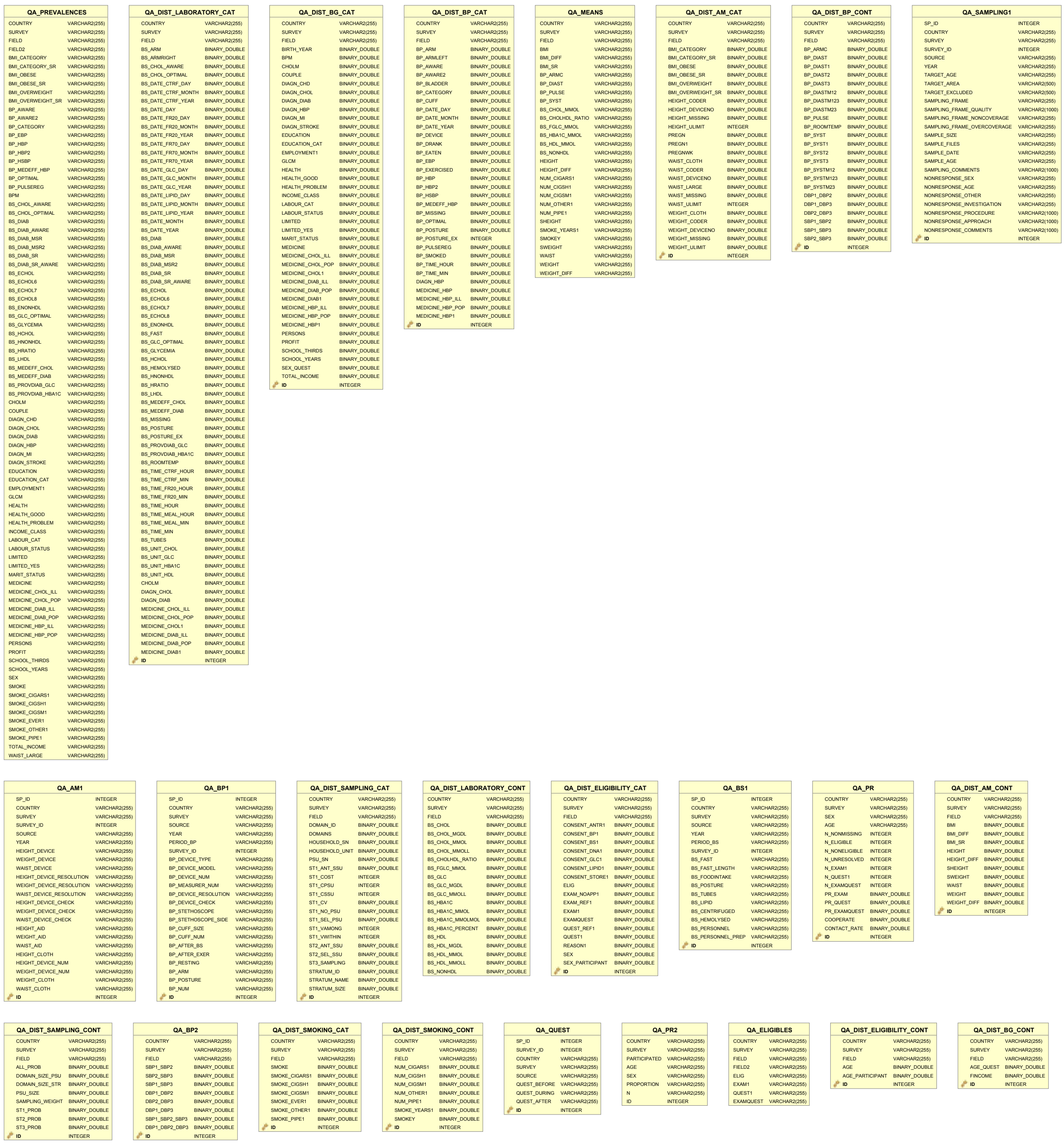
B. Tables for the QA data

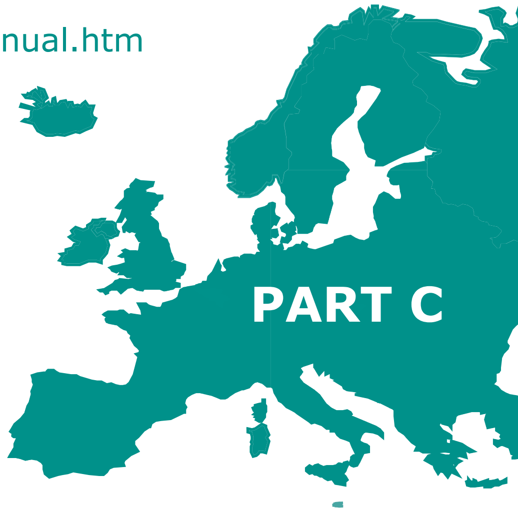
Table name	Description
QA_AM1	Quality assesment table for anthropometrics.
QA_BP1	Quality assesment table for blood pressure

Table name	Description
QA_BP2	Quality assesment table for blood pressure: differences between measurements
QA_BS1	Quality assesment table for blood samples
QA_DIST_AM_CAT	Quality assesment table for anthropometric data: distribution of continuous variables
QA_DIST_AM_CONT	Quality assesment table for anthropometric data: distribution of continuous variables
QA_DIST_BG_CAT	Quality assesment table for background and heath status data: distribution of categorical variables
QA_DIST_BG_CONT	Quality assesment table for background and heath status data: distribution of continuous variables
QA_DIST_BP_CAT	Quality assesment table for anthropometric data: distribution of categorical variables
QA_DIST_BP_CONT	Quality assesment table for blood pressure measurement data: distribution of continuous variables
QA_DIST_ELIGIBILITY_CAT	Quality assesment table for eligibility data: distribution of categorical variables
QA_DIST_ELIGIBILITY_CONT	Quality assesment table for eligibility data: distribution of continuous variables
QA_DIST_LABORATORY_CAT	Quality assesment table for laboratory data: distribution of categorical variables
QA_DIST_LABORATORY_CONT	Quality assesment table for laboratory data: distribution of continuous variables
QA_DIST_SAMPLING_CAT	Quality assesment table for sampling data: distribution of categorical variables
QA_DIST_SAMPLING_CONT	Quality assesment table for sampling data: distribution of continuous variables
QA_DIST_SMOKING_CAT	Quality assesment table for smoking data: distribution of categorical variables
QA_DIST_SMOKING_CONT	Quality assesment table for smoking data: distribution of continuous variables
QA_ELIGIBLES	Quality assesment table: Proportions of eligibility data by sex and age group.
QA_MEANS	Quality assesment table: Non-weighted means by sex and age group.
QA_PR	Participation rates.
QA_PR2	Sex and age distribution of participants and non-participants

Table name	Description
QA_PREVALENCES	Quality assesment table: Prevalences by sex and age group (non-weighted).
QA_QUEST	Quality assesment table for questionnaire
QA_SAMPLING1	Quality assesment table for sampling.

Graph of the QA database tables





3. Evaluation and quality assurance

Hanna Tolonen[1], Päivikki Koponen[1], Sanna Ahonen[1], Johanna Mäki-Opas[1], Ari Haukijärvi[1], Tarja Tuovinen[1], Katri Kilpeläinen[1], Kari Kuulasmaa[1]

[1]National Institute for Health and Welfare (THL), Helsinki, Finland

Evaluation of the national HESs and quality of the data collected by them, includes several steps:

1. National level evaluation through evaluation report
2. European level evaluation through
    - Site visits
    - Evaluation of the national manuals
    - Quality assessment of the collected data

3.1 National level evaluation

Evaluation report should be prepared after the survey is completed. The purpose of the evaluation report is to document experiences gathered during the survey process, in planning, preparation, fieldwork, data management and analysis. It should summarize what worked well, what were the major challenges and difficulties, and what was learned when carrying out the survey. There is no need to repeat the information already given in the national HES manual, rather use references to the national HES manual.

In countries with previous or ongoing national surveys, the evaluation should take into account the extra efforts and challenges met when adjusting existing protocols for European standards.

The evaluation report is targeted to:

- those who will plan and prepare the future survey(s), to point out what needs to be taken into account in the preparations;
- the EHES network, to share experiences, and

- those funding the survey, to assess if the survey was, and will it be in the future worth the investment.

A template for the evaluation report is given in the Appendix 3a. This template is a check list of issues, which at least should be included to the evaluation report. Questions listed under each topic are examples of issues relating to the topic. If other important issues have been observed nationally, it is important to document them as well. The evaluation report should include both positive and negative experiences.

If there is existing format for the evaluation report nationally, that can be used as well.

3.2 European level evaluation

3.2.1 Site visits

Site visits are an evaluation tool, during which the members of the EHES Reference Centre come and observe the field work of the national HES. Site visits are planned to take two days, during which actual measurements are observed with the written consent of the participant.

During the evaluation of the field work, following issues are observed:

- How the communication between team members and participants go
- How the informed consent is obtained
- How team members communicate between each other. What is the team spirit.
- How the privacy of the participants is ensured
- How the data security on the field is ensured
- What kind of devices are used for the measurements
- How often and how the used devices are checked and calibrated
- Is the log book about the checked, their outcomes and calibration kept
- How each measurement is conducted

A check list of the topics to be covered during the site visit has been prepared and is available in the Appendix 3b.

Also EHES RC members discuss with local survey organizers about:

- Selection of the sampling frame
- Sampling
- Obtaining ethical approval for the survey
- Time and resources needed for the planning
- Recruitment of fieldwork personnel
- Data management
- Dissemination plan
- Possibly faced difficulties

From the site visit, a written report is prepared which lists the observations and possible deviations from the EHES protocol.

3.2.2 Evaluation of the national manuals

The evaluation of the national HES manuals included detailed evaluation of how well national manual corresponds to the EHES Manual. Especially for the measurement protocols which should remain identical between all surveys are checked. Otherwise, the contents of national manuals is evaluated against the contents of the EHES Manual.

Topics to be looked in:

- Are all the topics outlined in the EHES Manual included to the local manual
- Sampling
- Ethics
- Data security
- Training
- Recruitment process and related documents
- Questionnaire(s)
- Measurement protocols
- Publicity and dissemination

To helpf on evaluation, a template with check list has been prepared. This template is available in the Appendix 3c.

3.2.3 Quality assessment of the survey data

Quality assessment of the survey data quantify how complete and accurate the collected survey data is.

Survey data will be assessed for the quality based on pre-defined criteria. The outcome of the assessment will be published in the evaluation report which is an important document to support the interpretation of the results. Data assessment is an essential step of the survey to obtain high quality data, which will be comparable between countries and in particular over time, so that reliable long-term trends can be calculated from the data in the future.

The survey data is evaluated and documented by EHES Reference Centre. For different data items different issues are checked.

For questionnaire items:

- questionnaire comparability, i.e. how well nationally used question(s) correspond to the EHES question(s)
- data extraction, i.e. how well EHES data items are extracted for data transfer from nationally collected data
- proportion of missing data for each questionnaire item
- distribution of answer alternatives for each categorical data item
- minimum values, maximum values, 10% percentile, median, 90% percentile and standard deviation for other data items

For measurements:

- recording form comparability with the EHES recording form
- level of measurement protocol details in the national HES manuals in comparison to the EHES Manual
- proportion of missing data
- minimum values, maximum values, 10% percentile, median, 90% percentile and standard deviation
- for blood pressure:
    - proportion of identical readings between subsequent measurements
    - distribution of terminal digits
- for anthropometric measurements:
    - proportion of terminal digits

The system for generating relevant tables and figures for evaluation reports was programmed using R, a free software environment for statistical computing and graphics (R 2011). The data assessment results are saved in the Derived Variables and QA Database (see Part C, Chapter 2) in a format that allows the use of different programs and systems for representing and combining the results. Program codes for generating quality assessment tables and figures of the survey data can be found in the Appendix 3d.

References

- R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/

Appendix 3a. Template for the national evaluation report

Introduction

- To which survey this evaluation report refers to?
- Who prepared and took part in the evaluation (internal, i.e. within the national survey organization and/or external, i.e. by people independent of the national survey organization)?
- How was the evaluation carried out (e.g. informal discussions, feed-back sessions, workshops, meetings, observations)?

Planning and preparation of the survey

- Personnel
  - Did you have enough personnel to plan and prepare the survey?
  - Were there any changes (turnover) in key personnel involved in the planning and preparation?
  - Did the personnel have all the required expertise?
  - If you did not have enough personnel or they were lacking some expertise, how did you solve these shortcomings?
- Time
  - Was there enough time for planning, preparations and fund raising before the actual survey fieldwork started?
  - If not, how did you solve the situation?
- Management and coordination
  - Was there adequate collaboration between internal personnel and other institutes involved in the survey planning and preparation?
  - Was the management structure and coordination adequately specified and functional?
  - Experiences in using management tools?
- **Lessons learned:** What worked well in the planning and preparation phase, what were the major challenges and difficulties, and what would you change and how for the next survey?

Budget

- Were you able to raise adequate funding or were any cuts made to the initial budget?
- How well did your original budget hold at different stages of the survey process?
- If your budget did not hold, what where the biggest differences, and how did you solve this?
- **Lessons learned:** What worked well in the budgeting, what were the major challenges and difficulties, and what would you change and how for the next survey?

Sampling

- Was it easy to get information about the available sampling frames, about their quality and access to them, or was the best/most feasible sampling frame previously well known?
- Did you have professional (statistical) help when doing the sample selection?
- Was the timing of the sample selection optimal in relation to starting the fieldwork?
- Did you obtain adequate contact information for the selected persons?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the sampling, what were the major challenges and difficulties, and what would you change and how for the next survey?

Ethical approval

- What were the key steps in your ethical approval process?
- Which institutes/committees needed to be consulted?
- Did you obtain the ethical approval in time to start the fieldwork as planned?
- What kind of questions the ethical committee(s) raised?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the ethical approval process, what were the major challenges and difficulties, and what would you change and how for the next survey?

Data management

- Did you have adequate professionals to plan, prepare and set up the data management system at national level?
- What were the key steps in planning, preparing and setting up the data management system?
- Did you have adequate hardware and software resources?
- Did your software and hardware function as planned (including fieldwork, data entry and data checking)?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the data management, what were the major challenges and difficulties, and what would you change and how for the next survey?

Training

- Did you collect any feedback from the trainees, e.g. fieldwork staff, after the training? If yes what were the key issues raised in the feedback?
- Did you train all relevant personnel groups?
- Did you have enough time to plan and prepare the training?
- Did you have adequate trainers?
- Was the length of the training period adequate?
- Did the training include all relevant components?
- Did you have enough practical training for the measurements?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the training, what were the major challenges and difficulties, and what would you change and how for the next survey?

Recruitment of fieldwork staff

- Were new persons recruited for the survey fieldwork or did you have previous/permanent personnel?
- Did you have part-time or full-time fieldwork staff? Did this lead to any problems?
- Did you have any problems in recruiting personnel with adequate qualifications and expertise?

- Did you have any turnover of personnel during the fieldwork?
- Were extra persons recruited and trained as substitutes for unexpected absences or sick leaves?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the recruitment of fieldwork staff, what were the major challenges and difficulties, and what would you change and how for the next survey?

Equipment

- Did you have adequate equipment for all measurements?
- Were there any problems in purchasing, hiring or leasing the equipment?
- Did you have any problems to get equipment serviced and/or calibrated?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well with the equipment, what were the major challenges and difficulties, and what would you change and how for the next survey?

Recruitment of participants

- What were the key steps in recruiting participants?
- If you used incentives, did they raise the willingness to participate? Were the incentives cost-effective?
- Did you change the recruitment method during the fieldwork? If yes, why and was it effective?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the recruitment of participants, what were the major challenges and difficulties, and what would you change and how for the next survey?

Appointment scheduling

- If you used software for appointment scheduling,
  - How was the program selected/developed?
  - Did the program work as planned?
- Did you get any feedback from the participants about the available appointment times?

- Did you extend your appointment times during the fieldwork? If yes, how and was it effective?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the appointment scheduling, what were the major challenges and difficulties, and what would you change and how for the next survey?

Examination sites

- How did you select your examination sites?
- Did the examination sites fulfill all your requirements (privacy, easy access etc.)?
- Was it easy/possible to find suitable examination sites?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the the selection of examination sites, what were the major challenges and difficulties, and what would you change and how for the next survey?

Selection of measurements/questionnaire items

- How did you balance between previous national questions and EHES questions/measurement procedures?
- What were your positive and/or negative experiences from the inclusion of the additional measurements/questionnaire components?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the selection of measurements and questionnaire items, what were the major challenges and difficulties, and what would you change and how for the next survey?

Questionnaire and recording form design

- Did you find shortcomings (e.g. errors, illogical jump rules, ambiguous questions) in the questionnaire(s) or recording forms?
- Did you get feedback from the participants about the questionnaires? If yes, what kind?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the questionnaire and recording form design, what were the major challenges and difficulties, and what would you change and how for the next survey?

Quality control

- Did the fieldwork team carry out daily/weekly checking and calibration as planned and record them?
- Were the quality control records monitored regularly?
- Were there any audit/control visits additional to the site visits by EHES RC?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the quality control, what were the major challenges and difficulties, and what would you change and how for the next survey?

Data checking

- What kind of errors did you observe during the routine data checking?
- What losses or potential losses of data were observed during the data inventory?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the data checking, what were the major challenges and difficulties, and what would you change and how for the next survey?

Feedback to the participants

- Was the feedback to the participant given as planned?
- Did the participants consider the feedback and the way how it was given useful?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the feedback to the participants, what were the major challenges and difficulties, and what would you change and how for the next survey?

Laboratory

- Were you satisfied with the laboratory?
- Did your laboratory participate to the External Quality Assessment by EHES RC? If yes, how did the laboratory feel about the assessment and the results?
- If you had any problems, how did you solve them?

- **Lessons learned:** What worked well with the laboratory, what were the major challenges and difficulties, and what would you change and how for the next survey?

Fieldwork coordination and supervision

- Was there sufficient central coordination for the fieldwork (personnel and other resources)?
- Was there adequate supervision and support for the fieldwork staff?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well in the fieldwork coordination and supervision, what were the major challenges and difficulties, and what would you change and how for the next survey?

Publicity

- Did you follow a publicity plan and/or organize any media campaigns?
- Did you have professional public relations personnel to help you?
- Were the actions to raise public awareness of your survey effective?
- Was the publicity correctly timed in relation to fieldwork?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well with publicity, what were the major challenges and difficulties, and what would you change and how for the next survey?

Other aspects of survey

- Were there any other fieldwork practices and aspects that need to be taken into account for the next survey?
- If you had any problems, how did you solve them?
- **Lessons learned:** What worked well with other aspects of survey, what were the major challenges and difficulties, and what would you change and how for the next survey?

Summary and conclusions

- Key findings and recommendations for the future.

Appendix 3b. Contents of the site visits

Sampling

Target population

- Geographical area covered
- Age group
- Permanent residents of the country/citizens/something else
- Inclusion of institutionalized (yes/no)
  - If yes, which institutions
- Other possible limitations to the eligibility (language, etc.)

Sampling frame

- Primary sampling frame
  - When last updated
  - Who maintains the sampling frame
- Secondary sampling frame
  - When last updated
  - Who maintains the sampling frame

Sample size

- Total sample size
- Stratification of the sample size

Sampling method

- Who is responsible for sample selection
- Procedure

Survey procedures

Invitation of the participants

- Form of first contact
- Max number of contacts
- Forms of other contacts

- Use of incentives
  - Have been used or not
  - If yes, what kind of
- On which languages

Examination site

- Fixed clinic(s)/ mobile clinic/ home/ other
- Number of examination sites
- Description of the examination site

Questionnaire administration

- On which languages
- Self-administered/ interviewed
  - Send by post
  - Filled in at the examination site
  - Telephone interview (CATI?)
  - Face-to-face interview (CAPI?)
  - Combination of above
- Length of the questionnaire (number of questions and pages)
- Is questionnaire checked when handed over

Examinations

- Months when examination are conducted
- Days of the week when examinations are conducted
- Hours during which the examinations are conducted

Blood pressure

- Equipment
  - BP measurement device (mercury vs. automated)
    - Brand and type
    - Number and size of cuffs
    - Stethoscope (if mercury); bell or d...
- Measurement conditions
  - Room temperature
  - Quiet/ noise
  - Lighting

- Adequate table for the measurement device and chair for the participant

Height measurement

- Equipment
    - What kind of equipment is used
    - Does it seem that equipment are used right and the calibration is done correctly
- Clothing
    - Is the height measured without shoes, heavy outer garments and hair ornaments?
- Position during the measurement
    - Are back of the head, shoulder blades, buttocks, heels touching the stadiometer/measurement device?
    - Is the ear canal in a level with the cheek bone?
- If the participant is taller than the measurer, was the steps used?

Weight measurement

- Equipment
    - What kind of equipment is used
- Does it seem that equipment are used right and the calibration is done correctly
- Participant's clothing
    - Is the weight measured in underwear?

Waist circumference

- Equipment
    - What kind of equipment is used
- Participant's clothing
    - Is waist measured on bare skin?
- Is the right measurement place palpated?
- Are the participants hands hanging beside the body?

Appendix 3c. National HES manual evaluation template

Evaluation of the national HES manual

During the evaluation process, the following issues are checked:

1. Does the national HES manual cover all the topics listed in the template of the national HES manual, provided by the EHES RC?
2. Are the descriptions under topics written clearly and taking into account the national circumstances?
3. In case the EHES Manual has detailed instructions how the topic, e.g. measurements, should be done, is the EHES recommendations followed so that the comparability of data is ensured?

If the above issues are dealt appropriately, the evaluation form has only a tick mark on 'Yes'. If there is a tick mark on 'No' clarifying comments / questions to those who have prepared the manual are documented.

Aims and purpose of the survey

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Organization and management of the national HES

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Timing of the survey

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Target population and sample size

Target population

	Yes	No
Is this topic included in the national HES manual?		
Does the target population correspond to the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Sample size

	Yes	No
Is this topic included in the national HES manual?		
Does the defined sample size correspond to the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Sampling procedures

Sampling frame

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Sampling procedure

	Yes	No
Is this topic included in the national HES manual?		
Does the procedure follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Legal, ethical and data confidentiality issues

National legislation and regulations

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Process for ethical approval

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Informed consent form

	Yes	No
Is this topic included in the national HES manual, with a form attached?		
Does the informed consent form follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Obtaining informed consent

	Yes	No

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Recruitment, scheduling appointments and motivating participants

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Quality assurance

Training programme

	Yes	No
Is this topic included in the national HES manual?		
Does the training programme follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Pilot survey

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Quality control

	Yes	No
Is this topic included in the national HES manual?		
Do the quality control procedures follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Data management

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Selected measurements

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation, explaining the rationale why each measurement is selected?		

Comments:

Non-participant information

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Selected examination site

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Questionnaire and its administration and validation

Questionnaire

	Yes	No
Is this topic included in the national HES manual, with a questionnaire attached?		
Does the questionnaire include all the items of the EHES questionnaire?		
Is the description clear and adapted to the national situation?		

Comments:

Questionnaire administration mode

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Checking questionnaires and interviewing

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Measurement procedures in the examinations

Blood pressure

	Yes	No
Is this topic included in the national HES manual?		
Do the measurement devices fulfill the EHES requirements?		
Does the measurement protocol follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Height

	Yes	No
Is this topic included in the national HES manual?		
Do the measurement devices fulfill the EHES requirements?		
Does the measurement protocol follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Weight

	Yes	No
Is this topic included in the national HES manual?		
Do the measurement devices fulfill the EHES requirements?		
Does the measurement protocol follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Waist circumference

	Yes	No
Is this topic included in the national HES manual?		
Do the measurement devices fulfill the EHES requirements?		
Does the measurement protocol follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Additional measurements

	Yes	No
Is this topic included in the national HES manual?		

Are international standards/reference tools followed (if yes, specify these under Comments)?		
Is the description clear and adapted to the national situation?		

Comments:

Feedback to the participants

	Yes	**No**
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Analytic laboratory, blood sample collection and storage of the samples

Selection of the analytic laboratory

	Yes	**No**
It this topic included in the national HES manual?		
Does the selection criteria correspond to the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Blood sample collection

	Yes	**No**
Is this topic included in the national HES manual?		
Do the materials and equipment fulfill the EHES requirements?		
Does the sample collection protocol follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Sample logistics

	Yes	**No**
Is this topic included in the national HES manual?		

Does the protocol follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Long term storage of the samples

	Yes	No
Is this topic included in the national HES manual?		
Does the protocol follow the EHES recommendations?		
Is the description clear and adapted to the national situation?		

Comments:

Fieldwork staff

	Yes	No
It this topic included in the national HES manual		
Is the description clear and adapted to the national situation		

Comments:

Coordination of the fieldwork

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Safety issues during fieldwork

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Fieldwork logistics

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Dissemination, publicity and reporting

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Budget and funding

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Risk assessment

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Evaluation

	Yes	No
Is this topic included in the national HES manual?		
Is the description clear and adapted to the national situation?		

Comments:

Appendix 3d. Program codes for quality assessment

Following R scripts are used in EHES data quality assessment of the survey data.

Database connection is established using R package RODBC (Brian Ripley and from 1999 to Oct 2002 Michael Lapsley (2012). RODBC: ODBC Database Access. R package version 1.3-5. http://CRAN.R-project.org/package=RODBC) and SQL (Structured Query Language) is used to import and export data from EHES database.

Program codes use database table variables for the definition of the EHES variables (e.g. code for missing value etc). The definition table variables is of the form:

VARNAME	Name of the variable
VARTYPE	Type of variable: Database, Sampling, Eligibility, Questionnaire, Measurement, Laboratory or Derived
DEFINITION	Definition of the variable and it's values
CONTINUOUS	1 if variable is continuous, 0 if variable is categorical
MISSING	Code for missing value if relevant for variable
DONTKNOW	Code for value "Don't know" if relevant for variable
IRRELEVANT	Code for irrelevant value if relevant for variable
EXAM	1 if variable uses information from examination
QUEST	1 if variable uses information from questionnaire
DIVIDER	For continuous variables, gives width of x axis for plotting
MINIMUM	For continuous variables, gives minimum value of x axis for plotting
MAXIMUM	For continuous variables, gives maximum value of x axis for plotting
MEAN	1 if continuous and variable is used for indicator calculation
PREVALENCE	1 if categorical and variable is used for indicator calculation

R functions for quality assessment of the survey data

drawtable_cat - to derive basic statistics and distribution tables for categorical variables (frequencies, proportion of missing values)

```
drawtable_cat <- function(data, variables, item=NA)  {
    ### data = dataset, variables = definition table
for variables, item = evaluation category
    # tab, output table
    tab <- NULL
    ## to get only categorical variables
    varnames_cat <- sort(variables$VARNAME[variables$C
ONTINUOUS == 0])
    if (item==”lab”) {
        varnames_cat <- setdiff(varnames_cat, c(”BS_
CODER”,”BS_DATE”, ”BS_TIME”, ”BS_TIME_MEAL”, ”BS_
DATE_CTRF”, ”BS_TIME_CTRF”, ”BS_DATE_FR20”, ”BS_TIME_
FR20”, ”BS_DATE_FR70”, ”BS_DATE_LIPID”, ”BS_DATE_GLC”,
”MEDICINE_DIAB”, ”MEDICINE_CHOL”))
        data$BS_DATE_DAY <- substr(data$BS_DATE, 7, 8)
        data$BS_DATE_MONTH <- substr(data$BS_DATE, 5,
6)
        data$BS_DATE_YEAR <- substr(data$BS_DATE, 1,
4)
        data$BS_TIME_HOUR <- substr(data$BS_TIME, 1,
2)
        data$BS_TIME_MIN <- substr(data$BS_TIME, 4, 5)
        data$BS_TIME_MEAL_HOUR <- substr(data$BS_TIME_
MEAL, 1, 2)
        data$BS_TIME_MEAL_MIN <- substr(data$BS_TIME_
MEAL, 4, 5)
        data$BS_DATE_CTRF_DAY <- substr(data$BS_DATE_
CTRF, 7, 8)
        data$BS_DATE_CTRF_MONTH <- substr(data$BS_
DATE_CTRF, 5, 6)
        data$BS_DATE_CTRF_YEAR <- substr(data$BS_DATE_
CTRF, 1, 4)
        data$BS_TIME_CTRF_HOUR <- substr(data$BS_TIME_
CTRF, 1, 2)
        data$BS_TIME_CTRF_MIN <- substr(data$BS_TIME_
CTRF, 4, 5)
        data$BS_DATE_FR20_DAY <- substr(data$BS_DATE_
FR20, 7, 8)
        data$BS_DATE_FR20_MONTH <- substr(data$BS_
DATE_FR20, 5, 6)
        data$BS_DATE_FR20_YEAR <- substr(data$BS_DATE_
FR20, 1, 4)
```

```
        data$BS_TIME_FR20_HOUR <- substr(data$BS_TIME_
FR20, 1, 2)
        data$BS_TIME_FR20_MIN <- substr(data$BS_TIME_
FR20, 4, 5)
        data$BS_DATE_FR70_DAY <- substr(data$BS_DATE_
FR70, 7, 8)
        data$BS_DATE_FR70_MONTH <- substr(data$BS_
DATE_FR70, 5, 6)
        data$BS_DATE_FR70_YEAR <- substr(data$BS_DATE_
FR70, 1, 4)
        data$BS_DATE_LIPID_DAY <- substr(data$BS_DATE_
LIPID, 7, 8)
        data$BS_DATE_LIPID_MONTH <- substr(data$BS_
DATE_LIPID, 5, 6)
        data$BS_DATE_LIPID_YEAR <- substr(data$BS_
DATE_LIPID, 1, 4)
        data$BS_DATE_GLC_DAY <- substr(data$BS_DATE_
GLC, 7, 8)
        data$BS_DATE_GLC_MONTH <- substr(data$BS_DATE_
GLC, 5, 6)
        data$BS_DATE_GLC_YEAR <- substr(data$BS_DATE_
GLC, 1, 4)
    }
    if (item=="smoking") {
        varnames_cat <- setdiff(varnames_cat,
c("SMOKE_CIGARS", "SMOKE_CIGSM", "SMOKE_CIGSH",
"SMOKE_PIPE", "SMOKE_OTHER", "NUM_CIGARS", "NUM_
CIGSM", "NUM_CIGSH", "NUM_PIPE", "NUM_OTHER", "SMOKE_
EVER", "SMOKE_YEARS"))
    }
    if (item=="background") {
        varnames_cat <- setdiff(varnames_cat,
c("EMPLOYMENT", "MEDICINE_DIAB", "MEDICINE_HBP",
"MEDICINE_CHOL"))
    }
    if (item=="bp") {
        varnames_cat <- setdiff(varnames_cat, c("BP_
DEVICENO","BP_DATE", "BP_TIME", "BP_CODER"))
        data$BP_DATE_DAY <- substr(data$BP_DATE, 7, 8)
        data$BP_DATE_MONTH <- substr(data$BP_DATE, 5,
6)
        data$BP_DATE_YEAR <- substr(data$BP_DATE, 1,
4)
        data$BP_TIME_HOUR <- substr(data$BP_TIME, 1,
2)
        data$BP_TIME_MIN <- substr(data$BP_TIME, 4, 5)
    }
    if (item=="pr") {
        varnames_cat <- setdiff(varnames_cat,
c("COUNTRY", "SURVEY", "SERIAL"))
```

```
    }
    if (item=="sampling") {
        data <- data[, c("COUNTRY", "SURVEY",
varnames_cat)]
    }
    else {
       data <- data[, c("COUNTRY", "SURVEY", "EXAM",
"QUEST", varnames_cat)]
    }
    countries <- sort(unique(data$COUNTRY))
    for (name in varnames_cat) {
        mis = variables$MISSING[variables$VARNAME==na
me]
        notknown = variables$DONTKNOW[variables$VARNAM
E==name]
        if (variables$IRRELEVANT[variables$VARNAME==na
me] != "") {
            irr = variables$IRRELEVANT[variables$VARNA
ME==name]
        }
        else if (name %in% c('EMPLOYMENT', 'MEDICINE_
HBP', 'MEDICINE_DIAB', 'MEDICINE_CHOL', 'SMOKE_CIGSM',
'SMOKE_CIGSH', 'SMOKE_CIGARS',
                   'SMOKE_PIPE', 'SMOKE_OTHER',
'SMOKE_EVER', 'BP_MISSING', 'BP_ARMLEFT', 'BP_POSTURE_
EX', 'HEIGHT_MISSING', 'WEIGHT_MISSING',
                    'HEIGHT_ULIMIT', 'WEIGHT_ULIMIT',
'PREGNWK', 'WAIST_MISSING', 'WAIST_ULIMIT', 'BS_
MISSING', 'BS_POSTURE_EX', 'BS_ARMRIGHT', 'BS_UNIT_
CHOL', 'BS_UNIT_HDL', 'BS_UNIT_GLC', 'BS_UNIT_HBA1C',
                    'PREGN')) {
            irr = variables$IRRELEVANT[variables$VARNA
ME==name]
        }
        else {
            irr = NA
        }
        ## finding right way to filter the participants
        if (variables$EXAM[variables$VARNAME==na
me]=="1" & variables$QUEST[variables$VARNAME==na
me]=="1") {
            filters <-  expression(EXAM == '1' | QUEST
== '1')
        }
        else if (variables$EXAM[variables$VARNAME==na
me]=="1") {
            filters <-  expression(EXAM == '1')
        }
        else if (variables$QUEST[variables$VARNAME==na
me]=="1") {
```

```
            filters <-  expression(QUEST == '1')
        }
        else {
            filters <- TRUE
        }
        for (country in countries) {
            surveys <- sort(unique(data[data$COUNTRY
== country,]$SURVEY))
            for (survey in surveys) {
                if (item != "sampling") {
                    selected <- data[data$COUNTRY
== country & data$SURVEY == survey , c("COUNTRY",
"SURVEY", "EXAM", "QUEST", name)]
                }
                else {
                    selected <- data[data$COUNTRY
== country & data$SURVEY == survey , c("COUNTRY",
"SURVEY", name)]
                }
                attach(selected)
                selected = selected[eval(filters) , ]
                detach(selected)
                ## values of the variable
                cols <- sort(unique(selected[, name]))
                ### number of relevants
                if (is.na(irr)) {
                    nrel <- nrow(selected)
                    cols <- setdiff(cols,"")
                }
                else {
                    nrel <- nrow(selected[
!(selected[, name] %in% c(irr,"")) , ])
                    cols <- setdiff(cols, irr)
                }
                ### number of irrelevants
                if (is.na(irr)) {
                    nirr <- NA
                }
                else {
                    nirr <- nrow(selected[  selected[,
name] %in% irr , ])
                }
                if (is.null(tab)) {
                    tab <- as.data.
frame(cbind(country, survey, "n_relevant"),
stringsAsFactors  = F )
                    tab[,dim(tab)[2]+1] <- nrel
                    names(tab) <- c("COUNTRY",
"SURVEY", "FIELD", name)
                }
```

```
                else {
                    if ( nrow(tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="n_relevant",]) != 0
) {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_relevant", name] <-
nrel
                    }
                    else {
                        tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"n_relevant")
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_relevant", name] <-
nrel
                    }
                }
                if ( nrow(tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_irrelevant",]) != 0
) {
                         tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_irrelevant", name]
<- nirr
                }
                else {
                    tab[dim(tab)[1]+1 , c("COUNTRY",
"SURVEY", "FIELD")] <- c(country, survey, "n_
irrelevant")
                    tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_irrelevant", name]
<- nirr
                }
                if ( length(cols) != 0 ) {

                    for ( col in cols ) {
                        n_field <-
nrow(selected[selected[, name] == col ,])
                        if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== col, ]) ==
0 ) {
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
col)
                            tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==col, name] <- n_field
                        }
                        else  {
                            tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==col, name] <- n_field
                        }
```

```
                    n_field = NULL
                }
                if (variables$EXAM[variables$VARNA
ME==name]=="1" & variables$QUEST[variables$VARNAME==na
me]=="1") {
                    nmis <- nrow(selected[
(selected$QUEST=="1" & selected$EXAM=="1") &
selected[, name] %in% setdiff(mis,irr) , ])
                }
                else if (variables$EXAM[variables$
VARNAME==name]=="1" ) {
                    nmis <- nrow(selected[
( selected$EXAM=="1") & selected[, name] %in%
setdiff(mis,irr) , ])
                }
                else if (variables$QUEST[variables
$VARNAME==name]=="1" ) {
                    nmis <- nrow(selected[  (
selected$QUEST=="1") & selected[, name] %in%
setdiff(mis,irr) , ])
                }
                else {
                    nmis <- nrow(selected[
selected[, name] %in% setdiff(mis,irr) , ])
                }
                #### proportion of missing data
                propmis <- 100* signif( (nmis) /
nrel , 2)
                nmis = NULL
                if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== "missing_
percent", ]) == 0 ) {
                    tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"missing_percent")
                    tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="missing_percent",
name] <- propmis
                }
                else  {
                    tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="missing_percent",
name] <- propmis
                }
                #### proportion of don't knows
                if (is.na(notknown)) {
                }
                else if (sum( notknown != "") !=
0) {
                    nnot <- nrow(selected[
```

```
selected[, name] %in% setdiff(notknown, irr), ])
                        propnot <- 100* signif( (nnot)
/ nrel , 2)
                        nnot = NULL
                        if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== "dontknow_
percent", ]) == 0 ) {
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"dontknow_percent")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="dontknow_percent",
name] <- propnot
                        }
                        else  {
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="dontknow_percent",
name] <- propnot
                        }
                    }
                }
            }
        }
    }
    return(tab)
}
```

drawtable_cont - to derive basic statistics and distribution tables for continuous variables (mean, standard deviation, percentiles, minimum and maximum values, proportion of missing values and terminal digits)

```
### Fuction to create dataset including distributions
of continuous measurement variables, QA_DIST_XXX
drawtable_cont <- function(data, variables, item=NA,
plots=F, distr=F, restr="")  {
    ### data = dataset, variables = definition table of
variables, item = evaluation categor, plots = true if
histogram is plotted
    tab <- NULL
    ## to get continuous variables
    varnames_cont <- sort(variables$VARNAME[variables$
CONTINUOUS == 1])
    if (item=="bp") {
        data$BP_ROOMTEMP = gsub(" ", ".", data$BP_
ROOMTEMP)
    }
    if (item=="lab") {
        varnames_cont = setdiff(varnames_cont, c("BS_
CHOL", "BS_HDL", "BS_GLC", "BS_HBA1C"))
```

```
    }
    if (item=="smoking") {
        varnames_cont <- setdiff(varnames_cont,
c("SMOKE_YEARS", "NUM_CIGSM", "NUM_CIGSH", "NUM_
CIGARS", "NUM_PIPE",  "NUM_OTHER"))
    }
    if (item=="sampling") {
        data <- data[, c("COUNTRY", "SURVEY",
varnames_cont)]
    }
    else {
       data <- data[, c("COUNTRY", "SURVEY", "EXAM",
"QUEST", varnames_cont)]
   }
    countries <- sort(unique(data$COUNTRY))
    for (name in varnames_cont) {
        if (variables$IRRELEVANT[variables$VARNAME==na
me] != "") {
            irr = variables$IRRELEVANT[variables$VARNA
ME==name]
        }
        else if (name %in% c('EMPLOYMENT', 'MEDICINE_
HBP', 'MEDICINE_DIAB', 'MEDICINE_CHOL', 'SMOKE_CIGSM',
'SMOKE_CIGSH', 'SMOKE_CIGARS', 'SMOKE_PIPE', 'SMOKE_
OTHER', 'NUM_CIGSM', 'NUM_CIGSH', 'NUM_CIGARS', 'NUM_
PIPE', 'NUM_OTHER', 'SMOKE_YEARS', 'SMOKE_EVER')) {
            irr = variables$IRRELEVANT[variables$VARNA
ME==name]
        }
        else {
            irr = NA
        }
        mis = variables$MISSING[variables$VARNAME==na
me]
        notknown = setdiff(variables$DONTKNOW[variable
s$VARNAME==name], irr)
        ## finding right way to filter the participants
        if (variables$EXAM[variables$VARNAME==na
me]=="1" & variables$QUEST[variables$VARNAME==na
me]=="1") {
            filters <-  expression(EXAM == '1' | QUEST
== '1')
        }
        else if (variables$EXAM[variables$VARNAME==na
me]=="1") {
            filters <-  expression(EXAM == '1')
        }
        else if (variables$QUEST[variables$VARNAME==na
me]=="1") {
            filters <-  expression(QUEST == '1')
```

```
        }
        else {
            filters <- TRUE
        }
        for (country in countries) {
            surveys <- sort(unique(data[data$COUNTRY
== country,]$SURVEY))
            for (survey in surveys) {
                if (item == "sampling") {
                    selected <- data[data$COUNTRY
== country & data$SURVEY == survey , c("COUNTRY",
"SURVEY", name)]
                }
                else {
                     selected <- data[data$COUNTRY
== country & data$SURVEY == survey , c("COUNTRY",
"SURVEY", "QUEST", "EXAM", name)]
                }
                attach(selected)
                selected = selected[eval(filters) , ]
                detach(selected)
                ### number of relevants
                if (is.na(irr)) {
                    nrel <- nrow(selected)
                }
                else {
                    nrel <- nrow(selected[
!(selected[, name] %in% irr) , ])
                }
                ### number of irrelevants
                if (is.na(irr)) {
                    nirr <- NA
                }
                else {
                    nirr <- nrow(selected[  selected[,
name] %in% irr , ])
                }
                ### putting to the table
                if (is.null(tab)) {
                    tab <- as.data.
frame(cbind(country, survey, "n_relevant"),
stringsAsFactors  = F )
                    tab[,dim(tab)[2]+1] <- nrel
                    names(tab) <- c("COUNTRY",
"SURVEY", "FIELD", name)
                }
                else {
                    if ( nrow(tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="n_relevant",]) != 0
) {
```

```
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_relevant", name] <-
nrel
                    }
                    else {
                        tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"n_relevant")
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_relevant", name] <-
nrel
                    }
                }
                if ( nrow(tab[tab$COUNTRY == country &
tab$SURVEY== survey & tab$FIELD== "n_irrelevant", ])
== 0 ) {
                    tab[dim(tab)[1]+1 , c("COUNTRY",
"SURVEY", "FIELD")] <- c(country, survey, "n_
irrelevant")
                    tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_irrelevant", name]
<- nirr
                }
                else  {
                    tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_irrelevant", name]
<- nirr
                }
                if (nrel != 0) {
                    ### Number of missing values
                    if (variables$EXAM[variables$VARNA
ME==name]=="1" & variables$QUEST[variables$VARNAME==na
me]=="1") {
                        nmis <- nrow(selected[
(selected$QUEST=="1" & selected$EXAM=="1") &
selected[, name] %in% mis , ])
                    }
                    else if (variables$EXAM[variables$
VARNAME==name]=="1" ) {
                        nmis <- nrow(selected[  (
selected$EXAM=="1") & selected[, name] %in% mis , ])
                    }
                    else if (variables$QUEST[variables
$VARNAME==name]=="1" ) {
                        nmis <- nrow(selected[  (
selected$QUEST=="1") & selected[, name] %in% mis , ])
                    }
                    else {
                        nmis <- nrow(selected[
selected[, name] %in% mis , ])
```

```
                    }
                    ### Number of nonmissing values
                    if (variables$EXAM[variables$VARNA
ME==name]=="1" & variables$QUEST[variables$VARNAME==na
me]=="1") {
                        nnonmis <- nrow(selected[
(selected$QUEST=="1" & selected$EXAM=="1") &
!(selected[, name] %in% mis) & !(selected[, name] %in%
irr)  & !(selected[, name] %in% notknown), ])
                    }
                    else if (variables$EXAM[variables$
VARNAME==name]=="1" ) {
                        nnonmis <- nrow(selected[  (
selected$EXAM=="1") & !(selected[, name] %in% mis) &
!(selected[, name] %in% irr)  & !(selected[, name]
%in% notknown) , ])
                    }
                    else if (variables$QUEST[variables
$VARNAME==name]=="1" ) {
                        nnonmis <- nrow(selected[  (
selected$QUEST=="1") & !(selected[, name] %in% mis) &
!(selected[, name] %in% irr)   & !(selected[, name]
%in% notknown), ])
                    }
                    else {
                        nnonmis <- nrow(selected[
!(selected[, name] %in% mis) & !(selected[, name] %in%
irr)  & !(selected[, name] %in% notknown) , ])
                    }
                    ### putting to the table
                    if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== "n_
nonmissing", ]) == 0 ) {
                        tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"n_nonmissing")
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_nonmissing", name]
<- nnonmis
                    }
                    else  {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_nonmissing", name]
<- nnonmis
                    }
                    if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== "n_
missing", ]) == 0 ) {
                        tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
```

```
"n_missing")
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_missing", name] <-
nmis
                    }
                    else  {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_missing", name] <-
nmis
                    }
                    ### proportion of missing data
                    propmis <- 100* signif( (nmis) /
nrel , 2)
                    nnonmis = NULL
                    nmis = NULL
                    nirr = NULL
                    ## putting to the table
                    if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== "missing_
percent", ]) == 0 ) {
                        tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"missing_percent")
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="missing_percent",
name] <- propmis
                    }
                    else  {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="missing_percent",
name] <- propmis
                    }
                    #### proportion of dont knows
                    if (is.na(notknown)) {
                    }
                    else if (sum( notknown != "") !=
0) {
                        nnot <- nrow(selected[
selected[, name] %in% notknown, ])
                        propnot <- 100* signif( (nnot)
/ nrel , 2)
                        nnot = NULL
                        if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== "dontknow_
percent", ]) == 0 ) {
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"dontknow_percent")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="dontknow_percent",
```

```
name] <- propnot
                        }
                        else  {
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="dontknow_percent",
name] <- propnot
                        }
                    }
                    ### removing irrelevants
                    if (variables$EXAM[variables$VARNA
ME==name]=="1" & variables$QUEST[variables$VARNAME==na
me]=="1") {
                        selected <- selected[
(selected$QUEST=="1" & selected$EXAM=="1") &
!(selected[, name] %in% irr), ]
                    }
                    else if (variables$EXAM[variables$
VARNAME==name]=="1" ) {
                        selected <- selected[
(selected$EXAM=="1") & !(selected[, name] %in% irr), ]
                    }
                    else if (variables$QUEST[variables
$VARNAME==name]=="1" ) {
                        selected <- selected[
(selected$QUEST=="1") & !(selected[, name] %in% irr),
]
                    }
                    else {
                        selected <- selected[
!(selected[, name] %in% irr), ]
                    }
                    ### changing to numeric
                    selected[ selected[, name] %in%
c(mis, notknown) , name] <- NA
                    selected[, name] <-
as.numeric(selected[, name])
                    ### 5 minimum  & maximum values
                    small <- sort(selected[, name])
[1:5]
                    large <- sort(selected[, name],
decreasing = T)[1:5]
                    for (k in 1:5) {
                        if ( nrow(tab[tab$COUNTRY
== country & tab$SURVEY== survey & tab$FIELD==
paste("min", k, sep=""), ]) == 0 ) {
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
paste("min", k, sep=""))
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==paste("min", k,
```

```
sep=””), name] <- small[k]
                        }
                        else  {
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==paste(”min”, k,
sep=””), name] <- small[k]
                        }
                        if ( nrow(tab[tab$COUNTRY
== country & tab$SURVEY== survey & tab$FIELD==
paste(”max”, k, sep=””), ]) == 0 ) {
                            tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
paste(”max”, k, sep=””))
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==paste(”max”, k,
sep=””), name] <- large[k]
                        }
                        else  {
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==paste(”max”, k,
sep=””), name] <- large[k]
                        }
                    }
                    small <- NULL
                    large = NULL
                    ### percentiles
                    quantiles <- quantile(selected[,
name],  probs = c(0.1, 0.5, 0.9), na.rm=T)
                    p <- c(10, 50, 90)
                    for (k in 1:3) {
                        if ( nrow(tab[tab$COUNTRY
== country & tab$SURVEY== survey & tab$FIELD==
paste(”percentile”, p[k], sep=””), ]) == 0 ) {
                            tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
paste(”percentile”, p[k], sep=””))
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==paste(”percentile”,
p[k], sep=””), name] <- quantiles[k]
                        }
                        else  {
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==paste(”percentile”,
p[k], sep=””), name] <- quantiles[k]
                        }
                    }
                    quantiles <- NULL

                    ## mean and SD
                    m <- mean(selected[, name],
```

```
na.rm=T)
                    std <- sd(selected[, name],
na.rm=T)
                    if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== ”mean”, ])
== 0 ) {
                        tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”mean”)
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==”mean”, name] <- m
                    }
                    else  {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==”mean”, name] <- m
                    }
                    if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== ”std”, ])
== 0 ) {
                        tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”std”)
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==”std”, name] <- std
                    }
                    else  {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==”std”, name] <- std
                    }
                    m=NULL
                    std=NULL

                    # proportion of odd readings
                    if (grepl(”BP_”, name, ignore.case
= T)) {
                        selected$odd[
as.numeric(selected[, name]) %% 2 != 0] <- 1
                        odd <- 100*
round((sum(selected$odd, na.rm=T) / nrel), 3)
                        if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== ”odd_
percent”, ]) == 0 ) {
                            tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”odd_percent”)
                            tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==”odd_percent”, name]
<- odd
                        }
                        else  {
```

```
                            tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="odd_percent", name]
<- odd
                        }
                        odd = NULL
                    }

                    if (item != "sampling") {
                    ## terminal digits, last digit
                        if (name %in% c("HEIGHT",
"WAIST")) {
                            selected[, name] =
format(round(selected[, name],1),  nsmall=1, trim=T)
                        }
                        else if (name == "WEIGHT") {
                            selected[, name] =
format(round(selected[, name],2),  nsmall=2, trim=T)
                        }
                        else {
                            selected[, name] =
format(selected[, name] )
                        }
                        l0 <-  100* (round( sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == "0" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                        l1 <-   100* (round( sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == "1" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                        l2 <-   100* (round( sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == "2" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                        l3 <-  100* (round(  sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == "3" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                        l4 <-  100* (round(  sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == "4" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                        l5 <-  100* (round(  sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == "5" ,na.rm=T)  /
sum(selected[, name] != "NA") , 3))
                        l6 <-  100* (round(  sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == "6" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
```

```
                        l7 <- 100* (round(   sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == ”7” ,na.rm=T) /
sum(selected[, name] != ”NA”) , 3))
                        l8 <-  100* (round(  sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == ”8” ,na.rm=T) /
sum(selected[, name] != ”NA”) , 3))
                        l9 <-  100* (round(  sum(
substr(selected[, name], nchar(selected[, name])
, nchar(selected[, name])) == ”9” ,na.rm=T) /
sum(selected[, name] != ”NA”) , 3))
                        if ( nrow(tab[tab$COUNTRY
== country & tab$SURVEY== survey & tab$FIELD==
”terminaldigit_0”, ]) == 0 ) {
                            tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”terminaldigit_0”)
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==”terminaldigit_0”,
name] <- l0
                            tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”terminaldigit_1”)
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==”terminaldigit_1”,
name] <- l1
                            tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”terminaldigit_2”)
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==”terminaldigit_2”,
name] <- l2
                            tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”terminaldigit_3”)
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==”terminaldigit_3”,
name] <- l3
                            tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”terminaldigit_4”)
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==”terminaldigit_4”,
name] <- l4
                            tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”terminaldigit_5”)
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD==”terminaldigit_5”,
```

```
name] <- l5
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_6")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_6",
name] <- l6
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_7")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_7",
name] <- l7
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_8")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_8",
name] <- l8
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_9")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_9",
name] <- l9
                        }
                        else  {
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_0",
name] <- l0
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_1",
name] <- l1
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_2",
name] <- l2
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_3",
name] <- l3
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_4",
name] <- l4
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_5",
name] <- l5
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_6",
name] <- l6
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_7",
```

```
name] <- l7
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_8",
name] <- l8
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_9",
name] <- l9
                        }
                        ## terminal digits, second
last digit
                        if (name %in% c("HEIGHT",
"WAIST")) {
                              l20 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
2 , nchar(selected[, name])-2) == "0" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l21 <-   100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
2 , nchar(selected[, name])-2) == "1" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l22 <-   100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
2 , nchar(selected[, name])-2) == "2" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l23 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
2 , nchar(selected[, name])-2) == "3" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l24 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
2 , nchar(selected[, name])-2) == "4" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l25 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
2 , nchar(selected[, name])-2) == "5" ,na.rm=T)  /
sum(selected[, name] != "NA") , 3))
                              l26 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
2 , nchar(selected[, name])-2) == "6" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l27 <- 100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
2 , nchar(selected[, name])-2) == "7" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l28 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
2 , nchar(selected[, name])-2) == "8" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l29 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
```

```
2 , nchar(selected[, name])-2) == "9" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                        }
                        else {
                              l20 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "0" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l21 <-   100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "1" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l22 <-   100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "2" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l23 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "3" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l24 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "4" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l25 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "5" ,na.rm=T)  /
sum(selected[, name] != "NA") , 3))
                              l26 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "6" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l27 <- 100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "7" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l28 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "8" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                              l29 <-  100* (round(
sum( substr(selected[, name], nchar(selected[, name])-
1 , nchar(selected[, name])-1) == "9" ,na.rm=T) /
sum(selected[, name] != "NA") , 3))
                        }
                        if ( nrow(tab[tab$COUNTRY
== country & tab$SURVEY== survey & tab$FIELD==
"terminaldigit_second_0", ]) == 0 ) {
                           tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
```

```
"terminaldigit_second_0")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_0", name] <- l20
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_second_1")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_1", name] <- l21
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_second_2")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_2", name] <- l22
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_second_3")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_3", name] <- l23
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_second_4")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_4", name] <- l24
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_second_5")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_5", name] <- l25
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_second_6")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_6", name] <- l26
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_second_7")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_7", name] <- l27
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_second_8")
                            tab[tab$COUNTRY==country
```

```
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_8", name] <- l28
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminaldigit_second_9")
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_9", name] <- l29
                        }
                        else  {
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_0", name] <- l20
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_1", name] <- l21
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_2", name] <- l22
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_3", name] <- l23
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_4", name] <- l24
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_5", name] <- l25
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_6", name] <- l26
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_7", name] <- l27
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_8", name] <- l28
                            tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="terminaldigit_
second_9", name] <- l29
                        }
                        # proportion of even values
                            even <-
sum(c(l0,l2,l4,l6,l8),na.rm=T )
                            if ( nrow(tab[tab$COUNTRY
== country & tab$SURVEY== survey & tab$FIELD==
"terminal_even_percent", ]) == 0 ) {
                                tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminal_even_percent")
```

```
tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$FIELD=="terminal_even_percent", name] <- even
                            }
                            else  {

tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$FIELD=="terminal_even_percent", name] <- even
                            }
                            even = NULL
                             even2 <-
sum(c(l20,l22,l24,l26,l28),na.rm=T )
                            if ( nrow(tab[tab$COUNTRY
== country & tab$SURVEY== survey & tab$FIELD==
"terminalsecond_even_percent", ]) == 0 ) {
                                tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"terminalsecond_even_percent")

tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$FIELD=="terminalsecond_even_percent", name] <-
even2
                            }
                            else  {

tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$FIELD=="terminalsecond_even_percent", name] <-
even2
                            }
                            even2 = NULL
                    }
                 }
            }
        }
    }
    return(tab)
}
```

plotdist - to plot simple distribution plot for continuous variables (png file)

```
plotdist <- function(data, varname, outpath, div, minim, maxim,
mis=NA, irr=NA, axmin=minim, axmax=maxim) {
    ### data = dataset, varname = name of the variable, outpath =
path to save the file, minim = lower limit of the variable, maxim
= upper limit of the variable,
    ### mis = code for missing value, irr = code for irrelevants,
axmin = minimum of the x-axis, axmax = minimum of the x-axis
    return=T
    variable <-data[, varname]
    novalue <- (variable %in% mis)
    totallymis <-  (variable %in% irr)
```

```
    if (sum(totallymis) + sum(novalue) == length(variable)) {
        return = FALSE
    }
    variable <- as.numeric(variable[!novalue & !totallymis])
    if (sum(variable >= minim & variable <= maxim) == 0) {
        return = FALSE
    }
    counts <- table(variable[variable >= minim & variable <=
maxim])
    classes <- as.numeric(names(counts))
    divpoints <- ((classes %% div) == 0)
    if (return) {
        png(file = paste(outpath,  tolower(data$COUNTRY[1]),
data$SURVEY[1], '_', tolower(varname), '.png', sep=''),
width=800, height=600)

        plot(classes, counts, type='h', axes=FALSE,
main=paste(varname, ' (', data$COUNTRY[1], 'survey',
data$SURVEY[1], ')'), xlab=varname, ylab='N', xlim=c(minim,
maxim), ylim=c(0, max(counts)))


        points(classes, counts, cex=0.6)

        # Number of outliers in the right:
        outl <- sum(variable > maxim)
        if (outl > 0)
            mtext(paste(outl, ' obs.\n>', maxim, sep=''), side=1,
line=3, at=maxim, cex=0.9)
        # Number of outliers in the left:
        outl <- sum(variable < minim)
        if (outl > 0)
            mtext(paste(outl, ' obs.\n<', minim, sep=''), side=1,
line=3, at=minim, cex=0.9)
        ## Number of observations and missing values
        mtext(paste('n_relevant = ',length(variable) +
sum(novalue, na.rm=T), sep=''), side=3, line=-2, cex=0.9, adj=1)
        mtext(paste(sum(novalue), ' missing', sep=''), side=3,
line=-3,  cex=0.9, adj=1)
        box()
        axis(1, at=seq(axmin, axmax, by=div), labels=seq(axmin,
axmax, by=div), cex.axis=0.9)
        axis(2, cex.axis=0.9)
        mtext(paste('Updated on', Sys.Date()), side=3, line=0.3,
adj=1, cex=0.9)
        dev.off()
    }
    variable = NULL
}
```

create_bpdiff - to calculate differences between three blood pressure measurements

```
### Function to create differences between BP
measurements
create_bpdiff <- function(data, variables)  {
      ### data = dataset, variables = definition table
for variables
```

```
     mis = variables$MISSING[variables$VARNAME=="BP_
SYST1"]
     irr = variables$IRRELEVANT[variables$VARNAME=="
BP_SYST1"]
    ### Differences between BP measurements
    data$SBP1_SBP2[!(data$BP_SYST1 %in% mis) &
!(data$BP_SYST2 %in% mis)] <- as.numeric(data$BP_
SYST1[!(data$BP_SYST1 %in% mis) & !(data$BP_SYST2 %in%
mis)]) - as.numeric(data$BP_SYST2[!(data$BP_SYST1 %in%
mis) & !(data$BP_SYST2 %in% mis)])
    data$SBP1_SBP2[(data$BP_SYST1 %in% mis) |
(data$BP_SYST2 %in% mis)]   <- mis
    data$SBP1_SBP2[(data$BP_SYST1 %in% irr) |
(data$BP_SYST2 %in% irr)]   <- irr

    data$SBP2_SBP3[!(data$BP_SYST2 %in% mis) &
!(data$BP_SYST3 %in% mis)] <- as.numeric(data$BP_
SYST2[!(data$BP_SYST2 %in% mis) & !(data$BP_SYST3 %in%
mis)]) - as.numeric(data$BP_SYST3[!(data$BP_SYST2 %in%
mis) & !(data$BP_SYST3 %in% mis)])
    data$SBP2_SBP3[(data$BP_SYST2 %in% mis) |
(data$BP_SYST3 %in% mis)]   <- mis
    data$SBP2_SBP3[(data$BP_SYST2 %in% irr) |
(data$BP_SYST3 %in% irr)]   <- irr

    data$SBP1_SBP3[!(data$BP_SYST1 %in% mis) &
!(data$BP_SYST3 %in% mis)] <- as.numeric(data$BP_
SYST1[!(data$BP_SYST1 %in% mis) & !(data$BP_SYST3 %in%
mis)]) - as.numeric(data$BP_SYST3[!(data$BP_SYST1 %in%
mis) & !(data$BP_SYST3 %in% mis)])
    data$SBP1_SBP3[(data$BP_SYST1 %in% mis) |
(data$BP_SYST3 %in% mis)]   <- mis
    data$SBP1_SBP3[(data$BP_SYST1 %in% irr) |
(data$BP_SYST3 %in% irr)]   <- irr

    data$DBP1_DBP2[!(data$BP_DIAST1 %in% mis) &
!(data$BP_DIAST2 %in% mis)] <- as.numeric(data$BP_
DIAST1[!(data$BP_DIAST1 %in% mis) & !(data$BP_DIAST2
%in% mis)]) - as.numeric(data$BP_DIAST2[!(data$BP_
DIAST1 %in% mis) & !(data$BP_DIAST2 %in% mis)])
    data$DBP1_DBP2[(data$BP_DIAST1 %in% mis) |
(data$BP_DIAST2 %in% mis)]   <- mis
    data$DBP1_DBP2[(data$BP_DIAST1 %in% irr) |
(data$BP_DIAST2 %in% irr)]   <- irr

    data$DBP2_DBP3[!(data$BP_DIAST2 %in% mis) &
!(data$BP_DIAST3 %in% mis)] <- as.numeric(data$BP_
DIAST2[!(data$BP_DIAST2 %in% mis) & !(data$BP_DIAST3
%in% mis)]) - as.numeric(data$BP_DIAST3[!(data$BP_
DIAST2 %in% mis) & !(data$BP_DIAST3 %in% mis)])
```

```
    data$DBP2_DBP3[(data$BP_DIAST2 %in% mis) |
(data$BP_DIAST3 %in% mis)]   <- mis
    data$DBP2_DBP3[(data$BP_DIAST2 %in% irr) |
(data$BP_DIAST3 %in% irr)]   <- irr

    data$DBP1_DBP3[!(data$BP_DIAST1 %in% mis) &
!(data$BP_DIAST3 %in% mis)] <- as.numeric(data$BP_
DIAST1[!(data$BP_DIAST1 %in% mis) & !(data$BP_DIAST3
%in% mis)]) - as.numeric(data$BP_DIAST3[!(data$BP_
DIAST1 %in% mis) & !(data$BP_DIAST3 %in% mis)])
    data$DBP1_DBP3[(data$BP_DIAST1 %in% mis) |
(data$BP_DIAST3 %in% mis)]   <- mis
    data$DBP1_DBP3[(data$BP_DIAST1 %in% irr) |
(data$BP_DIAST3 %in% irr)]   <- irr

    data$DBP1_DBP2_DBP3 <- 0
    data$DBP1_DBP2_DBP3[!(data$BP_DIAST1 %in% mis) &
!(data$BP_DIAST2 %in% mis) & !(data$BP_DIAST3 %in%
mis) & data$BP_DIAST1 == data$BP_DIAST2 & data$BP_
DIAST1 == data$BP_DIAST3] <- 1
    data$DBP1_DBP2_DBP3[(data$BP_DIAST1 %in% mis) |
(data$BP_DIAST2 %in% mis) | (data$BP_DIAST3 %in% mis)]
<- mis
    data$DBP1_DBP2_DBP3[(data$BP_DIAST1 %in% irr) |
(data$BP_DIAST2 %in% irr) | (data$BP_DIAST3 %in% irr)]
<- irr


    data$SBP1_SBP2_SBP3 <- 0
    data$SBP1_SBP2_SBP3[!(data$BP_SYST1 %in% mis) &
!(data$BP_SYST2 %in% mis) & !(data$BP_SYST3 %in% mis)
& data$BP_SYST1 == data$BP_SYST2 & data$BP_SYST1 ==
data$BP_SYST3] <- 1
    data$SBP1_SBP2_SBP3[(data$BP_SYST1 %in% mis) |
(data$BP_SYST2 %in% mis) | (data$BP_SYST3 %in% mis)]
<- mis
    return(data)
}
```

create_bsconv - to calculate unit conversions of lipid and glucose measurements

```
create_bsconv <- function(data, variables)  {
    ### data = dataset, variables = definition table
for variables
     mis = variables$MISSING[variables$VARNAME=="BS_
CHOL"]
     irr = variables$IRRELEVANT[variables$VARNAME=="
BS_CHOL"]
    ## Separating different units
```

```
    data$BS_CHOL_MMOLL[data$BS_UNIT_CHOL == "1" ] <-
data$BS_CHOL[data$BS_UNIT_CHOL == "1" ]
    data$BS_CHOL_MMOLL[data$BS_UNIT_CHOL != "1" ]   <-
"."
    data$BS_CHOL_MMOLL[data$BS_CHOL == mis ] <- mis
    data$BS_CHOL_MGDL[data$BS_UNIT_CHOL != "2"]
<-"."
    data$BS_CHOL_MGDL[data$BS_UNIT_CHOL == "2" ] <-
data$BS_CHOL[data$BS_UNIT_CHOL == "2" ]
    data$BS_CHOL_MGDL[data$BS_CHOL == mis ] <- mis
    data$BS_HDL_MMOLL[data$BS_UNIT_HDL != "1" ]   <-
"."
    data$BS_HDL_MMOLL[data$BS_UNIT_HDL == "1" ] <-
data$BS_HDL[data$BS_UNIT_HDL == "1" ]
    data$BS_HDL_MMOLL[data$BS_HDL == mis ] <- mis
    data$BS_HDL_MGDL[data$BS_UNIT_HDL != "2"]   <- "."
    data$BS_HDL_MGDL[data$BS_UNIT_HDL == "2" ] <-
data$BS_HDL[data$BS_UNIT_HDL == "2" ]
    data$BS_HDL_MGDL[data$BS_HDL == mis ] <- mis
    data$BS_GLC_MMOLL[data$BS_UNIT_GLC != "1" ]   <-
"."
    data$BS_GLC_MMOLL[data$BS_UNIT_GLC == "1" ] <-
data$BS_GLC[data$BS_UNIT_GLC == "1" ]
    data$BS_GLC_MMOLL[data$BS_GLC == mis ] <- mis
    data$BS_GLC_MGDL[data$BS_UNIT_GLC != "2"]   <- "."
    data$BS_GLC_MGDL[data$BS_UNIT_GLC == "2" ] <-
data$BS_GLC[data$BS_UNIT_GLC == "2" ]
    data$BS_GLC_MGDL[data$BS_GLC == mis ] <- mis
    data$BS_HBA1C_MMOLMOL[data$BS_UNIT_HBA1C != "1" ]
<- "."
    data$BS_HBA1C_MMOLMOL[data$BS_UNIT_HBA1C == "1" ]
<- data$BS_HBA1C[data$BS_UNIT_HBA1C == "1" ]
    data$BS_HBA1C_MMOLMOL[data$BS_HBA1C == mis ] <-
mis
    data$BS_HBA1C_PERCENT[data$BS_UNIT_HBA1C  != "2"]
<- "."
    data$BS_HBA1C_PERCENT[data$BS_UNIT_HBA1C  == "2" ]
<- data$BS_HBA1C[data$BS_UNIT_HBA1C == "2" ]
    data$BS_HBA1C_PERCENT[data$BS_HBA1C == mis ] <-
mis

    return(data)
}
```

drawtable_bpqa - to derive specific quality assessment checks for blood pressure (proportion of incomplete data, proportion of identical measurements)

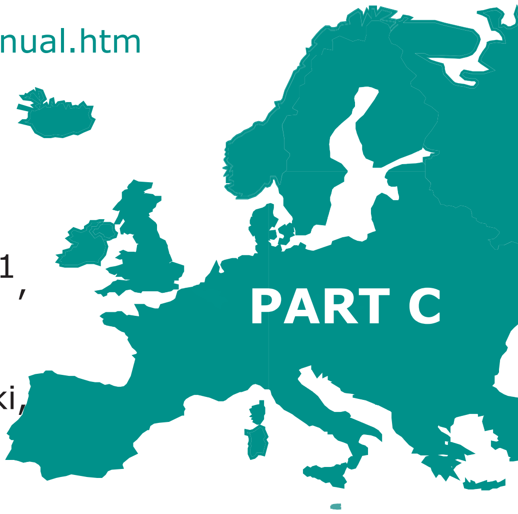
```
drawtable_bpqa <- function(data, variables)  {
    ### data = dataset, variables = definition table
```

```
for variables
    # tab, output table
    tab <- NULL
    variables <- variables[grepl("BP_SY",
variables$VARNAME, ignore.case = TRUE) | grepl("BP_
DI", variables$VARNAME, ignore.case = TRUE),]
    variables[is.na(variables)] = ""
    data <- data[ , c("COUNTRY", "SURVEY", "EXAM",
variables$VARNAME)]
    countries <- sort(unique(data$COUNTRY))
    mis = variables$MISSING[variables$VARNAME=="BP_
SYST1"]
    irr = variables$IRRELEVANT[variables$VARNAME=="
BP_SYST1"]
    data = create_bpdiff(data, variables)
    varnames =  c("SBP1_SBP2", "SBP2_SBP3", "SBP1_
SBP3" ,"DBP1_DBP2" ,"DBP2_DBP3", "DBP1_DBP3", "SBP1_
SBP2_SBP3", "DBP1_DBP2_DBP3")
    for (name in varnames) {
        for (country in countries) {
            surveys <- sort(unique(data[data$COUNTRY
== country,]$SURVEY))
            for (survey in surveys) {
                selected <- data[data$COUNTRY ==
country & data$SURVEY == survey , c("COUNTRY",
"SURVEY", "EXAM", name)]
                ### number of non-irrelevants
                nrel <- nrow(selected[  (
selected$EXAM=="1") & !(selected[, name] %in% irr) ,
])
                ### number of irrelevants
                nirr <- nrow(selected[  (
selected$EXAM=="1") & selected[, name] %in% irr,])
                ### putting to the table
                if (is.null(tab)) {
                    tab <- as.data.
frame(cbind(country, survey, "n_relevant"),
stringsAsFactors  = F )
                    tab[,dim(tab)[2]+1] <- nrel
                    names(tab) <- c("COUNTRY",
"SURVEY", "FIELD", name)
                }
                else {
                    if ( nrow(tab[tab$COUNTRY==country
& tab$SURVEY==survey & tab$FIELD=="n_relevant",]) != 0
) {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_relevant", name] <-
nrel
                    }
```

```
                    else {
                        tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"n_relevant")
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_relevant", name] <-
nrel
                    }
                }
                if ( nrow(tab[tab$COUNTRY == country &
tab$SURVEY== survey & tab$FIELD== "n_irrelevant", ])
== 0 ) {
                    tab[dim(tab)[1]+1 , c("COUNTRY",
"SURVEY", "FIELD")] <- c(country, survey, "n_
irrelevant")
                    tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_irrelevant", name]
<- nirr
                }
                else  {
                    tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_irrelevant", name]
<- nirr
                }
                if (nrel != 0) {
                    ### Number of missing values
                    nmis <- nrow(selected[   (
selected$EXAM=="1") & selected[, name] %in% mis , ])

                    ### Number of nonmissing values
                    nnonmis <- nrow(selected[   (
selected$EXAM=="1") & !(selected[, name] %in% mis) &
!(selected[, name] %in% irr)   , ])

                    ### putting to the table
                    if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== "n_
nonmissing", ]) == 0 ) {
                        tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"n_nonmissing")
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_nonmissing", name]
<- nnonmis
                    }
                    else  {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="n_nonmissing", name]
<- nnonmis
                    }
```

```
                    if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== ”n_
missing”, ]) == 0 ) {
                        tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”n_missing”)
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==”n_missing”, name] <-
nmis
                    }
                    else  {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==”n_missing”, name] <-
nmis
                    }
                    ### proportion of missing data
                    propmis <- 100* signif( (nmis) /
nrel , 2)
                    nnonmis = NULL
                    nmis = NULL
                    nirr = NULL
                    ## putting to the table
                    if ( nrow(tab[tab$COUNTRY ==
country & tab$SURVEY== survey & tab$FIELD== ”missing_
percent”, ]) == 0 ) {
                        tab[dim(tab)[1]+1 ,
c(”COUNTRY”, ”SURVEY”, ”FIELD”)] <- c(country, survey,
”missing_percent”)
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==”missing_percent”,
name] <- propmis
                    }
                    else  {
                        tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD==”missing_percent”,
name] <- propmis

                    }
                    ### removing irrelevants
                    selected <- selected[
(selected$EXAM==”1”) & !(selected[, name] %in% irr), ]
                    if (!(name %in% c(”SBP1_SBP2_
SBP3”, ”DBP1_DBP2_DBP3”))) {
                        ### changing to numeric
                        selected[ selected[, name]
%in% mis , name] <- NA
                        selected[, name] <-
as.numeric(selected[, name])
                        #proportion of identical
results
```

```
                        ident <- 100*round(nrow(select
ed[selected[,name] == 0, ]  ) / nrel, 3)
                       if ( nrow(tab[tab$COUNTRY
== country & tab$SURVEY== survey & tab$FIELD==
"identical_percent", ]) == 0 ) {
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"identical_percent")
                            tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="identical_percent",
name] <- ident
                        }
                        else  {
                            tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="identical_percent",
name] <- ident
                        }
                     }
                     else {
                        #proportion of identical
results
                        ident <- 100*round(nrow(select
ed[selected[,name] == 1, ]  ) / nrel, 3)
                       if ( nrow(tab[tab$COUNTRY
== country & tab$SURVEY== survey & tab$FIELD==
"identical_percent", ]) == 0 ) {
                            tab[dim(tab)[1]+1 ,
c("COUNTRY", "SURVEY", "FIELD")] <- c(country, survey,
"identical_percent")
                            tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="identical_percent",
name] <- ident
                        }
                        else  {
                            tab[tab$COUNTRY==country &
tab$SURVEY==survey & tab$FIELD=="identical_percent",
name] <- ident
                        }
                     }
                 }
             }
        }
    }
    return(tab)
}
```



4. Indicators

Kari Kuulasmaa[1], Päivikki Koponen[1], Johanna Mäki-Opas[1], Tarja Tuovinen[1], Katri Kilpeläinen[1] and Hanna Tolonen[1]

[1]National Institute for Health and Welfare (THL), Helsinki, Finland

Following indicators are defined to be used in EHES reporting, both in quality assessments and basic reporting of the results. For more details how to calculate the indicators from EHES data see EHES Manual, Part C, Chapter 5.

Indicators which at least should be included in the basic report nationally and in the EHES basic report are marked with green tick mark (❖).

4.1 Definition of indicators

4.1.1 Blood pressure

4.1.1.1 Systolic blood pressure (mmHg)

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.1.2 Diastolic blood pressure (mmHg)

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation
- Order statistics: 5 smallest values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 largest values
- Distribution curve

- Cumulative distribution curve

4.1.1.3 Hypertension and categories of blood pressure

Prevalence of self-reported high blood pressure (hypertension) (%) ❖

- *Definition:* Proportion of those who reported that they have been diagnosed by a medical doctor to have a high blood pressure (hypertension) among all survey participants
- *Comment:* A subjective indicator from questionnaire data.

Prevalence of actual and potential hypertensives (%) ❖

- *Definition:* Proportion of those whose systolic blood pressure was at least 140 mmHg or diastolic blood pressure was at least 90 mmHg or who reported that they are taking medication to lower their blood pressure among all survey participants.
- *Comment:* Serves as a proxy for the prevalence of hypertension in the population, whether diagnosed or undiagnosed. An objective indicator based on measured blood pressure.

Prevalence of elevated blood pressure (%)

- *Definition:* Proportion of those whose systolic blood pressure was at least 140 mmHg or diastolic blood pressure was at least 90 mmHg among all survey participants.

Prevalence of elevated systolic blood pressure (%)

- *Definition:* Proportion of those whose systolic blood pressure was at least 140 mmHg and diastolic blood pressure was less than 90 mmHg among all survey participants. (Graham, Atar et al. 2007)

Distribution of categories of blood pressure (%)

- *Definition:* Proportion of categories of blood pressure in the population (%) (Graham , Atar et al. 2007)

- Optimal if systolic blood pressure was less than 120 mmHg and diastolic blood pressure was less than 80 mmHg
- Normal if systolic blood pressure was 120-129 mmHg and/or diastolic blood pressure was 80-84 mmHg and neither of these is above the upper limit
- High normal if systolic blood pressure was 130-139 mmHg and/or diastolic blood pressure was 85-89 mmHg and neither of these is above the upper limit
- Grade 1 hypertension if systolic blood pressure was 140-159 mmHg and/or diastolic blood pressure was 90-99 mmHg and neither of these is above the upper limit
- Grade 2 hypertension if systolic blood pressure was 160-179 mmHg and/or diastolic blood pressure was 100-109 mmHg and neither of these is above the upper limit
- Grade 3 hypertension if systolic blood pressure was at least 180 mmHg and/or diastolic blood pressure was at least 110 mmHg

4.1.1.4 Awareness of elevated blood pressure

Awareness of elevated blood pressure among actual and potential hypertensives (%) ❖

- *Definition:* Proportion of those who reported that they have been diagnosed by a medical doctor to have a high blood pressure (hypertension) among those defined as having actual or potential hypertension.

4.1.1.5 Anti-hypertensive drug use

Prevalence of anti-hypertensive drug use in the population (%) ❖

- *Definition:* Proportion of those who reported that they are taking medication to lower their blood pressure among all survey participants.

Prevalence of anti-hypertensive drug use among actual and potential hypertensives (%) ❖

- *Definition:* Proportion of those who reported that they are taking medication to lower their blood pressure among those defined as having actual or potential hypertension.

Prevalence of optimal blood pressure among those using anti-hypertensive drugs (%) ❖

- *Definition:* Proportion of those whose systolic blood pressure is below 140 mmHg and diastolic blood pressure below 90 mmHg among those who report taking medication to lower their blood pressure.

4.1.1.6 Blood pressure measurement

Proportion of the population with blood pressure measurement in the past 12 months (%)

- *Definition:* Proportion of those who reported that their blood pressure was measured by a health professional in the past 12 months among all survey participants.

Proportion of the population with blood pressure measurement in the past 5 years (%) ❖

- *Definition:* Proportion of those who reported that their blood pressure was measured by a health professional in the past 5 years among all survey participants.

4.1.1.7 Pulse rate (beats/min)

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 smallest values, 10th percentile, median, 90th percentile and 5 largest values
- Distribution curve
- Cumulative distribution curve

4.1.2 Lipids

4.1.2.1 Serum total cholesterol (mmol/l)

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation
- Order statistics: 5 minimum values, 10th percentile, 25th percentile, 50th percentile (median), 75th percentile, 90th percentile and 5 maximum values
- Distribution curve

- Cumulative distribution curve

4.1.2.2 Serum high-density lipoprotein (HDL) cholesterol (mmol/l)

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.2.3 Serum non-HDL cholesterol (mmol/l)

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.2.4 Serum total cholesterol to HDL cholesterol ratio

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.2.5 Categories of cholesterol level

Prevalence of self-reported elevated blood cholesterol (%) ❖

- *Definition:* Proportion of those who reported that they have been diagnosed by a medical doctor to have elevated blood cholesterol among all survey participants.
- *Comment:* A subjective indicator of elevated blood cholesterol.

Prevalence of elevated serum total cholesterol in the population (%)

- *Definition:* Proportion of those whose serum total cholesterol was at least 5.0 mmol/l among all survey participants.
- *Comment:* Serves as a proxy for the prevalence of elevated total serum cholesterol in the population. An objective indicator or elevated total cholesterol.

Prevalence of actual or potential elevated serum total cholesterol in the population (%) ❖

- *Definition:* Proportion of those whose serum total cholesterol was at least 5.0 mmol/l or who reported that they are taking medication to lower their blood cholesterol among all survey participants.
- *Comment:* An objective indicator or elevated total cholesterol.

Distribution of categories of serum total cholesterol (%)

- *Definition:* Proportion of categories of serum total cholesterol in population (%) among all survey participants
  - below 5.0 mmol/l
  - at least 5.0 mmol/l and below 6.0 mmol/l
  - at least 6.0 mmol/l and below 7.0 mmol/l
  - at least 7.0 mmol/l and below 8.0 mmol/l
  - at least 8.0 mmol/l

Prevalence of low HDL cholesterol (%)

- *Definition:* Proportion of those whose HDL cholesterol is lower than 1.0 mmol/l in men and lower than 1.2

mmol/l in women among all survey participants. (Grahan, Atar et al. 2007)

Prevalence of high total cholesterol to HDL cholesterol ratio (%)

- *Definition:* Proportion of those who whose serum total cholesterol to HDL cholesterol ratio is greater than 5 among all survey participants. (Graham, Atar et al 2007)

Prevalence of elevated non-HDL cholesterol (%)

- *Definition:* Proportion of those whose non-HDL cholesterol is greater than 3.5 mmol/l among all survey participants. (REF?)

Prevalence of actual or potential elevated non-HDL cholesterol in the population (%)

- *Definition:* Proportion of those whose non-HDL cholesterol is greater than 3.5 mmol/l or who reported that they are taking medications to lower their blood cholesterol among all survey participants.

4.1.2.6 Awareness of elevated total cholesterol

Awareness of elevated serum cholesterol among those with actual or potential elevated serum total cholesterol (%) ❖

- *Definition:* Proportion of those who reported that they have been diagnosed by a medical doctor to have elevated blood cholesterol among those defined as having actual or potential elevated serum total cholesterol

4.1.2.7 Lipid lowering drug use

Prevalence of lipid lowering drug use in the population (%) ❖

- *Definition:* Proportion of those who reported that they are taking medication to lower their blood cholesterol among all survey participants.

Prevalence of lipid lowering drug use among those with elevated total cholesterol (%) ❖

- *Definition:* Proportion of those who reported that they are taking medication to lower their blood cholesterol among those defined as having elevated serum total cholesterol.

Prevalence of optimal total cholesterol level among those using lipid lowering drugs (%) ❖

- *Definition:* Proportion of those whose serum total cholesterol is less than 5.0 mmol/l among those who are taking medication to lower their blood cholesterol.

4.1.2.8 Cholesterol measurement

Proportion of the population with cholesterol measurement in the past 12 months (%)

- *Definition:* Proportion of those who reported that their blood cholesterol have been measured by a health professional in the past 12 months among all survey participants.

Proportion of the population with cholesterol measurement in the past 5 years (%) ❖

- *Definition:* Proportion of those who reported that their blood cholesterol have been measured by a health professional in the past 5 years among all survey participants.

4.1.3 Glucose

4.1.3.1 Fasting plasma glucose

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation
- Order statistics: 5 minimum values, 10th percentile, 25th percentile, 50th percentile (median), 75th percentile, 90th percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.3.2 Glycated haemoglobin (HbA1c)

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curves

4.1.3.3 Diabetes and impaired fasting glucose

Prevalence of self-reported diabetes (%) ❖ (EHCI indicator name: Diabetes: self-reported prevalence)

- *Definition:* Proportion of those who reported that they have been diagnosed by a medical doctor to have a diabetes among all survey participants.
- *Comment:* A subjective indicator.

Prevalence of impaired fasting glucose (IFG) (%)

- *Definition:* Proportion of those with fasting plasma glucose more than 6.0 mmol/l but less than 7.0 mmol/l (WHO 2006) among all survey participants.

Prevalence of diabetes (%) based on fasting plasma glucose

- *Definition:* Proportion of those with fasting plasma glucose at least 7.0 mmol/l (WHO 2006) among all survey participants.

Prevalence of diabetes (%) based on HbA1c

- *Definition:* Proportion of those with NGSP HbA1c at least 6.5% (IFCC HbA1c at least 48 mmol/mol) (WHO 2011a) among all survey participants.

Prevalence of diabetes (%) based on fasting glucose or HbA1c

- *Definition:* Proportion of those with fasting plasma glucose at least 7.0 mmol/l or NGSP HbA1c at least 6.5%

(IFCC HbA1c at least 48 mmol/mol) among all survey participants.

Prevalence of diabetes (%) based on fasting glucose, HbA1c or self-reported status

- *Definition:* Proportion of those with fasting plasma glucose at least 7.0 mmol/l or NGSP HbA1c at least 6.5% (IFCC HbA1c at least 48 mmol/mol) or who reported that they have been diagnosed by a medical doctor to have diabetes among all survey participants

Prevalence of diabetes (%) based on fasting glucose, HbA1c or use of diabetes medication ❖

- *Definition:* Proportion of those with fasting plasma glucose at least 7.0 mmol/l or NGSP HbA1c at least 6.5% (IFCC HbA1c at least 48 mmol/mol) or who reported that they are taking medication for diabetes among all survey participants

4.1.3.4 Use of diabetes medication

Prevalence of diabetes medication in population (%) ❖

- *Definition:* Proportion of those who reported that they are taking medication for diabetes among all survey participants.

Prevalence of diabetes medication among actual and potential diabetics (%) ❖

- *Definition:* Proportion of those who reported that they are taking medication for diabetes among those defined as having actual or potential diabetics based on fasting glucose, HbA1c or use of diabetic medication

Prevalence of optimal glucose level among those using diabetes medication (%)

- *Definition:* Proportion of those whose blood glucose is less than 6.0 mmol/l or HbA1c is less than ??% (IFCC HbA1c less than ?? mmol/mol) among those who are taking medication for their diabetes. (IDF 2005)

4.1.3.5 Awareness of diabetes

Awareness of diabetes among diabetics (%) ❖

- *Definition:* Proportion of those who reported that they have been diagnosed by a medical doctor to have diabetes among those defined to have diabetes based fasting glucose, HbA1c or use of diabetes medication.

4.1.3.6 Glucose measurement

Prevalence of the population with blood glucose measured in the past 12 months (%)

- *Definition:* Proportion of those who reported that their blood glucose was measured by a health professional in the past 12 months among all survey participants

Prevalence of the population with blood glucose measured in the past 5 years (%) ❖

- *Definition:* Proportion of those who reported that their blood glucose was measured by a health professional in the past 5 years among all survey participants

4.1.4 Anthropometrics and obesity

4.1.4.1 Height (measured)

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10th percentile, 25th percentile, 50th percentile (median), 75th percentile, 90th percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.14.2 Height (self-reported)

- Population mean, standard error and 95 % confidence interval
- Population standard deviation

- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.4.3 Difference between measured and self-reported height

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.4.4 Weight (measured)

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.4.5 Weight (self-reported)

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.4.6 Difference between measured and self-reported weight

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.4.7 Waist circumference

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.4.8 BMI (Based on measured height and weight)

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.4.9 BMI (based on self-reported height and weight)

- Population mean, standard error and 95 % confidence interval ❖
- Population standard deviation

- Order statistics: 5 minimum values, 10th percentile, 25th percentile, 50th percentile (median), 75th percentile, 90th percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.4.10 Difference between measured and self-reported BMI

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10th percentile, 25th percentile, 50th percentile (median), 75th percentile, 90th percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.1.4.11 Obesity and other BMI categories

(WHO 1995, WHO 2000, WHO 2004)

Prevalence of overweight (%) based on measured BMI ❖

- *Definition:* Proportion of those who have BMI > 25 among all survey participants.

Prevalence of obesity (%) based on measured BMI ❖

- *Definition:* Proportion of those who have BMI > 30 among all survey participants.

Distribution of BMI categories based on measured weight and height

Category of BMI	**BMI**
underweight	< 18.50
normal range	18.50-24.99
overweight	≥ 25.00
obese	≥ 30.00
obese grade I	30.00-34.99
obese grade II	35.00-39.99
obese grade III	≥ 40.00

Prevalence of overweight (%) based on self-reported BMI

- *Definition:* Proportion of those who have BMI > 25 among all survey participants.

Prevalence of obesity (%) based on self-reported BMI

- *Definition:* Proportion of those who have BMI > 30 among all survey participants.

Distribution of BMI categories based on self-reported weight and height

Category of BMI	BMI
underweight	< 18.50
normal range	18.50-24.99
overweight	≥ 25.00
obese	≥ 30.00
obese grade I	30.00-34.99
obese grade II	35.00-39.99
obese grade III	≥ 40.00

4.1.4.12 Distribution of categories of waist circumference

Categories of waist circumference (WHO 2011b)

	Men	Women
Waist circumference (cm)	> 102 cm	> 88 cm

4.1.5 Self-perceived health and other self-reported chronic diseases

Self-perceived health ❖

- *Definition:* Proportion of those who reported that their health is (very) good among all survey participants. (ECHI)
- *Comment:* Part of the EU Minimum Health Module.

Prevalence of self-reported chronic morbidity (%) ❖

- *Definition:* Proportion of those who reporting that they have any long-standing chronic illness or long-standing health problem. (ECHI)
- *Comment:* Part of the EU Minimum Health Module.

Prevalence of long-term activity limitations (%) ❖

- *Definition:* Proportion of those who reported that they have limitations in their daily activities due to health problems among all survey participants. (ECHI)
- *Comment:* Part of the EU Minimum Health Module.

Prevalence of self-reported myocardial infarction (%) ❖

- *Definition:* Proportion of individuals reporting to have been diagnosed to have had myocardial infarction

Prevalence of self-reported coronary heart disease (%) ❖

- *Definition:* Proportion of individuals reporting to have been diagnosed to have coronary heart disease

Prevalence of self-reported stroke (%) ❖

- *Definition:* Proportion of individuals reporting to have been diagnosed to have had stroke

4.1.6 Smoking

Prevalence of daily smokers (%) ❖

- *Definition:* Proportion of those who reported that they smoke daily among all survey participant.
- *Comment:* ECHI indicator name is "Regular smokers"

Prevalence of ex-daily smokers (%) ❖

- *Definition:* Proportion of those who reported that they have smoked daily or almost daily but do not smoke daily nowadays among all survey participants.

Prevalence of occasional smokers (%)

- *Definition:* Proportion of those who reported that they smoke occasionally among all survey participants.

Prevalence of manufactured cigarette smoking among daily smokers

- *Definition:* Proportion of those who reported that they frequently smoke manufactured cigarettes among daily smokers.

Prevalence of self-rolled cigarette smoking among daily smokers

- *Definition:* Proportion of those who reported that they frequently smoke self-rolled cigarettes among daily smokers.

Prevalence of pipe smoking among daily smokers

- *Definition:* Proportion of those who reported that they frequently smoke pipe among daily smokers.

Prevalence of cigar smoking among daily smokers

- *Definition:* Proportion of those who reported that they frequently smoke cigars among daily smokers.

Number of times smoked per day among daily smokers

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10th percentile, 25th percentile, 50th percentile (median), 75th percentile, 90th percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

Number of times smoked per day in population

- Population mean, standard error and 95 % confidence interval
- Population standard deviation

- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

Number of years smoked among daily smokers

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

Number of years smoked in population

- Population mean, standard error and 95 % confidence interval
- Population standard deviation
- Order statistics: 5 minimum values, 10^{th} percentile, 25^{th} percentile, 50^{th} percentile (median), 75^{th} percentile, 90^{th} percentile and 5 maximum values
- Distribution curve
- Cumulative distribution curve

4.2 Background variables

Area ❖

- *Definition:* Country

Sex ❖

- *Definition:* Biological sex
- Categories
  - Men
  - Women

Age ❖

- *Definition:* Age in completed years at the date of the health examination
- Categories
  - 25-64, age standardized
  - 25-34
  - 35-44
  - 45-54
  - 55-64
- If available
  - 18-24
  - 65-74
  - 75+

4.2.1 Socio-economic

Men and women separately for age group 25-64 years, age standardized.

Education ❖

- Highest level of completed education or training
- *Definition:* International Standard Classification of Education (ISCED), is used to define levels of education (UNESCO) Note: ISCED 1997 with 6 levels is used here, a new version ISCED 2011 with 8 levels has been proposed, but not yet accepted.
- Categories (ISCED 1997, ECHI)
  - Low or Basic education (ISCED level 0-2)
  - Middle or Secondary education (ISCED level 3 -4)
  - High or Tertiary education (ISCED level 5-6)

Thirds of years of education

- *Definition:* estimated thirds of years spent at school or in full-time study, taking into account birth years (Karvanen et al 2007)

Labour (employment) status ❖

- *Definition:* Person’s own perception of his/her main activity at present

- Categories
  - Employed (carrying out a job or profession, including unpaid work)
  - unemployed
  - others (student, retired, military or community service, domestic tasks, other inactive persons)

Living as a couple

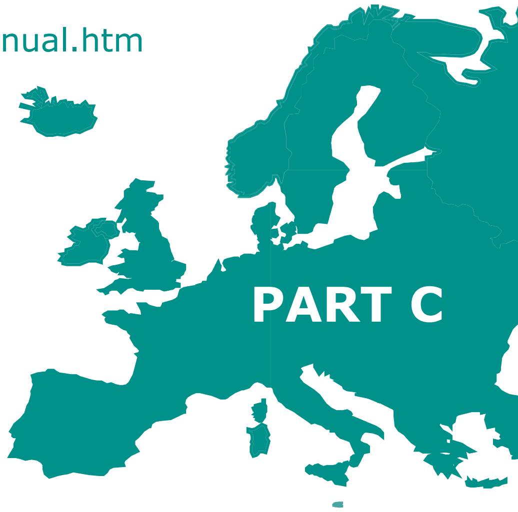
- *Definition:* de facto arrangement, including those who are legally married, living in consensual union or registered partnership
- Categories:
  - yes
  - no

Household's income (equivalised disposable income)

- *Definition:* Equivalised disposable income is the total income of a household, after tax and other deductions, that is available for spending or saving, divided by the number of household members converted into equalised adults; household members are equalised or made equivalent by weighting each according to their age, using the so-called modified OECD equivalence scale(Eurostat). The modified OECD equivalence scale (Hagernaas et al. 1994) gives the following weigths for the members of a household:
  - 1.0 to the first adult;
  - 0.5 to the second adult and each subsequent person aged 14 and over; and
  - 0.3 to each child aged under 14.
- Categories
  - Five categories. These are approximately fifths of the distribution of the equivalised income in each survey. Details of the derivation of the categories are given in the appendix.
- *Comment:* Information about children under 14 was not available in the questionnaire used in the EHES Pilot Project, but it was added to the final EHES questionnaire. The analysis of the EHES Pilot Project data used a measure for equivalised disposable income where the total income was divided by the square root of the household's size (OECD 2008).

References

- Eurostat modified OECD scale, available at http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Glossary:Equivalised_disposable_income
- Graham I, Atar D, Borch-Johnsen K, et al. (2007) European guidelines on cardiovascular disease prevention in clinical practice: executive summary. Fourth Joint Task Force of the European Society of Cardiology and other societies on cardiovascular disease prevention in clinical practice (constituted by representatives of nine societies and by invited experts). Eur J Card Prev and Rehab. 2007; 14(Supp 2): E1-E40
- Hagenaars A, de Vos K, Zaidi MA. Poverty statistics in the late 1980s: research based on micro-data, Office for official publications of the European communities. Luxemburg, 1994.
- IDF (2005) International Diabetes Federation. Clinical Guidelines Task Force. Global Guidelines for Type 2 Diabetes, 2005. Available from http://www.staff.ncl.ac.uk/philip.home/IDF%20GGT2D.pdf
- Karvanen J, Veronesi G, Kuulasmaa K. Defining thirds of schooling years in population studies. Eur J Epidemiol. 2007;22(8):487-92. Epub 2007 Jun 13.
- OECD. Growing unequal? Income distribution and poverty in OECD countries, Paris. 2008. Available from http://www.oecd-ilibrary.org/social-issues-migration-health/growing-unequal_9789264044197-en
- UNESCO (1997), ISCED available from http://www.uis.unesco.org/Education/Pages/international-standard-classification-of-education.aspx
- WHO (1995) Physical status: the use and interpretation of anthropometry. Report of a WHO Expert Committee. WHO Technical Report Series 854. Geneva: World Health Organization, 1995. Available from http://whqlibdoc.who.int/trs/WHO_TRS_854.pdf
- WHO (2000) Obesity: preventing and managing the global epidemic. Report of a WHO Consultation. WHO Technical Report Series 894. Geneva: World Health Organization, 2000. Available from http://whqlibdoc.who.int/trs/WHO_TRS_894.pdf
- WHO (2004) WHO Expert consultation. Appropriate body-mass index for Asian populations and its implications for policy and intervention strategies. The Lancet, 2004; 157-163
- WHO (2006) Definition and diagnosis of diabetes mellitus and intermediate hyperglycaemia. Report of the WHO/IDF consultation. World Health Organization, 2006. Available from http://www.who.int/diabetes/publications/Definition%20and%20diagnosis%20of%20diabetes_new.pdf
- WHO (2011a) Use of glycated haemoglobin (HbA1c) in the diagnosis of diabetes mellitus. Abbreviated report of a WHO consultation. World Health Organization 2011. Available from http://www.who.int/diabetes/publications/report-hba1c_2011.pdf
- WHO (2011b) Waist circumference and waist-hip ration. Report of a WHO Expert Consultation. Geneva, World Health Organization, 2011. Available from http://whqlibdoc.who.int/publications/2011/9789241501491_eng.pdf



5. Reporting

Tarja Tuovinen[1], Johan Heldal[2], Hanna Tolonen[1], Päivikki Koponen[1], Kari Kuulasmaa[1]

[1]National Institute for Health and Welfare (THL), Helsinki, Finland
[2]Statistics Norway, Oslo, Norway

Basic reports of the results of the HESs will be prepared using established EHES reporting system and database. This chapter introduces the outline of the basic report, established reporting system and principles of estimating the indicators.

5.1 Outline of the basic report

EHES basic reporting includes both country-specific and European level report on survey results. European level reporting of the results of the HESs is done by Reference Centre using central EHES database. The template of the basic report is specified in the Appendix 5a. Basic reports include relevant information about background and conducting of HES, description of the used methods and health indicator results.

5.2 Reporting system

European level basic reporting is done by EHES RC. Age standardized indicators by sex and age-group are calculated for each survey using HES data in central EHES database by statistical package R (R Development Core Team 2011). Results on indicators are saved in the database. Reporting system consists of calculation of individual based derived variables and estimation of indicators by sex and age. Specification of the derived variables used in the basic reporting can be found in the Appendix 5b. Technical definitions of the calculation oh the indicators are specified in the Appendix 5c. Program codes for executing the reporting system can be found in the Appendix 5d.

Derived variables are defined using survey data items that are specified in data transfer format in EHES Manual, Part C, Chapter 2. Calculation is done at individual based and each derived variable are saved in EHES database.

Indicators are estimated for each survey whose HES data is received to EHES RC. Each indicator estimate is categorized by sex and 10-year age-group for individuals aged 25-64 years. Indicators are saved in EHES database where they are in the format that can be used by different reporting systems.

5.3 Estimation of indicators

5.3.1 Introduction

This chapter defines the methods used in estimation of indicators for the EHES basic report. The used methods should be a kind that they can be applied to full-sized surveys in all countries with few modifications as long as they have been carried out according to the EHES recommendations. In the EHES RC estimation has been implemented using statistical package R (R Development Core Team 2011).

- Weighting is not used in the pilot phase due to small sample sizes. Weighting could be done by combining sampling weights and non-response weights for unit non-response.
- Age standardized estimates for 25-64-year-olds using ten year age-groups with equal weights are produced. Estimates for 10 year age-groups based on equal weighting of one year age groups estimates can be calculated also by e.g. smoothing with splines to increase precision of estimates.
- Imputation is not used in pilot phase but it can be performed to item non-response, not to unit non-response.
- Density functions are not estimated in pilot phase. Densities can be estimated with an adaptive kernel method using age standardization.

5.3.2 Weighting

When estimating based on survey samples we need weights. The weights will generally consist of two components.

The sampling weights w_i, also called Horwitz-Thompson weights are the inverse of the inclusion probabilities π_i for the individuals in the survey and should be known for all individuals invited to the survey whether they participated or not, $w_i = \frac{1}{\pi_i}$. If the value of a variable x is known for the entire sample, the sum

$$\hat{T} = \sum_{i \in s} w_i x_i$$

provides an unbiased estimate of the population total $T = \sum_{population} x_i$. Here s is the entire sample and $i \in s$ means that the sum is over all members of the sample. The formula can be applied to any subgroup of the population. The simple formula $\hat{N} = \sum_{i \in s} w_i$ is an unbiased estimator of the population size N and

$$\hat{\mu} = \frac{\hat{T}}{\hat{N}} = \frac{\sum_{i \in s} w_i x_i}{\sum_{i \in s} w_i} = \sum_{i \in s} \left(\frac{w_i}{\sum_{i \in s} w_i} \right) x_i$$

is an approximately unbiased estimator of the mean $\mu = \frac{T}{N}$. Whereas wi are weights for estimating population totals $\frac{w_i}{\sum_{j \in s} w_j}$ act as weights for estimating population means.

1. The non-response weights g_i. These weights are intend as corrections for unit non-response and are usually based on variables known for everyone in the original sample such as sex, age and geographic affiliation. In some countries education and income are available. The non-response weights expands the net sample consisting of participants to the gross sample of all invitees. They are most often constructed using some kind of post-stratification or calibration methods so that estimating totals in the gross sample based on the net sample for some of the variables known for the entire gross sample will provide exactly correct figures. Combining the sampling weights and the non-response weights for all participants to $\hat{v} = g_i w_i$ provides the final estimation weights.

Item non-response is not accounted for by the g_i weights. However, this is less of a problem when estimating averages than when estimating population totals. Estimation of population totals results in underestimation if there is missing values. In the estimation of averages, missing values can be discarded both in numerator and denominator and missingness does not produce bias. The estimators of the population averages are form

$$\hat{\mu} = \sum_{i \in r} \left(\frac{v_i}{\sum_{j \in r} v_j} \right) x_i = \sum_{i \in r} \omega_i x_i$$

where r is the set of participants for whom we have non-missing value of x and $\omega_i = \frac{v_i}{\sum_{j \in r} v_j}$.

An alternative to calculate weights g_i is to calibrate them to population totals rather than just gross sample totals. This can be done in one step or in two steps by first calibrating them to gross sample level and then to population level. This requires good statistics at national level for variables that are observed with good quality for all participants. This way more variables may be accessible than the ones we have in the sampling frame. There must also be a balance: too many calibrating variables may twist the original weights so much out of shape that we lose control of how they work.

For comparability of the results between countries, the methods to establish the g_i weights should be the same in all countries. This puts limits on what methods can be used. Another aspect is the used age standardization and estimation separately for men and women. That's why the use of sex and age for weighting is less interesting: calibrations for sex and age will cancel out in the ωi weights. What remains is the adjustment for different participation rates by geographic area (strata in sampling design).

5.3.3 Estimation

Estimated indicators are age standardized estimates for 25-64-year-old men and women separately using ten year age groups. If the mean for age-group a (25-34-, 35-44-, 45-54-, 55-64-year-olds) is μa then the age standardized population mean is

$$\mu_{25-64} = \frac{1}{4}\sum_{a} \mu_a .$$

Such a age standardized mean is actually the mean in a hypothetical population where the age distribution is flat. It controls for differential age distribution among the countries and is necessary to make estimates comparable across Europe. For comparability over the time in later health examination surveys, the age-weights should also be kept constant.

Let r_a be the set of respondents in the age-group a and the variable of interest is y . If a individual i in the sample belongs to r_a then it will have a weight

$$\omega_i^a = \frac{v_i}{\sum_{i \in r_a} v_j} = \omega^\wedge(i) \frac{a}{\sum_{j \in r} \omega_j}$$

The parameter μ_a is then estimated as

$$\widehat{\mu_a} = \sum_{i \in r_a} \omega_i{}^\wedge(a) y_i$$

and

$$\hat{\mu}_{25-64} = \frac{1}{4} \sum_a \hat{\mu}_a$$

5.3.4 Imputation

One approach to deal with missingness in the survey data is to do multiple imputations for missing values (Rubin 1987). The purpose of imputation is to construct complete data sets that can be analysed without having to exclude records that have missing values on one or more variables so that valuable information is lost. It has also been seen as desirable that analysts should be able to analyse such data sets with standard software as if all data were real and without having to pay too much attention to the fact that some of them are imputed.

In survey sampling theory we basically distinguish between two kinds of missing data.

1. **Unit missing.** This includes all individuals that have not participated in the survey and for whom we don't have any other information than what we could get from the sampling frame.
2. **Item missing.** This is missing on particular variables (questions or measurements) for people who participated at least in parts of the survey.

Another aspect of missing data is the statistical mechanism that causes some people to participate and others to say no. Rubin (1976) introduced the following distinctions

1. **Missing Completely At Random (MCAR)** in relation to a variable means that all individuals have the same probability of being measured on that variable. For instance, if we observe age for everyone invited to the sample but weight may be missing and the probability that weight is missing the same for all regardless of age and weight we can say that the data are MCAR.

2. However, if the probability that weight is missing depends on age but not on weight for individuals with the same age we say that weight is **Missing At Random (MAR)**.
3. If the probability of weight missing depends on the weight directly, for instance if people who are excessively overweight do not want to come and hear something they already know very well, then we say that weight is **Not Missing At Random (NMAR)**.

The mechanisms affecting unit missingness and item missingness may be very different. It is likely that missing-mechanisms are NMAR for quite many of the HES variables. Treating NMAR correctly requires a thorough understanding of how the missing mechanisms work to establish a model for doing imputations. Such models will sometimes have to depend on assumptions that are difficult to verify and the information that can be used to estimate such models is often weak. It is often recommended to use the information that is available, pretend that the missing mechanism is MAR and rather evaluate the sensitivity of deviations from the MAR assumption later. Models suggested in this note will assume a MAR mechanism.

Whenever doing imputations, the general practice in survey sampling is to do imputation on items missing but not on unit missing. Unit missing is taken care of by weighting. When imputing a missing value for a specific variable on a record it is, at least in principle, possible to take advantage of the information from other variables that have valid values on the record.

The variables from the sampling frame are all we have for non-participants. The frame should at least contain information on the geographical affiliation of the sampling unit. If the sampling unit is a household or address a visit to the address may provide information on the number of eligible living there and their distribution with respect to sex and age even if they have not cooperated with the survey. But this will not always be the case. When samples have been taken from registers some kind of information on sex and age, sometimes marital status and other demographic information may come along.

For people participating in the survey we will most often have the interview, but we may miss one or more, perhaps all the other modules. In rarer cases we may have some of the modules but not the interview. In some of the modules, e.g. the interview or anthropometric module, some items may be missing while we have others. A question that arises if doing imputations is to which extent we should allow imputations of entire modules based on information from others.

As estimates or predictions of the true missing values imputations are always uncertain. This uncertainty adds to the sampling uncertainty and uncertainty in measurements. If an imputed data set is analyzed with standard methods not paying attention to the imputation method used, the variance estimates will be underestimated and test will reject too often.

There are several imputation techniques. The oldest ones are the “hot-deck methods” where the records have been grouped according to known variables (age, sex, geography etc.). Then each missing observation is replaced with a value drawn at random among the observed values in the same group. Second, we have nearest neighbor methods which do not require grouping but sorting and there are mean and regression imputation methods. Regression imputation can be carried out deterministically by replacing the missing value with a predicted one. Alternatively a randomly drawn regression residual can be added to reflect prediction uncertainty.

Multiple imputation where several values are imputed for one missing value goes back to Rubin 1978. The purpose of multiple imputation is to have the imputation uncertainty reflected in the data. Rubin devised a variance formula which should take care of the imputation uncertainty in a correct manner. However, for Rubin’s formula to be correct the imputation has to be ‘proper’. This means that the imputation must be based on a model for the missing data given the observed data. A prior distribution for the parameters in the model is required and a full Bayesian posterior distribution must be established. Then the imputed values must be drawn from this posterior distribution. Multiple imputation can be done with other methods than that of Rubin. In principle all kinds of stochastic imputation can be carried out multiply.

If imputation is carried out for the indicators in the basic reports, the imputed values must be stored at the data files as documentation and to make it possible to reproduce the estimates in the future. This also has consequences for the layout of the stored data files. Every variable that has been subject to imputation must have an associated indicator variable telling if the value on the record has been imputed or not. If multiple imputation has been used there must be a column for every imputation done. If for a variable missing values have been imputed five times there must be five columns to store the data for the variable. The values in these columns will differ for records where the variable has been imputed but will be identical for records having a valid measured value. In addition there can be a column that for each record contains the average of the imputed values on the record.

5.3.5 Smoothing

If the estimates for 10 year age-groups are calculated using one year age groups a^*, each estimate $\hat{\mu}_a$ will be based on the small number of observations that falls in age group a^* . The sequence $\hat{\mu}_{25}, \hat{\mu}_{26}, \dots, \hat{\mu}_{64}$ will therefore behave somewhat erratic, jumping up and down. However, for many indicators the underlying sequence of true values $\mu_{25}, \mu_{26}, \dots, \mu_{64}$ will behave smoothly. Smoothing the sequence $\mu_{25}, \mu_{26}, \dots, \mu_{64}$using for instance smoothing splines (or some other technique) would then strengthen the estimates $\hat{\mu}_a$ for each age-group by borrowing strength from their neighbors. This will also strengthen the averages $\hat{\mu}_{25-34}$ etc. But since the smoothed estimates, call them $\hat{\mu}^s_{25}, \hat{\mu}^s_{26}, \dots, \hat{\mu}^s_{64}$, will be highly correlated (which the basic estimates are not) the improvement for a ten year average will be smaller than for each one-year group. The smoothed trend will for many indicators be of interest in itself.

5.3.6 Density estimation

Density estimation is used to be able to depict the distribution of a variable graphically. The general form of a kernel density estimator is

$$\widehat{f(x)} = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - y_i}{h}\right)$$

where y_i is the observed value, n is the number of observations, h is the bandwidth (also called the smoothing parameter) and K is kernel function. K must be a function symmetric around 0 and integrate to unity. Usually a probability distribution, for instance the standard normal distribution, is used. A large bandwidth produces a smooth density estimate that may underestimate the peaks in the density function while a too small bandwidth will produce a more erratic function. It is possible through a cross-validation approach to estimate h to minimize the integrated mean square error of $\widehat{f(x)}$,

$$\int \left(\widehat{f(x)} - f(x)\right)^2 dx$$

Density functions are not estimated in pilot phase.

It is possible to improve the density estimate above using an adaptive bandwidth which is larger in low-density areas and smaller in high-density areas. One way of doing this is to intro-

duce a local bandwidth factor λ_i

$$\widehat{f(x)} = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h\lambda_i} K\left(\frac{x - y_i}{h\lambda_i}\right)$$

For further details, see Silverman (1986) p. 100-119.

Like estimates for ordinary averages, estimates for density functions can be weighted. This can be done by replacing 1/n with the

weights ω_i and we get

$$\widehat{f(x)} = \frac{1}{n}\sum_{i \in S} \frac{\omega_i}{h\lambda_i} K\left(\frac{x - X_i}{h\lambda_i}\right)$$

Notice that this density estimator reproduces the estimated means $\hat{\mu}$, i.e.

$$\int x\widehat{f(x)}dx = \sum_{i \in S} \omega_i x_i = \hat{\mu}_x$$

Like the estimates for averages the density estimates should be age standardized. There are two ways of doing this

- Estimate for each age-group, i.e.

$$\hat{f}_a(x) = \sum_{i \in r_a} \frac{\omega_i^a}{h_a \lambda\hat{}(i)(a)} K\left(\frac{x - y_i}{h_a \lambda_i\hat{}(a)}\right)$$

and calculate

$$\hat{f}_{25-64}(x) = \frac{1}{4}\sum_{a} \hat{f}_a(x)$$

Another and probably better approach is to estimate $f_{25-64}(x)$ etc. more directly as

$$\hat{f}_{25-64}(x) = \sum_{i \in 25-64} \omega_i{}^\wedge \frac{a}{10h\lambda_i} K\left(\frac{x - y_i}{h\lambda_i}\right)$$

where h is estimated with a common value for all age groups. Both estimates can be seen as age-standardized.

References

- R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.
- Rubin, D.B. (1976): Inference and missing data. Biometrika, 63, 581-592.
- Rubin, D.B. (1987): Multiple imputation for nonresponse in surveys. J. Wiley & Sons, New York.

Appendix 5a. Template of the basic report

- Introduction
- Background
  - What is known about previous health surveys?
  - What is known or not known about health status based on existing data sources and previous surveys?
- Survey aims
  - Why survey is conducted?
- Target population and sampling
  - Which population groups are covered and which are not covered by target population?
  - How the sampling was conducted and which sampling frames were used?
- Sample characteristics and participation rates (incl. eligibility) by sex and 10-year age groups
  - Tablel1. Proportion of eligibles, Proportion of not-contacted, Proportion of non-eligible
  - Table 2. Proportion of participants
    - questionnaire only
    - examination only
    - both
  - Table 3. Proportion of non-participants
- Data collection methods
  - How data was collected?
    - Questionnaire administration
      - Use of proxies (if used)
    - Measurement procedures
- Analysis
  - What kind of weighting was used?
  - Sub-groups used for reporting
- Results
  - Rationale for each indicator
  - Definition of the indicator
  - Table with results by sex and 10-year age group, age standardized 25-64

- Summary of the results by indicator, including notes if known that data has some problems, shortcomings or special features.
- Background items
  - Area
  - Sex
  - Age
  - Education
  - Labour (employment) status
- Health indicators
  - Blood pressure
    - Systolic blood pressure: Population mean, standard error and 95 % confidence interval
    - Diastolic blood pressure: Population mean, standard error and 95 % confidence interval
    - Prevalence of self-reported high blood pressure (hypertension) (%)
    - Prevalence of actual and potential hypertensives (%)
    - Awareness of elevated blood pressure among actual and potential hypertensives (%)
    - Prevalence of anti-hypertensive drug use in the population (%)
    - Prevalence of anti-hypertensive drug use among actual and potential hypertensives (%)
    - Prevalence of optimal blood pressure among those using anti-hypertensive drugs (%)
    - Proportion of the population with blood pressure measurement in the past 5 years (%)
  - Lipids
    - Serum total cholesterol: Population mean, standard error and 95 % confidence interval
    - Serum high-density lipoprotein cholesterol: Population mean, standard error and 95 % confidence interval
    - Serum non-HDL cholesterol: Population mean, standard error and 95 % confidence interval
    - Prevalence of self-reported elevated blood cholesterol (%)
    - Prevalence of actual or potential elevated serum total cholesterol in the population (%)

    - Prevalence of lipid lowering drug use in the population (%)
    - Prevalence of lipid lowering drug use among those with elevated total cholesterol (%)
    - Prevalence of optimal total cholesterol level among those using lipid lowering drugs (%)
    - Proportion of the population with cholesterol measurement in the past 5 years (%)
- Glucose
    - Fasting plasma glucose: Population mean, standard error and 95 % confidence interval
    - Glycated haemoglobin (HbA1c): Population mean, standard error and 95 % confidence interval
    - Prevalence of self-reported diabetes (%)
    - Prevalence of diabetes (%) based on fasting glucose, HbA1c or use of diabetes medication
    - Prevalence of diabetes medication in population (%)
    - Prevalence of diabetes medication among actual and potential diabetics (%)
    - Awareness of diabetes among diabetics (%)
    - Prevalence of the population with blood glucose measured in the past 5 years (%)
- Anthropometric measurements
    - Waist circumference: Population mean, standard error and 95 % confidence interval
    - BMI (measured): Population mean, standard error and 95 % confidence interval
    - BMI self-reported: Population mean, standard error and 95 % confidence interval
    - Prevalence of overweight (%) based on measured BMI
    - Prevalence of obesity (%) based on measured BMI
    - Categories of waist circumference
- Self-perceived health and chronic diseases
    - Self-perceived health
    - Prevalence of self-reported chronic morbidity (%)
    - Prevalence of long-term activity limitations (%)

    - Prevalence of self-reported myocardial infarction (%)
    - Prevalence of self-reported coronary heart disease (%)
    - Prevalence of self-reported stroke (%)
- Smoking
    - Prevalence of daily smokers (%)
    - Prevalence of ex-daily smokers (%)

Appendix 5b. Specification of derived variables

Definitions of variables transferred to EHES RC can be found in EHES Manual, Part C, Chapter 2. Further information and program codes for calculation of derived variables, see the Appendix 5d.

Abbreviations:
!= not equal.

Background items

COLUMN NAME	TYPE	DEFINITION
EDUCATION_CAT	integer	*Highest education leaving certificate obtained*: 1 = Low or Basic education (ISCED level 0-2), if EDUCATION = 1 or 2 or 3. 2 = Middle or Secondary education (ISCED level 3-4), if EDUCATION = 4 or 5. 3 = High or Tertiary education (ISCED level 5-6), if EDUCATION = 6 or 7. 9 = Missing data if EDUCATION = 9.
LABOUR_CAT	integer	*Current labour (employment) status*: 1 = Employed (carrying out a job or profession, including unpaid work), if LABOUR_STATUS = 1. 2 = Unemployed, if LABOUR_STATUS = 2. 3 = Other (student, retired, military or community service, domestic tasks, other inactive persons), if LABOUR_STATUS = 3 or 4 or 5 or 6 or 7 or 8. 9 = Missing data if LABOUR_STATUS = 9.

COLUMN NAME	TYPE	DEFINITION
EMPLOYMENT1	integer	*Employee, self-employed or working without payment as a family worker:* 1 = Employe if PROFIT = 1 and EMPLOYMENT = 1. 2 = Self-employe if PROFIT = 1 and EMPLOYMENT = 2. 3 = Family worker if PROFIT = 1 and EMPLOYMENT = 3. 9 = Missing data if PROFIT = 1 and EMPLOYMENT = 9. . = irrelevant if PROFIT != 1.

Blood pressure

COLUMN NAME	TYPE	DEFINITION
BP_SYSTM23	decimal (0.01)	*Systolic blood pressure (mmHg), 2nd and 3rd measurement*. If EXAM = 1 or QUEST = 1: Mean of BP_SYST2 and BP_SYST3. 999.99 (missing) if EXAM = 2 or BP_SYST2 = 999 or BP_SYST3 = 999.
BP_SYSTM12	decimal (0.01)	*Systolic blood pressure (mmHg), 1st and 2nd measurement*. If EXAM = 1 or QUEST = 1: Mean of BP_SYST1 and BP_SYST1. 999.99 (missing) if EXAM = 2 or BP_SYST1 = 999 or BP_SYST2 = 999.
BP_SYSTM123	decimal (0.01)	*Systolic blood pressure (mmHg), 1st, 2nd and 3rd measurement*. If EXAM = 1 or QUEST = 1: Mean of BP_SYST1, BP_SYST2 and BP_SYST3. 999.99 (missing) if EXAM = 2 or BP_SYST1 = 999 or BP_SYST2 = 999 or BP_SYST3 = 999.

COLUMN NAME	TYPE	DEFINITION
BP_DIASTM23	decimal (0.01)	*Diastolic blood pressure (mmHg), 2nd and 3rd measurement.* If EXAM = 1 or QUEST = 1: Mean of BP_DIAST2 and BP_DIAST3. 999.99 (missing) if EXAM = 2 or BP_DIAST2 = 999 or BP_DIAST3 = 999.
BP_DIASTM12	decimal (0.01)	*Diastolic blood pressure (mmHg), 1st and 2nd measurement.* If EXAM = 1 or QUEST = 1: Mean of BP_DIAST1 and BP_DIAST2. 999.99 (missing) if EXAM = 2 or (BP_DIAST1 = 999 or BP_DIAST2 = 999).
BP_DIASTM123	decimal (0.01)	*Diastolic blood pressure (mmHg), 1st, 2nd and 3rd measurement*. If EXAM = 1 or QUEST = 1: Mean of BP_DIAST1, BP_DIAST2 and BP_DIAST3 if ELIG = 1 and EXAM = 1. 999.99 (missing) if EXAM = 2 or (BP_DIAST1 = 999 or BP_DIAST2 = 999 or BP_DIAST3 = 999).
BP_SYST	decimal (0.01)	*Systolic blood pressure (mmHg)*. If EXAM = 1 or QUEST = 1: BP_SYSTM23 if not missing, then BP_SYSTM12 if not missing, then BP_SYST2 if not missing, then BP_SYST3 if not missing, then BP_SYST1 if not missing, then 999.99 (missing) if EXAM = 2 or BP_SYST1 = BP_SYST2 = BP_SYST3 = 999 (missing).

COLUMN NAME	TYPE	DEFINITION
BP_DIAST	decimal (0.01)	*Diastolic blood pressure (mmHg)*. If EXAM = 1 or QUEST = 1: BP_DIASTM23 if not missing, then BP_DIASTM12 if not missing, then BP_DIAST2 if not missing, then BP_DIAST3 if not missing, then BP_DIAT1 if not missing, then 999.99 (missing) if EXAM = 2 or BP_DIAST1 = BP_DIAST2 = BP_DIAST3 = 999 (missing).
		Hypertension:
MEDICINE_HBP1	integer	*Medicines for high blood pressure*. If MEDICINE = 1 and (EXAM = 1 or QUEST = 1): 1 (yes) if MEDICINE = 1 and MEDICINE_HBP = 1. 2 (no) if ((MEDICINE = 1 and MEDICINE_HBP = 2). 9 (missing) if QUEST = 2 or (MEDICINE = 1 and MEDICINE_HBP != 1 and MEDICINE_HBP != 2). . (irrelevant) if MEDICINE != 1.
MEDICINE_HBP_POP	integer	*Medicines for high blood pressure among population.* If EXAM = 1 or QUEST = 1: 1 (yes) if MEDICINE_HBP1 = 1. 2 (no) if MEDICINE = 2 or MEDICINE_HBP1 = 2. 9 (missing) if QUEST = 2 or (MEDICINE = 1 and MEDICINE_HBP1 = 9) or MEDICINE != (1 and 2) .

COLUMN NAME	TYPE	DEFINITION
BP_HBP	integer	*Actual or potential hypertension*. If EXAM = 1 or QUEST = 1: 1 (yes) if none of 3 is missing and (BP_SYST >= 140 or BP_DIAST >= 90 or MEDICINE_HBP1 = 1). 2 (no) if none of 3 is missing and (BP_SYST < 140 and BP_DIAST < 90 and MEDICINE_HBP1 = 2). 9 (missing) if EXAM = 2 or QUEST = 2 or BP_SYST = missing or BP_DIAST = missing or (MEDICINE_HBP1 != 1 and MEDICINE_HBP1 != 2).
MEDICINE_HBP_ILL	integer	*Medicines for high blood pressure among actual or potential hypertensives.* If EXAM = 1 or QUEST = 1: 1 (yes) if BP_HBP = 1 and MEDICINE_HBP1 = 1. 2 (no) if (BP_HBP = 1 and MEDICINE_HBP1 = 2) or (BP_HBP = 1 and MEDICINE = 2). 9 (missing) if BP_HBP = 1 and (MEDICINE_HBP1 = 9 or MEDICINE = 9). . (irrelevant) if BP_HBP != 1.
BP_AWARE	integer	*Awareness of hypertension among actual or potential hypertensives*. If EXAM = 1 or QUEST = 1: 1 (yes) if BP_HBP = 1 and DIAGN_HBP = 1. 2 (no) if BP_HBP = 1 and (DIAGN_HBP =2 or DIAGN_HBP = 8). 9 (missing) if BP_HBP = 1 and DIAGN_HBP != 1 and DIAGN_HBP != 2 and DIAGN_HBP != 8. . (irrelevant) if BP_HBP != 1.

COLUMN NAME	TYPE	DEFINITION
BP_OPTIMAL	integer	*Optimal blood pressure among those using anti-hypertensive drugs.* If EXAM = 1 or QUEST = 1: 1 (yes) if MEDICINE_HBP1 = 1 and BP_EPB = 2. 2 (no) if MEDICINE_HBP1 = 1 and BP_EPB = 1. 9 (missing) if MEDICINE_HBP1 = 1 and BP_EPB != 1 and BP_EPB != 2. . (irrelevant) if MEDICINE_HBP1 != 1.

Lipids

COLUMN NAME	TYPE	DEFINITION
BS_CHOL_MMOL	decimal 0.0001	*Serum total cholesterol in mmol/l*. If EXAM = 1 or QUEST = 1: BS_CHOL if BS_UNIT_CHOL= 1. BS_CHOL divided by 38.67 if BS_UNIT_CHOL= 2. 9999.9999 (missing) if EXAM = 2 or BS_CHOL = 9999.99.
BS_HDL_MMOL	decimal	*Serum HDL cholesterol in mmol/l*. If EXAM = 1 or QUEST = 1: BS_HDL if BS_UNIT_HDL = 1 . BS_HDL divided by 38.67 if BS_UNIT_HDL = 2. 9999.9999 (missing) if EXAM = 2 or BS_HDL = 9999.99.
BS_NONHDL	decimal	*Serum non-HDL cholesterol (mmol/l)*. If EXAM = 1 or QUEST = 1: BS_CHOL_MMOL - BS_HDL_MMOL if BS_CHOL_MMOL and BS_HDL_MMOL ≠ 9999.9999. 9999.9999 (missing) if EXAM = 2 or BS_CHOL_MMOL = 9999.9999 or BS_HDL_MMOL = 9999.9999.

COLUMN NAME	TYPE	DEFINITION
MEDICINE_CHOL1	integer	*Medicines for lowering the blood cholesterol level*. If EXAM = 1 or QUEST = 1: 1 (yes) if MEDICINE = 1 and MEDICINE_CHOL = 1. 2 (no) if ((MEDICINE = 1 and MEDICINE_CHOL = 2)). 9 (missing) if QUEST = 2 or (MEDICINE = 1 and MEDICINE_CHOL != 1 and MEDICINE_CHOL != 2)). . (irrelevant) if MEDICINE != 1.
BS_HCHOL	integer	*Actual or potential elevated serum total cholesterol*. If EXAM = 1 or QUEST = 1: 1 (yes) if neither one is missing and (BS_CHOL_MMOL >= 5 or MEDICINE_CHOL1 = 1). 2 (no) if neither one is missing and (BS_CHOL_MMOL < 5 and MEDICINE_CHOL1 = 2). 9 (missing) if EXAM = 2 or QUEST = 2 or BS_CHOL_MMOL = 999.9999 or (MEDICINE_CHOL1 != 1 and MEDICINE_CHOL1 != 2).
BS_CHOL_AWARE	integer	*Awareness of elevated cholesterol among those with actual or potential elevated serum total cholesterol (incl. use of medication).* If EXAM = 1 or QUEST = 1: 1 (yes) if BS_HCHOL = 1 and DIAGN_CHOL = 1. 2 (no) if BS_HCHOL = 1 and (DIAGN_CHOL **=**2 or DIAGN_CHOL = 8). 9 (missing) if BS_HCHOL = 1 and DIAGN_CHOL != 1 and DIAGN_CHOL != 2 and DIAGN_CHOL != 8. . (irrelevant) if BS_HCHOL != 1.

COLUMN NAME	TYPE	DEFINITION
BS_CHOL_ OPTIMAL	integer	*Optimal total cholesterol level among those using lipid lowering drugs.* If EXAM = 1 or QUEST = 1: 1 (yes) if MEDICINE_CHOL1 = 1 and BS_ECHOL = 2. 2 (no) if MEDICINE_CHOL1 = 1 and BS_ECHOL = 1. 9 (missing) if MEDICINE_CHOL1 = 1 and BP_ECHOL != 1 and BP_ECHOL != 2. . (irrelevant) if MEDICINE_HBP1 != 1.

Glucose

COLUMN NAME	TYPE	DEFINITION
BS_HBA1C_MMOL	decimal	*Whole blood glycated haemoglobin concentration in mmol/l (IFCC).* If EXAM = 1 or QUEST = 1: BS_HBA1C if BS_UNIT_HBA1C = 1. (BS_HBA1C - 2.15) * 10.929 if BS_UNIT_HBA1C = 2. 9999.9999 (missing) if EXAM = 2 or BS_HBA1C = 9999.99.
BS_FGLC_MMOL	decimal 0.0001	*Fasting plasma glucose in mmol/l.* If EXAM = 1 or QUEST = 1: BS_GLC if BS_UNIT_GLC = 1 and BS_FAST = 1. BS_GLC * 0.0555 if BS_UNIT_GLC = 2 and BS_FAST = 1. 9999.9999 (missing) if EXAM = 2 or BS_GLC = 9999.99 or BS_FAST ≠ 1.

COLUMN NAME	TYPE	DEFINITION
MEDICINE_DIAB1	integer	*Medicines for diabetes*. If EXAM = 1 or QUEST = 1: 1 (yes) if MEDICINE = 1 and MEDICINE_DIAB = 1. 2 (no) if (MEDICINE = 1 and MEDICINE_DIAB = 2). 9 (missing) if QUEST = 2 or (MEDICINE = 1 and MEDICINE_DIAB != 1 and MEDICINE_DIAB != 2). . (irrelevant) if MEDICINE != 1.
BS_DIAB_MSR	integer	*Actual or potential diabetes based on fasting glucose or HbA1c.* If EXAM = 1 or QUEST = 1: 1 (yes) if BS_HBA1C_MMOL >= 48 or (BS_HBA1C_MMOL = 9999.9999 and BS_FGLC_MMOL >= 7.0). 2 (no) if BS_HBA1C_MMOL < 48 or (BS_HBA1C_MMOL = 9999.9999 and BS_FGLC_MMOL < 7.0) . 9 (missing) if EXAM = 2 or (BS_FGLC_MMOL = 9999.9999 and BS_HBA1C_MMOL= 9999.9999).
BS_DIAB	integer	*Actual or potential diabetes based on fasting glucose or HbA1c or use of medication*. If EXAM = 1 or QUEST = 1: 1 (yes) if neither one is missing and (BS_DIAB_MSR = 1 or MEDICINE_DIAB1 = 1). 2 (no) if BS_DIAB_MSR = 2 and MEDICINE_DIAB1 = 2. 9 (missing) if EXAM = 2 or QUEST = 2 or (BS_DIAB_MSR != 1 and BS_DIAB_MSR != 2 and MEDICINE_DIAB1 != 1 and MEDICINE_DIAB1 != 2).

COLUMN NAME	TYPE	DEFINITION
BS_DIAB_AWARE	integer	*Awareness of diabetes among actual or potential diabetics (incl. use of medication)*. If EXAM = 1 or QUEST = 1: 1 (yes) if BS_DIAB = 1 and DIAGN_DIAB = 1. 2 (no) if BS_DIAB = 1 and (DIAGN_DIAB =2 or DIAGN_DIAB = 8). 9 (missing) if BS_DIAB = 1 and DIAGN_DIAB != 1 and DIAGN_DIAB != 2 and DIAGN_DIAB != 8. . (irrelevant) if BS_DIAB != 1.
BS_GLC_OPTIMAL	integer	*Optimal glucose/HbA1C level among those using diabetes medication.* If EXAM = 1 or QUEST = 1: 1 (yes) if MEDICINE_DIAB1 = 1 and BS_DIAB_MSR2 = 2. 2 (no) if MEDICINE_DIAB1 = 1 and BS_DIAB_MSR2= 1. 9 (missing) if MEDICINE_DIAB1 = 1 and BS_DIAB_MSR2 != 1 and BS_DIAB_MSR2 != 2. . (irrelevant) if MEDICINE_DIAB1 != 1.

Anthropometrics and obesity

COLUMN NAME	TYPE	DEFINITION
BMI	decimal	*Body mass index (kg/m2) based on measured height and weight.* If EXAM = 1 or QUEST = 1: WEIGHT / (HEIGHT/100)^2. 999.99 (missing) if EXAM = 2 or WEIGHT = 999.99 or HEIGHT = 999.9.
BMI_SR	decimal	*Body mass index (kg/m2) based on self-reported height and weight.* If EXAM = 1 or QUEST = 1: SWEIGHT / (SHEIGHT/100)^2. 999.99 (missing) if QUEST = 2 or SWEIGHT = (8 or 9) or SHEIGHT = (8 or 9).

COLUMN NAME	TYPE	DEFINITION
BMI_ OVERWEIGHT	integer	*Overweight based on measured BMI*. If EXAM = 1 or QUEST = 1: 1 (yes) if BMI >= 25. 2 (no) if BMI < 25. 9 (missing) if EXAM = 2 or BMI = 999.99.
BMI_OBESE	integer	*Obesity based on measured BMI*. If EXAM = 1 or QUEST = 1: 1 (yes) if BMI >= 30. 2 (no) if BMI < 30. 9 (missing) if EXAM = 2 or BMI = 999.99.
WAIST_LARGE	integer	*Large waist circumference*. If EXAM = 1 or QUEST = 1: 1 (yes) if (SEX = 1 (male) and WAIST > 102) or (SEX = 2 (female) and WAIST > 88). 2 (no) if (SEX = 1 and WAIST <= 102) or (SEX = 2 and WAIST <= 88). 9 (missing) if EXAM = 2 or WAIST = 999.9 or (SEX != 1 and SEX != 2).

Self-perceived health

COLUMN NAME	TYPE	DEFINITION
HEALTH_GOOD	integer	*Self perceived health at least good.* If EXAM = 1 or QUEST = 1: 1 (yes) if HEALTH = 1 or 2. 2 (no) if HEALTH = 3 or 4 or 5. 8 (don't know) if HEALTH = 8. 9 (missing) if QUEST = 2 or HEALTH = 9.
LIMITED_YES	integer	*Long-term activity limitations due to health problems*. If EXAM = 1 or QUEST = 1: 1 (yes) if LIMITED = 1 or 2. 2 (no) if LIMITED = 3. 8 (don't know) if LIMITED = 8. 9 (missing) if QUEST = 2 or LIMITED = 9.

Specification of additional derived variables

Further information and program codes for calculation of derived variables, see the Appendix 5d.

Questionnaire items

COLUMN NAME	TYPE	DEFINITION
SCHOOL_THIRDS	integer	*Thirds of years of education.* Derived from BIRTH_YEAR, SCHOOL_YEAR. Reference: J. Karvanen, G. Veronesi, K. Kuulasmaa (2007). Defining thirds of schooling years in population studies. *European Journal of Epidemiology*, Volume 22, Number 8, 487--492.
INCOME_EQUI	integer	*Household's equivalised disposable income class, 5 classes.(See Appendix 5e.)*

Blood pressure

COLUMN NAME	TYPE	DEFINITION
BP_HSBP	integer	*Systolic hypertension*. If EXAM = 1 or QUEST = 1: 1 (yes) if BP_SYST >= 140. 2 (no) if BP_SYST < 140. 9 (missing) if EXAM = 2 or BP_SYST = 999.99.
BP_EPB	integer	*Elevated blood pressure.* If EXAM = 1 or QUEST = 1: 1 (yes) if neither one is missing and (BP_SYST >= 140 or BP_DIAST >= 90). 2 (no) if neither one is missing and (BP_SYST < 140 and BP_DIAST < 90). 9 (missing) if EXAM = 2 or BP_SYST = 999.99 or BP_DIAST = 999.99.

COLUMN NAME	TYPE	DEFINITION
BP_CATEGORY	integer	*Categories of blood pressure.* If EXAM = 1 or QUEST = 1: 1 (optimal) if BP_SYST < 120 and BP_DIAST < 80, neither one missing. 2 (normal) if (120 <= BP_SYST < 130 and BP_DIAST < 85) or (80 <= BP_DIAST < 85 and BP_SYST < 130), neither one missing. 3 (high normal) if (130 <= BP_SYST < 140 and BP_DIAST > 90) or (85 <= BP_DIAST < 90 and BP_SYST<140), neither one missing. 4 (grade 1 hypertension) if (140 <= BP_SYST < 160 and BP_DIAST < 100) or (90 <= BP_DIAST < 100 and BP_SYST < 160), neither one missing. 5 (grade 2 hypertension) if (160 <= BP_SYST < 180 and BP_DIAST<110) or (100 <= BP_DIAST < 110 and BP_SYST < 180), neither one missing. 6 (grade 3 hypertension) if BP_SYST >= 180 or BP_DIAST >= 110, neither one missing. 9 (missing) if EXAM = 2 or BP_SYST = 999.99. or BP_DIAST = 999.99.
BP_MEDEFF_HBP	integer	*Effective antihypertensive drug treatment.* If EXAM = 1 or QUEST = 1: 1 (yes) if BP_SYST < 140 and BP_DIAST < 90 and MEDICINE_HBP1 = 1. 2 (no) if (neither one is missing and (BP_SYST >= 140 or BP_DIAST >= 90)) and MEDICINE_HBP1 = 1. 9 (missing) if MEDICINE_HBP1 = 1 and (BP_SYST = missing or BP_DIAST = missing). . (irrelevant) if MEDICINE_HBP1 != 1.

Lipids

COLUMN NAME	TYPE	DEFINITION
BS_CHOLHDL_ RATIO	decimal	*Ratio of serum total cholesterol and HDL cholesterol.* If EXAM = 1 or QUEST = 1: BS_CHOL_MMOL divided by BS_HDL_MMOL if BS_CHOL_ MMOL and BS_HDL_MMOL ≠ 9999.9999. 9999.9999 (missing) if EXAM = 2 or BS_CHOL_MMOL = 9999.9999 or BS_HDL_MMOL = 9999.9999.
BS_ECHOL6	integer	*Elevated serum total cholesterol, different cutpoints: 6 mmol/l*. If EXAM = 1 or QUEST = 1: 1 (yes) if BS_CHOL_MMOL >= 6. 2 (no) if BS_CHOL_MMOL < 6. 9 (missing) if EXAM = 2 or BS_ CHOL_MMOL = 9999.9999.
BS_ECHOL7	integer	*Elevated serum total cholesterol, different cutpoints: 7 mmol/l.* If EXAM = 1 or QUEST = 1: 1 (yes) if BS_CHOL_MMOL >= 7. 2 (no) if BS_CHOL_MMOL < 7. 9 (missing) if EXAM = 2 or BS_ CHOL_MMOL = 9999.9999.
BS_ECHOL8	integer	*Elevated serum total cholesterol, different cutpoints: 8 mmol/l.* If EXAM = 1 or QUEST = 1: 1 (yes) if BS_CHOL_MMOL >= 8. 2 (no) if BS_CHOL_MMOL < 8. 9 (missing) if EXAM = 2 or BS_ CHOL_MMOL = 9999.9999.
BS_CHOL_ CATEGORY	integer	*Categories of serum total cholesterol.* If EXAM = 1 or QUEST = 1: 1 if BS_CHOL_MMOL < 5. 2 if BS_CHOL_MMOL >= 5 and BS_CHOL_MMOL < 6. 3 if BS_CHOL_MMOL >= 6 and BS_CHOL_MMOL < 7. 4 if BS_CHOL_MMOL >= 7 and BS_CHOL_MMOL < 8. 5 if BS_CHOL_MMOL >= 8. 9 (missing) if EXAM = 2 or BS_ CHOL_MMOL = 9999.9999

COLUMN NAME	TYPE	DEFINITION
BS_LHDL	integer	*Low serum HDL cholesterol*. If EXAM = 1 or QUEST = 1: 1 (yes) if (SEX = 1 (male) and BS_CHOL_MMOL < 1) or (SEX = 2 and BS_CHOL_MMOL < 1.2). 2 (no) if (SEX = 1 and BS_CHOL_MMOL >= 1) or (SEX = 2 and BS_CHOL_MMOL >= 1.2). 9 (missing) if EXAM = 2 or BS_CHOL_MMOL = 9999.9999 or (SEX != 1 and SEX != 2). (empty value), if ELIG ≠ 1 or EXAM = 2.
BS_HRATIO	integer	*High serum total cholesterol to HDL cholesterol ratio.* If EXAM = 1 or QUEST = 1: 1 (yes) if BS_CHOLHDL_RATIO > 5. 2 (no) if BS_CHOLHDL_RATIO <= 5. 9 (missing) if EXAM = 2 or BS_CHOLHDL_RATIO = 9999.9999.
BS_ECHOL	integer	*Elevated serum total cholesterol*. If EXAM = 1 or QUEST = 1: 1 (yes) if BS_CHOL_MMOL >= 5 . 2 (no) if BS_CHOL_MMOL < 5. 9 (missing) if EXAM = 2 or BS_CHOL_MMOL = 9999.9999.
BS_ENONHDL	integer	*Elevated non-HDL cholesterol*. If EXAM = 1 or QUEST = 1: 1 (yes) if BS_NONHDL > 3.5. 2 (no) if BS_NONHDL <= 3.5. 9 (missing) if EXAM = 2 or BS_NONHDL = 9999.9999.

COLUMN NAME	TYPE	DEFINITION
BS_HNONHDL	integer	*Actual or potential elevated non-HDL cholesterol*. If EXAM = 1 or QUEST = 1: 1 (yes) if neither one is missing and (BS_NONHDL > 3.5 or MEDICINE_CHOL1 = 1) . 2 (no) if neither one is missing and (BS_NONHDL <= 3.5 and MEDICINE_CHOL1 = 2). 9 (missing) if EXAM = 2 or QUEST = 2 or BS_NONHDL = 9999.9999 or (MEDICINE_CHOL1 != 1 and MEDICINE_CHOL1 != 2).
BS_MEDEFF_ CHOL	integer	*Effective cholesterol treatment*. If EXAM = 1 or QUEST = 1: 1 (yes) if BS_CHOL_MMOL < 5 and MEDICINE_CHOL1 = 1. 2 (no) if BS_CHOL_MMOL >= 5 and MEDICINE_CHOL1 = 1. 9 (missing) if MEDICINE_CHOL1 = 1 and BS_CHOL_MMOL = 9999.9999. . (irrelevant) if MEDICINE_CHOL1 != 1.

Glucose

COLUMN NAME	TYPE	DEFINITION
BS_GLYCEMIA	integer	*Provisional diagnosis of impaired fasting glucose (IFG)*. If EXAM = 1 or QUEST = 1: 1 (yes) if 6.0 < BS_FGLC_MMOL < 7.0. 2 (no) if BS_FGLC_MMOL <= 6.0 or BS_FGLC_MMOL >= 7.0. 9 (missing) if EXAM = 2 or BS_ FGLC_MMOL = 9999.9999.
BS_PROVDIAB_ GLC	integer	*Provisional diagnosis of diabetes based on fasting plasma glucose*. If EXAM = 1 or QUEST = 1: 1 (yes) if BS_FGLC_MMOL >= 7.0. 2 (no) if BS_FGLC_MMOL < 7.0. 9 (missing) if EXAM = 2 or BS_ FGLC_MMOL = 9999.9999.

COLUMN NAME	TYPE	DEFINITION
BS_PROVDIAB_ HBA1C	integer	*Provisional diagnosis of diabetes based on HbA1c*. If EXAM = 1 or QUEST = 1: 1 (yes) if BS_HBA1C_MMOL >= 48. 2 (no) if BS_HBA1C_MMOL < 48. 9 (missing) if EXAM = 2 or BS_ HBA1C_MMOL = 9999.9999.
BS_DIAB_SR	integer	*Actual or potential diabetes based on fasting glucose or HbA1c or self-reported status.* If EXAM = 1 or QUEST = 1: 1 (yes) if neither one is missing and (BS_DIAB_MSR = 1 or DIAGN_DIAB = 1). 2 (no) if BS_DIAB_MSR = 2 and DIAGN_DIAB = 2. 9 (missing) if EXAM = 2 or QUEST = 2 or (BS_DIAB_MSR != 1 and BS_DIAB_MSR != 2 and DIAGN_DIAB != 1 and DIAGN_ DIAB != 2).
BS_MEDEFF_ DIAB	integer	*Effective diabetes treatment*. If EXAM = 1 or QUEST = 1: 1 (yes) if (BS_HBA1C_MMOL < 48 or (BS_HBA1C_ MMOL=9999.9999 and BS_FGLC_MMOL < 6.0)) and MEDICINE_DIAB1 = 1. 2 (no) if (BS_HBA1C_MMOL >= 48 or (BS_HBA1C_ MMOL=9999.9999 and BS_ FGLC_MMOL >= 6.0)) and MEDICINE_DIAB1 = 1. 9 (missing) if MEDICINE_DIAB1 = 1 and BS_FGLC_MMOL = 9999.9999 and BS_HBA1C_ MMOL= 9999.9999. . (irrelevant) if MEDICINE_DIAB1 != 1 .

COLUMN NAME	TYPE	DEFINITION
BS_DIAB_SR_AWARE	integer	*Awareness of diabetes among actual or potential diabetics (incl. self-reported status).* If EXAM = 1 or QUEST = 1: 1 (yes) if BS_DIAB_SR = 1 and DIAGN_DIAB = 1. 2 (no) if BS_DIAB_SR = 1 and (DIAGN_DIAB **=**2 or DIAGN_DIAB = 8). 9 (missing) if BS_DIAB_SR = 1 and DIAGN_DIAB != 1 and DIAGN_DIAB != 2 and DIAGN_DIAB != 8. . (irrelevant) if BS_DIAB_SR != 1.

Anthropometrics and obesity

COLUMN NAME	TYPE	DEFINITION
HEIGHT_DIFF	decimal	*Difference between measured and self-reported height (cm).* If EXAM = 1 or QUEST = 1: SHEIGHT - HEIGHT. 999.9999 (missing) if EXAM = 2 or QUEST = 2 or HEIGHT = 999.99 or SHEIGHT = (9 or 8).
WEIGHT_DIFF	decimal	*Difference between measured and self-reported weight (kg).* If EXAM = 1 or QUEST = 1: SWEIGHT - WEIGHT. 999.9999 (missing) if EXAM = 2 or QUEST = 2 or WEIGHT = 999.99 or SWEIGHT = (9 or 8).
BMI_DIFF	decimal	*Difference between measured and self-reported BMI (kg/m2).* If EXAM = 1 or QUEST = 1: BMI_SR - BMI if ELIG = 1 and EXAM = 1 and QUEST = 1. 999.9999 (missing) if BMI = 999.99 or BMI_SR = 999.99 and ELIG = 1 and EXAM = 1 and QUEST = 1. (empty value), if ELIG ≠ 1 or EXAM =2 or QUEST = 2.

COLUMN NAME	TYPE	DEFINITION
BMI_CATEGORY	integer	*BMI categories based on measured BMI*. If EXAM = 1 or QUEST = 1: 1 (underweight) if BMI < 18.5. 2 (normal) if 18.5 <= BMI < 25. 3 (overweight) if 25 <= BMI < 30. 4 (grade 1 obese) if 30 <= BMI < 35. 5 (grade 2 overweight) if 35 <= BMI < 40. 6 (grade 3 overweight) if BMI >= 40. 9 (missing) if EXAM = 2 or BMI = 999.99.
BMI_ OVERWEIGHT_SR	integer	*Overweight based on self-reported BMI*. If EXAM = 1 or QUEST = 1: 1 (yes) if BMI_SR >= 25. 2 (no) if BMI_SR < 25. 9 (missing) if QUEST = 2 or BMI_ SR = 999.99.
BMI_OBESE_SR	integer	*Obesity based on self-reported BMI*. If EXAM = 1 or QUEST = 1: 1 (yes) if BMI_SR >= 30. 2 (no) if BMI_SSR < 30. 9 (missing) if QUEST = 2 or BMI_ SR = 999.99.
BMI_CATEGORY_SR	integer	*BMI categories based on self-reported BMI*. If EXAM = 1 or QUEST = 1: 1 (underweight) if BMI_SR < 18.5. 2 (normal) if 18.5 <= BMI_SR < 25. 3 (overweight) if 25 <= BMI_SR < 30. 4 (grade 1 obese) if 30 <= BMI_ SR < 35. 5 (grade 2 overweight) if 35 <= BMI_SR < 40. 6 (grade 3 overweight) if BMI_SR >= 40. 9 (missing) if QUEST = 2 or BMI_ SR = 999.99.

Smoking

COLUMN NAME	TYPE	DEFINITION
SMOKEY	integer	*Smoking years among all survey participants*. If EXAM = 1 or QUEST = 1: SMOKE_YEARS if SMOKE = 1 or SMOKE_EVER = 1. 0, if (SMOKE = 2 or SMOKE = 3) and SMOKE_EVER = 2. 99 (missing) if QUEST = 2 or SMOKE_YEARS = 99 or (SMOKE != 1 and SMOKE != 2 and SMOKE != 3 and SMOKE_EVER != 1 and SMOKE_EVER != 2).

Eligibility and consent items

COLUMN NAME	TYPE	DEFINITION
EXAM1	integer	*Participation in examination*. EXAM if ELIG = 1. . (irrelvant) if ELIG != 1.
QUEST1	integer	*Participation in questionnaire*. QUEST if ELIG = 1. . (irrelvant) if ELIG != 1.
REASON1	integer	*Reason for non-eligibility.* REASON if ELIG = 2. . (irrelvant) if ELIG != 2.
EXAM_REF1	integer	*Refusal to participate to the examination.* EXAM_REF if EXAM1 = 2. . (irrelvant) if EXAM1 != 2.
EXAM_NOAPP1	integer	*Interested but failed to make appointment for examination.* EXAM_NOAPP if EXAM_REF1 != 2. . (irrelvant) if EXAM_REF1 = 2.

Appendix 5c. Specification of indicators

For further information on indicators, see EHES Manual, Part C, Chapter 4.

Field name refer to the flag variable of the EHES database and variable names and values are defined in the EHES Manual, Part C, Chapter 2 for transferred data items and in the Appendix 5b. for the derived variables.

Each indicator are calculated for men and women separately by 10-year age-groups and age-standardized for 25-64-year-olds. Other subgroups, e.g. socioeconomic status, can also be used. Indicators marked as green (❖) are those that at least should be included in the basic report.

Blood pressure indicators

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
n_nonmissing	*Number of individuals with non-missing value.* Variables: BP_DIAST - Mean of diastolic blood pressure (mmHg). ❖ BP_SYST - Mean of systolic blood pressure (mmHg). ❖ BP_PULSE - Pulse rate (60 seconds).
mean	*Population mean*. Variables: BP_DIAST - Mean of diastolic blood pressure (mmHg). ❖ BP_SYST - Mean of systolic blood pressure (mmHg). ❖ BP_PULSE - Pulse rate (60 seconds).
se	*Standard error of the mean*. Variables: BP_DIAST - Mean of diastolic blood pressure (mmHg). ❖ BP_SYST - Mean of systolic blood pressure (mmHg). ❖ BP_PULSE - Pulse rate (60 seconds).
cil	*Lower limit of 95 % confidence interval of the mean*. Variables: BP_DIAST - Mean of diastolic blood pressure (mmHg). ❖ BP_SYST - Mean of systolic blood pressure (mmHg). ❖ BP_PULSE - Pulse rate (60 seconds).

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
ciu	*Upper limit of 95 % confidence interval of the mean*. Variables: BP_DIAST - Mean of diastolic blood pressure (mmHg). ❖ BP_SYST - Mean of systolic blood pressure (mmHg). ❖ BP_PULSE - Pulse rate (60 seconds).
	Hypertension:
n	*Number of individuals in each category.* Variables: BP_AWARE - Awareness of hypertension among actual or potential hypertensives (140/90). ❖ BP_HBP - Actual or potential hypertension (140/90). ❖ BPM - When was blood pressure last measured by a health professional. 1 = Within the past 12 moths, 2 = 1-5 years ago, 3 = Never or more than 5 years ago. ❖ BP_OPTIMAL - Optimal blood pressure among actual or potential hypertension (140/90). ❖ DIAGN_HBP - High blood pressure (hypertension) diagnosed by medical doctor. ❖ MEDICINE_HBP_POP - Medicines for high blood pressure during the past 2 two weeks among population. ❖ MEDICINE_HBP_ILL - Medicines for high blood pressure during the past 2 two weeks among actual or potential hypertensives. ❖ BP_CATEGORY - Categories of blood pressure: 1 = Optimal, 2 = Normal, 3 = High normal, 4 = Grade 1 hypertension, 5 = Grade 2 Hypertension, 6 = Grade 3 Hypertension. BP_EBP - Elevated blood pressure. BP_HSBP - Systolic hypertension. BP_MEDEFF_HBP - Effective antihypertensive drug treatment.

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
percent	*Proportion of each category among population with non-missing value.* Variables: BP_AWARE - Awareness of hypertension among actual or potential hypertensives (140/90). ❖ BP_HBP - Actual or potential hypertension (140/90). ❖ BPM - When was blood pressure last measured by a health professional. 1 = Within the past 12 moths, 2 = 1-5 years ago, 3 = Never or more than 5 years ago. ❖ BP_OPTIMAL - Optimal blood pressure among actual or potential hypertension (140/90). ❖ DIAGN_HBP - High blood pressure (hypertension) diagnosed by medical doctor. ❖ MEDICINE_HBP_POP - Medicines for high blood pressure during the past 2 two weeks among population. ❖ MEDICINE_HBP_ILL - Medicines for high blood pressure during the past 2 two weeks among actual or potential hypertensives. ❖ BP_CATEGORY - Categories of blood pressure: 1 = Optimal, 2 = Normal, 3 = High normal, 4 = Grade 1 hypertension, 5 = Grade 2 Hypertension, 6 = Grade 3 Hypertension. BP_EBP - Elevated blood pressure. BP_HSBP - Systolic hypertension. BP_MEDEFF_HBP - Effective antihypertensive drug treatment.

Lipid indicators

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
n_nonmissing	*Number of individuals with non-missing value.* Variables: BS_CHOL_MMOL - Serum total cholesterol in mmol/l. ❖ BS_HDL_MMOL - Serum HDL cholesterol in mmol/l. ❖ BS_NONHDL - Serum non-HDL cholesterol (mmol/l). ❖ BS_CHOLHDL_RATIO - Ratio of serum total cholesterol and HDL cholesterol.
mean	*Population mean*. Variables: BS_CHOL_MMOL - Serum total cholesterol in mmol/l. ❖ BS_HDL_MMOL - Serum HDL cholesterol in mmol/l. ❖ BS_NONHDL - Serum non-HDL cholesterol (mmol/l). ❖ BS_CHOLHDL_RATIO - Ratio of serum total cholesterol and HDL cholesterol.

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
se	*Standard error of the mean*. Variables: BS_CHOL_MMOL - Serum total cholesterol in mmol/l. ❖ BS_HDL_MMOL - Serum HDL cholesterol in mmol/l. ❖ BS_NONHDL - Serum non-HDL cholesterol (mmol/l). ❖ BS_CHOLHDL_RATIO - Ratio of serum total cholesterol and HDL cholesterol.
cil	*Lower limit of 95 % confidence interval of the mean*. Variables: BS_CHOL_MMOL - Serum total cholesterol in mmol/l. ❖ BS_HDL_MMOL - Serum HDL cholesterol in mmol/l. ❖ BS_NONHDL - Serum non-HDL cholesterol (mmol/l). ❖ BS_CHOLHDL_RATIO - Ratio of serum total cholesterol and HDL cholesterol.
ciu	*Upper limit of 95 % confidence interval of the mean*. Variables: BS_CHOL_MMOL - Serum total cholesterol in mmol/l. ❖ BS_HDL_MMOL - Serum HDL cholesterol in mmol/l. ❖ BS_NONHDL - Serum non-HDL cholesterol (mmol/l). ❖ BS_CHOLHDL_RATIO - Ratio of serum total cholesterol and HDL cholesterol.

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
n	*Number of individuals in each category.* Variables: BS_CHOL_AWARE - Awareness of elevated cholesterol among those with actual or potential elevated serum total cholesterol. 1 = Yes, 2 = No. ❖ BS_CHOL_OPTIMAL - Optimal total cholesterol level (<5mmol/l) among those using lipid lowering drugs. 1 = Yes, 2 = No. ❖ BS_HCHOL - Actual or potential elevated serum total cholesterol. 1 = Yes, 2 = No. ❖ CHOLM - When was blood cholesterol last measured. 1 = Within the past 12 moths, 2 = 1-5 years ago, 3 = Never or more than 5 years ago. ❖ DIAGN_CHOL - Elevated blood cholesterol diagnosed by medical doctor: 1 = Yes, 2 = No. ❖ MEDICINE_CHOL_ILL - Medicines for lowering the blood cholesterol level during the past 2 two weeks among those with actual or potential elevated serum total cholesterol. 1 = Yes, 2 = No. ❖ MEDICINE_CHOL_POP - Medicines for lowering the blood cholesterol level during the past 2 two weeks among population. 1 = Yes, 2 = No. ❖ BS_CHOL_CATEGORY - Categories of serum total cholesterol. 1 = below 5 mmol/l, 2 = at least 5 and below 6, 3 = at least 6 and below 7, 4 = at least 7 and below 8, 5 = at least 8 mmol/l. BS_ECHOL - Elevated serum total cholesterol, cutpoint: 5 mmol/l. 1 = Yes, 2 = No. BS_ENONHDL - Elevated non-HDL cholesterol (>3.5mmol/l): 1 = Yes, 2 = No. BS_HNONHDL - Actual or potential elevated non-HDL cholesterol: 1 = Yes, 2 = No. BS_HRATIO - High serum total cholesterol to HDL cholesterol ratio (>5). 1 = Yes, 2 = No. BS_LHDL - Low serum HDL cholesterol <1mmol/l for men <1.2mmol/l for women): 1 = Yes, 2 = No. BS_MEDEFF_CHOL - Effective cholesterol treatment. 1 = Yes, 2 = No.

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
percent	*Proportion of each category among population with non-missing value.* Variables: BS_CHOL_AWARE - Awareness of elevated cholesterol among those with actual or potential elevated serum total cholesterol. 1 = Yes, 2 = No. ❖ BS_CHOL_OPTIMAL - Optimal total cholesterol level (<5mmol/l) among those using lipid lowering drugs. 1 = Yes, 2 = No. ❖ BS_HCHOL - Actual or potential elevated serum total cholesterol. 1 = Yes, 2 = No. ❖ CHOLM - When was blood cholesterol last measured. 1 = Within the past 12 moths, 2 = 1-5 years ago, 3 = Never or more than 5 years ago. ❖ DIAGN_CHOL - Elevated blood cholesterol diagnosed by medical doctor: 1 = Yes, 2 = No. ❖ MEDICINE_CHOL_ILL - Medicines for lowering the blood cholesterol level during the past 2 two weeks among those with actual or potential elevated serum total cholesterol. 1 = Yes, 2 = No. ❖ MEDICINE_CHOL_POP - Medicines for lowering the blood cholesterol level during the past 2 two weeks among population. 1 = Yes, 2 = No. ❖ BS_CHOL_CATEGORY - Categories of serum total cholesterol. 1 = below 5 mmol/l, 2 = at least 5 and below 6, 3 = at least 6 and below 7, 4 = at least 7 and below 8, 5 = at least 8 mmol/l. BS_ECHOL - Elevated serum total cholesterol, cutpoint: 5 mmol/l. 1 = Yes, 2 = No. BS_ENONHDL - Elevated non-HDL cholesterol (>3.5mmol/l): 1 = Yes, 2 = No. BS_HNONHDL - Actual or potential elevated non-HDL cholesterol: 1 = Yes, 2 = No. BS_HRATIO - High serum total cholesterol to HDL cholesterol ratio (>5). 1 = Yes, 2 = No. BS_LHDL - Low serum HDL cholesterol (<1mmol/l for men <1.2mmol/l for women): 1 = Yes, 2 = No. BS_MEDEFF_CHOL - Effective cholesterol treatment. 1 = Yes, 2 = No.

Glucose and diabetes indicators

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
n_nonmissing	*Number of individuals with non-missing value.* Variables: BS_FGLC_MMOL - Fasting plasma glucose in mmol/l. Relevant if BS_FAST = Yes. ❖ BS_HBA1C_MMOL - Whole blood glycated haemoglobin concentration in mmol/mol. ❖
mean	*Population mean*. Variables: BS_FGLC_MMOL - Fasting plasma glucose in mmol/l. Relevant if BS_FAST = Yes. ❖ BS_HBA1C_MMOL - Whole blood glycated haemoglobin concentration in mmol/mol. ❖
se	*Standard error of the mean*. Variables: BS_FGLC_MMOL - Fasting plasma glucose in mmol/l. Relevant if BS_FAST = Yes. ❖ BS_HBA1C_MMOL - Whole blood glycated haemoglobin concentration in mmol/mol. ❖
cil	*Lower limit of 95 % confidence interval of the mean*. Variables: BS_FGLC_MMOL - Fasting plasma glucose in mmol/l. Relevant if BS_FAST = Yes. ❖ BS_HBA1C_MMOL - Whole blood glycated haemoglobin concentration in mmol/mol. ❖
ciu	*Upper limit of 95 % confidence interval of the mean*. Variables: BS_FGLC_MMOL - Fasting plasma glucose in mmol/l. Relevant if BS_FAST = Yes. ❖ BS_HBA1C_MMOL - Whole blood glycated haemoglobin concentration in mmol/mol. ❖

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
n	*Number of individuals in each category.* Variables: BS_DIAB - Actual or potential diabetes based on fasting glucose or HbA1c or use of medication: 1 = Yes, 2 = No. ❖ BS_DIAB_AWARE - Awareness of diabetes among actual or potential diabetics (incl. use of medication). 1 = Yes, 2 = No. ❖ BS_GLC_OPTIMAL - Optimal fasting glucose or HbA1C level (<7mmol/l or <48mmol/mol) among those using diabetes medication. 1 = Yes, 2 = No. ❖ DIAGN_DIAB - Diabetes diagnosed by medical doctor: 1 = Yes, 2 = No, 8 = Do not know. ❖ GLCM - When was blood sugar (glucose) last measured. 1 = Within the past 12 moths, 2 = 1-5 years ago, 3 = Never or more than 5 years ago. ❖ MEDICINE_DIAB_ILL - Medicines for diabetes during the past 2 two weeks among actual or potential diabetics. 1 = Yes, 2 = No. ❖ MEDICINE_DIAB_POP - Medicines for diabetes during the past 2 two weeks among population. 1 = Yes, 2 = No. ❖ BS_DIAB_MSR - Actual or potential diabetes based on fasting glucose or HbA1c: 1 = Yes, 2 = No. BS_DIAB_MSR2 - Actual or potential diabetes based on fasting glucose or HbA1c, lower limits. 1 = Yes, 2 = No. BS_DIAB_SR - Actual or potential diabetes based on fasting glucose or HbA1c or self-reported status: 1 = Yes, 2 = No. BS_GLYCEMIA - Provisional diagnosis of impaired fasting glucose (IFG): 1 = Yes, 2 = No. BS_DIAB_SR_AWARE - Awareness of diabetes among actual or potential diabetics (incl. self-reported status). 1 = Yes, 2 = No. BS_MEDEFF_DIAB - Effective diabetes treatment: 1 = Yes, 2 = No. BS_PROVDIAB_GLC - Provisional diagnosis of diabetes based on fasting plasma glucose: 1 = Yes, 2 = No. BS_PROVDIAB_HBA1C - Provisional diagnosis of diabetes based on HbA1c: 1 = Yes, 2 = No.

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
percent	*Proportion of each category among population with non-missing value.* Variables: BS_DIAB - Actual or potential diabetes based on fasting glucose or HbA1c or use of medication: 1 = Yes, 2 = No. ❖ BS_DIAB_AWARE - Awareness of diabetes among actual or potential diabetics (incl. use of medication). 1 = Yes, 2 = No. ❖ BS_GLC_OPTIMAL - Optimal fasting glucose or HbA1C level (<7mmol/l or <48mmol/mol) among those using diabetes medication. 1 = Yes, 2 = No. ❖ DIAGN_DIAB - Diabetes diagnosed by medical doctor: 1 = Yes, 2 = No, 8 = Do not know. ❖ GLCM - When was blood sugar (glucose) last measured. 1 = Within the past 12 moths, 2 = 1-5 years ago, 3 = Never or more than 5 years ago. ❖ MEDICINE_DIAB_ILL - Medicines for diabetes during the past 2 two weeks among actual or potential diabetics. 1 = Yes, 2 = No. ❖ MEDICINE_DIAB_POP - Medicines for diabetes during the past 2 two weeks among population. 1 = Yes, 2 = No. ❖ BS_DIAB_MSR - Actual or potential diabetes based on fasting glucose or HbA1c: 1 = Yes, 2 = No. BS_DIAB_MSR2 - Actual or potential diabetes based on fasting glucose or HbA1c, lower limits. 1 = Yes, 2 = No. BS_DIAB_SR - Actual or potential diabetes based on fasting glucose or HbA1c or self-reported status: 1 = Yes, 2 = No. BS_GLYCEMIA - Provisional diagnosis of impaired fasting glucose (IFG): 1 = Yes, 2 = No. BS_DIAB_SR_AWARE - Awareness of diabetes among actual or potential diabetics (incl. self-reported status). 1 = Yes, 2 = No. BS_MEDEFF_DIAB - Effective diabetes treatment: 1 = Yes, 2 = No. BS_PROVDIAB_GLC - Provisional diagnosis of diabetes based on fasting plasma glucose: 1 = Yes, 2 = No. BS_PROVDIAB_HBA1C - Provisional diagnosis of diabetes based on HbA1c: 1 = Yes, 2 = No.

Anthropometric and obesity indicators

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
n_nonmissig	*Number of individuals with non-missing value.* Variables: BMI - Body mass index (kg/m2) based on measured height and weight. ❖ BMI_SR - Body mass index (kg/m2) based on self-reported height and weight. ❖ WAIST - Waist circumference in cm. ❖ BMI_DIFF - Difference between measured and self-reported BMI in kg/m2 (self-reported - measured). HEIGHT - Height in cm. HEIGHT_DIFF - Difference between measured and self-reported height in cm (self-reported - measured). SHEIGHT - Self-reported height (cm). SWEIGHT - Self-reported weight (kg). WEIGHT - Weight in kg. WEIGHT_DIFF - Difference between measured and self-reported weight in kg (self-reported - measured).
mean	*Population mean*. Variables: BMI - Body mass index (kg/m2) based on measured height and weight. ❖ BMI_SR - Body mass index (kg/m2) based on self-reported height and weight. ❖ WAIST - Waist circumference in cm. ❖ BMI_DIFF - Difference between measured and self-reported BMI in kg/m2 (self-reported - measured). HEIGHT - Height in cm. HEIGHT_DIFF - Difference between measured and self-reported height in cm (self-reported - measured). SHEIGHT - Self-reported height (cm). SWEIGHT - Self-reported weight (kg). WEIGHT - Weight in kg. WEIGHT_DIFF - Difference between measured and self-reported weight in kg (self-reported - measured).

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
se	*Standard error of the mean*. Variables: BMI - Body mass index (kg/m2) based on measured height and weight. ❖ BMI_SR - Body mass index (kg/m2) based on self-reported height and weight. ❖ WAIST - Waist circumference in cm. ❖ BMI_DIFF - Difference between measured and self-reported BMI in kg/m2 (self-reported - measured). HEIGHT - Height in cm. HEIGHT_DIFF - Difference between measured and self-reported height in cm (self-reported - measured). SHEIGHT - Self-reported height (cm). SWEIGHT - Self-reported weight (kg). WEIGHT - Weight in kg. WEIGHT_DIFF - Difference between measured and self-reported weight in kg (self-reported - measured).
cil	*Lower limit of 95 % confidence interval of the mean*. Variables: BMI - Body mass index (kg/m2) based on measured height and weight. ❖ BMI_SR - Body mass index (kg/m2) based on self-reported height and weight. ❖ WAIST - Waist circumference in cm. ❖ BMI_DIFF - Difference between measured and self-reported BMI in kg/m2 (self-reported - measured). HEIGHT - Height in cm. HEIGHT_DIFF - Difference between measured and self-reported height in cm (self-reported - measured). SHEIGHT - Self-reported height (cm). SWEIGHT - Self-reported weight (kg). WEIGHT - Weight in kg. WEIGHT_DIFF - Difference between measured and self-reported weight in kg (self-reported - measured).

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
ciu	*Upper limit of 95 % confidence interval of the mean*. Variables: BMI - Body mass index (kg/m2) based on measured height and weight. ❖ BMI_SR - Body mass index (kg/m2) based on self-reported height and weight. ❖ WAIST - Waist circumference in cm. ❖ BMI_DIFF - Difference between measured and self-reported BMI in kg/m2 (self-reported - measured). HEIGHT - Height in cm. HEIGHT_DIFF - Difference between measured and self-reported height in cm (self-reported - measured). SHEIGHT - Self-reported height (cm). SWEIGHT - Self-reported weight (kg). WEIGHT - Weight in kg. WEIGHT_DIFF - Difference between measured and self-reported weight in kg (self-reported - measured).
n	*Number of individuals in each category.* Variables: BMI_OBESE - Obesity based on measured BMI (>=30): 1 = Yes, 2 = No. ❖ BMI_OVERWEIGHT - Overweight based on measured BMI (>=25): 1 = Yes, 2 = No. ❖ WAIST_LARGE - Large waist circumference (>102cm for men, >88cm for women). 1 = Yes, 2 = No. ❖ BMI_CATEGORY - BMI categories based on measured BMI. 1 = Underweight (<18.5), 2 = Normal (18.5<=BMI<25), 3 = Overweight (25<=BMI<30), 4 = Grade 1 obese (30<=BMI<35), 5 = Grade 2 obese (35<=BMI<40), 6 = Grade 3 obese (BMI>=40). BMI_CATEGORY_SR - BMI categories based on self-reported BMI. 1 = Underweight (<18.5), 2 = Normal (18.5<=BMI<25), 3 = Overweight (25<=BMI<30), 4 = Grade 1 obese (30<=BMI<35), 5 = Grade 2 obese (35<=BMI<40), 6 = Grade 3 obese (BMI>=40). BMI_OBESE_SR - Obesity based on self-reported BMI (>=30): 1 = Yes, 2 = No. BMI_OVERWEIGHT_SR - Overweight based on self-reported BMI (>=25): 1 = Yes, 2 = No.

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
percent	*Proportion of each category among population with non-missing value.* Variables: BMI_OBESE - Obesity based on measured BMI (>=30): 1 = Yes, 2 = No. ❖ BMI_OVERWEIGHT - Overweight based on measured BMI (>=25): 1 = Yes, 2 = No. ❖ WAIST_LARGE - Large waist circumference (>102cm for men, >88cm for women). 1 = Yes, 2 = No. ❖ BMI_CATEGORY - BMI categories based on measured BMI. 1 = Underweight (<18.5), 2 = Normal (18.5<=BMI<25), 3 = Overweight (25<=BMI<30), 4 = Grade 1 obese (30<=BMI<35), 5 = Grade 2 obese (35<=BMI<40), 6 = Grade 3 obese (BMI>=40). BMI_CATEGORY_SR - BMI categories based on self-reported BMI. 1 = Underweight (<18.5), 2 = Normal (18.5<=BMI<25), 3 = Overweight (25<=BMI<30), 4 = Grade 1 obese (30<=BMI<35), 5 = Grade 2 obese (35<=BMI<40), 6 = Grade 3 obese (BMI>=40). BMI_OBESE_SR - Obesity based on self-reported BMI (>=30): 1 = Yes, 2 = No. BMI_OVERWEIGHT_SR - Overweight based on self-reported BMI (>=25): 1 = Yes, 2 = No.

Self-perceived health and chronic disease indicators

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
n	*Number of individuals in each category.* Variables: HEALTH_GOOD - Self-perceived health reported to be good or very good. 1 = Yes, 2 = No. ❖ LIMITED_YES - Person reported that she has limitations in daily activities due to health problems. 1 = Yes, 2 = No. ❖ HEALTH_PROBLEM - Longstanding illness or health problem. 1 = Yes, 2 = No. DIAGN_MI - Myocardial infarction diagnosed by medical doctor: 1 = Yes, 2 = No. DIAGN_STROKE - Stroke (cerebral haemorrhage, cerebral thrombosis) diagnosed by medical doctor: 1 = Yes, 2 = No. DIAGN_CHD - Coronary heart disease (angina pectoris) diagnosed by medical doctor: 1 = Yes, 2 = No.

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
percent	*Proportion of each category among population with non-missing value.* Variables: HEALTH_GOOD - Self-perceived health reported to be good or very good. 1 = Yes, 2 = No. ❖ LIMITED_YES - Person reported that she has limitations in daily activities due to health problems. 1 = Yes, 2 = No. ❖ HEALTH_PROBLEM - Longstanding illness or health problem. 1 = Yes, 2 = No.DIAGN_MI - Myocardial infarction diagnosed by medical doctor: 1 = Yes, 2 = No. DIAGN_STROKE - Stroke (cerebral haemorrhage, cerebral thrombosis) diagnosed by medical doctor: 1 = Yes, 2 = No. DIAGN_CHD - Coronary heart disease (angina pectoris) diagnosed by medical doctor: 1 = Yes, 2 = No.

Smoking indicators

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
n_nonmissing	*Number of individuals with non-missing value.* Variables: SMOKEY - Smoking years in whole population. SMOKE_YEARS - Daily smoking years.
mean	*Population mean*. Variables: SMOKEY - Smoking years in whole population. SMOKE_YEARS - Daily smoking years.
se	*Standard error of the mean*. Variables: SMOKEY - Smoking years in whole population. SMOKE_YEARS - Daily smoking years.
cil	*Lower limit of 95 % confidence interval of the mean*. Variables: SMOKEY - Smoking years in whole population. SMOKE_YEARS - Daily smoking years.
ciu	*Upper limit of 95 % confidence interval of the mean*. Variables: SMOKEY - Smoking years in whole population. SMOKE_YEARS - Daily smoking years.
n	*Number of individuals in each category.* Variables: SMOKE - Current status of smoking. 1 = Daily, 2 = Occasionally, 3 = Not at all. ❖ SMOKE_EVER - Past daily or almost daily smoking amon non-daily smokers. 1 = Yes, 2 = No. ❖

FIELD NAME	DEFINITION AND VARIABLE NAME AND VALUES
percent	*Proportion of each category among population with non-missing value.* Variables: SMOKE - Current status of smoking. 1 = Daily, 2 = Occasionally, 3 = Not at all. (/)SMOKE_EVER - Past daily or almost daily smoking amon non-daily smokers. 1 = Yes, 2 = No. ❖

Background variables

VARIABLE	DEFINITION AND VALUES
Survey ❖	Identifies the survey.
Sex ❖	*Sex*: Male (SEX = 1). Female (SEX = 2).
Age group ❖	*Age in completed years at the date of the health examination.* 25-34 (25 <= AGE < 34). 35-44 (35 <= AGE < 45). 45-54 (45 <= AGE < 55). 55-64 (55 <= AGE < 65). 25-64 (25 <= AGE < 64, age-standardized using equal weights for age groups.
EDUCATION_CAT	*Highest level of completed education or training*. Low (EDUCATION = 1 or 2 or 3). Medium (EDUCATION = 4 or 5). High (EDUCATION = 6 or 7).
LABOUR_CAT	*Labour (employment) status*. Employed (LABOUR_STATUS = 1). Unemployed (LABOUR_STATUS = 2). Other (LABOUR_STATUS = 3 or 4 or 5 or 6 or 7 or 8).
SCHOOL_THIRDS	*Thirds of years of education*. 1st third (SCHOOL_THIRDS = 1). 2nd third (SCHOOL_THIRDS = 2). 3rd third (SCHOOL_THIRDS = 3).
COUPLE	*Living as a couple*. Yes (COUPLE = 1). No (COUPLE = 2.
INCOME_EQUI	*Household's income category (equivalised disposable income)*. 1st fifth (INCOME_EQUI = 1). 2nd fifth (INCOME_EQUI = 2). 3rd fifth (INCOME_EQUI = 3). 4th fifth (INCOME_EQUI = 4). 5th fifth (INCOME_EQUI = 5).

Appendix 5d. Program codes for basic reporting

Following R scripts are used in EHES basic reporting.

Database connection is established using R package RODBC (Brian Ripley and from 1999 to Oct 2002 Michael Lapsley (2012). RODBC: ODBC Database Access. R package version 1.3-5. http://CRAN.R-project.org/package=RODBC) and SQL (Structured Query Language) is used to import and export data from EHES database.

Program codes use database table variables for the definition of the EHES variables (e.g. code for missing value etc). The definition table variables is of the form:

VARNAME	Name of the variable
VARTYPE	Type of variable: Database, Sampling, Eligibility, Questionnaire, Measurement, Laboratory or Derived
DEFINITION	Definition of the variable and it's values
CONTINUOUS	1 if variable is continuous, 0 if variable is categorical
MISSING	Code for missing value if relevant for variable
DONTKNOW	Code for value "Don't know" if relevant for variable
IRRELEVANT	Code for irrelevant value if relevant for variable
EXAM	1 if variable uses information from examination
QUEST	1 if variable uses information from questionnaire
DIVIDER	For continuous variables, gives width of x axis for plotting
MINIMUM	For continuous variables, gives minimum value of x axis for plotting
MAXIMUM	For continuous variables, gives maximum value of x axis for plotting
MEAN	1 if continuous and variable is used for indicator calculation
PREVALENCE	1 if categorical and variable is used for indicator calculation

General functions

select_obs - to select relevant individuals for variable varname in the data using filters defined in variables

```
select_obs <- function(data, variables, varname) {
    ### data = dataset where to select observations, variables =
definition table for variables, varname = name of the variable
    irr = variables$IRRELEVANT[variables$VARNAME==varname]
    ## filtering individuals using participation indicator variables EXAM
and QUEST
    if (variables$EXAM[variables$VARNAME==varname]=="1" & variables$QUES
T[variables$VARNAME==varname]=="1") {
        selected <- data[data$EXAM == "1" & data$QUEST == "1" & !(data[,
varname] %in% irr) ,]
    }
    else if (variables$EXAM[variables$VARNAME==varname]=="1" ) {
        selected <- data[data$EXAM == "1" & !(data[, varname] %in% irr)
,]
    }
    else if (variables$QUEST[variables$VARNAME==varname]=="1" ) {
        selected <- data[data$QUEST == "1" & !(data[, varname] %in% irr)
,]
    }
    return(selected)

}
```

code_miss_numeric - to code missing and irrelevant values as NA and convert item to numeric

```
code_miss_numeric  <- function(data, item, mis, irr, filter) {
    ### data = dataset, item = name of the variable, mis = code for
missing value, irr = code for irrelevant value, filter = indicator for
participation
    attach(data)
    data[,item][!eval(filter)] <- ""
    data[ data[,item] %in% irr ,item] <- ""
    selection <- data[, item]
    detach(data)
    selection[ selection %in% mis ] <- NA
    selection = as.numeric(selection)
    return(selection)
}
```

Calculation of derived variables

derived_variables_bp - to calculate derived variables for blood pressure

```
derived_variables_bp <- function(data=NULL, variables) {
    ### data = dataset, variables = definition table for variables
    # *******************************************************************
```

```
********** #
    ############ SYSTOLIC BP, MEANS ###############
    # ******************************************************************
********** #
    tstart <- Sys.time()
    outdata=NULL
    if (is.null(data) ) {
        stop('\n***ERROR:***\n No data or database connection given.
Execution halted.')
    }
    cat('Creating variables: BP_SYSTM23, BP_SYSTM12, BP_SYSTM123, BP_
SYST', '\n', sep='')
    selected <- try(data[,  c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'BP_SYST1', 'BP_SYST2', 'BP_SYST3')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset ', data,
'\n', sep=""))
    }
    # to filter the participants
    filters <-  expression(EXAM == '1' | QUEST == '1')
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
    irr <- ""
    mis = variables$MISSING[variables$VARNAME == "BP_SYST1"]
    ## coding missing values and convert to numeric
    selected$BP_SYST1 = code_miss_numeric(selected, 'BP_SYST1', mis=mis,
irr=irr, filter=filters)
    selected$BP_SYST2 = code_miss_numeric(selected, 'BP_SYST2',
mis=mis, irr=irr, filter=filters)
    selected$BP_SYST3 = code_miss_numeric(selected, 'BP_SYST3',
mis=mis, irr=irr, filter=filters)
    ### BP_SYSTM23, BP_SYSTM12, BP_SYSTM123
    mis = variables$MISSING[variables$VARNAME == "BP_SYST"]
    irr = variables$IRRELEVANT[variables$VARNAME == "BP_SYST"]
    selected$BP_SYSTM23 <- format(round(rowMeans( selected[, c('BP_
SYST2', 'BP_SYST3')]), 2), nsmall=2, trim=T)
    attach(selected)
    selected$BP_SYSTM23[!eval(filters)] <- irr
    selected$BP_SYSTM23[eval(filters) & selected$BP_SYSTM23 == "NA"] <-
mis
    detach(selected)
    selected$BP_SYSTM12 <- format(round(rowMeans( selected[, c('BP_
SYST1', 'BP_SYST2')]), 2), nsmall=2, trim=T)
    attach(selected)
    selected$BP_SYSTM12[!eval(filters)] <- irr
    selected$BP_SYSTM12[eval(filters) & selected$BP_SYSTM12 == "NA"] <-
mis
    detach(selected)
    selected$BP_SYSTM123 <- format(round(rowMeans( selected[, c('BP_
SYST1', 'BP_SYST2', 'BP_SYST3')]), 2), nsmall=2, trim=T)
    attach(selected)
    selected$BP_SYSTM123[!eval(filters)] <- irr
    selected$BP_SYSTM123[eval(filters) & selected$BP_SYSTM123 == "NA"] <-
mis
```

```
    detach(selected)
    #### BP_SYST
    selected$BP_SYST <- mean_variables_bp(selected, item=c('BP_SYST') ,
variables=variables)
    attach(selected)
    selected$BP_SYST[!eval(filters)] <- irr
    selected$BP_SYST[eval(filters) & selected$BP_SYST == "NA"] <- mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'BP_SYSTM23', 'BP_SYSTM12', 'BP_SYSTM123', 'BP_SYST')],
TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
      outdata <- selected
    }
    selected = NULL
    filters = NULL

    # ******************************************************************
********** #
    ############ DIASTOLIC BP, MEANS ###############
    # ******************************************************************
********** #
    cat('Creating variables: BP_DIASTM23, BP_DIASTM12, BP_DIASTM123, BP_
DIAST', '\n', sep='')
    selected <- try(data[ , c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST',  'BP_DIAST1', 'BP_DIAST2', 'BP_DIAST3')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset\n',
sep=""))
    }
     ### filtering all derived variables as EXAM=1 or QUEST=1
    filters <-  expression(EXAM == '1' | QUEST == '1')
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
    irr <- variables$IRRELEVANT[variables$VARNAME == "BP_DIAST1"]
    mis = variables$MISSING[variables$VARNAME == "BP_DIAST1"]
    ## coding missing values and convert to numeric
    selected$BP_DIAST1 = code_miss_numeric(selected, 'BP_DIAST1', mis,
irr, filter=filters )
    selected$BP_DIAST2  = code_miss_numeric(selected, 'BP_DIAST2', mis,
irr, filter=filters)
    selected$BP_DIAST3  = code_miss_numeric(selected, 'BP_DIAST3', mis,
irr, filter=filters)
    ### BP_DIASTM23, BP_DIASTM12, BP_DIASTM123
    mis = variables$MISSING[variables$VARNAME == "BP_DIASTM23"]
    irr = variables$IRRELEVANT[variables$VARNAME == "BP_DIASTM23"]
    selected$BP_DIASTM23 <- format(round(rowMeans( selected[, c('BP_
```

```
DIAST2', 'BP_DIAST3')]), 2), nsmall=2, trim=T)
    attach(selected)
    selected$BP_DIASTM23[!eval(filters)] <- irr
    selected$BP_DIASTM23[eval(filters) & selected$BP_DIASTM23 == "NA"] <-
mis
    detach(selected)
    attach(selected)
    selected$BP_DIASTM12 <- format(round(rowMeans( selected[, c('BP_
DIAST1', 'BP_DIAST2')]), 2), nsmall=2, trim=T)
    selected$BP_DIASTM12[!eval(filters)] <- irr
    selected$BP_DIASTM12[eval(filters) & selected$BP_DIASTM12 == "NA"] <-
mis
    detach(selected)
    attach(selected)
    selected$BP_DIASTM123 <- format(round(rowMeans( selected[, c('BP_
DIAST1', 'BP_DIAST2', 'BP_DIAST3')]), 2), nsmall=2, trim=T)
    selected$BP_DIASTM123[!eval(filters)] <- irr
    selected$BP_DIASTM123[eval(filters) & selected$BP_DIASTM123 == "NA"]
<- mis
    detach(selected)
    #### BP_DIAST
    selected$BP_DIAST <- mean_variables_bp(selected, item=c('BP_DIAST'),
variables=variables )
    attach(selected)
    selected$BP_DIAST[!eval(filters)] <- irr
    selected$BP_DIAST[eval(filters) & selected$BP_DIAST == "NA"] <- mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'BP_DIASTM23', 'BP_DIASTM12', 'BP_DIASTM123', 'BP_
DIAST')], TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
      outdata <- selected
    }

    # ****************************************************************
********** #
    ############ HYPERTENSION ###############
    # ****************************************************************
********** #
    cat('Creating variables: MEDICINE_HBP1, BP_HBP, BP_HBP2, BP_HSBP,
BP_EBP, BP_CATEGORY, BP_MEDEFF_HBP, BP_AWARE, BP_AWARE2\n, MEDICINE_HBP_
POP, MEDICINE_HBP_ILL, BP_OPTIMAL', '\n', sep='')
    selected1 <- try(data[ , c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST' , 'MEDICINE', 'MEDICINE_HBP', 'DIAGN_HBP')] , TRUE)
    if (class( selected1 ) == "try-error") {
        stop(paste('Selected variables not found from dataset\n',
sep=""))
    }
```

```
    selected <- merge(selected1, outdata , all=T)
    ### filtering all derived variables as EXAM=1 or QUEST=1
    filters <-  expression(EXAM == '1' | QUEST == '1')
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
    irr <- variables$IRRELEVANT[variables$VARNAME == "BP_SYST"]
    mis = variables$MISSING[variables$VARNAME == "BP_SYST"]
    selected$BP_SYST = code_miss_numeric(selected, 'BP_SYST', mis, irr,
filter=filters)
    selected$BP_DIAST = code_miss_numeric(selected, 'BP_DIAST', mis,
irr, filter=filters)

     # *****************************************************************
*********** #
    ############ TREATMENT FOR HYPERTENSION ###############
    # ******************************************************************
********** #
    ###       Medicines for high blood pressure MEDICINE_HBP1
    mis = variables$MISSING[variables$VARNAME == "MEDICINE_HBP1"]
    attach(selected)
    selected$MEDICINE_HBP1[!eval(filters)] <- irr
    selected$MEDICINE_HBP1[eval(filters) &  MEDICINE != "1" ] <- "."
    selected$MEDICINE_HBP1[eval(filters) & MEDICINE == "1" & MEDICINE_HBP
== "1" ] <- "1"
    selected$MEDICINE_HBP1[eval(filters) & ((MEDICINE == "1" & MEDICINE_
HBP == "2") ) ] <- "2"
    selected$MEDICINE_HBP1[eval(filters) & (MEDICINE == "1" & MEDICINE_
HBP != "1"  & MEDICINE_HBP != "2")  ] <- mis
    detach(selected)
    #### Actual or potential hypertension BP_HBP
    mis = variables$MISSING[variables$VARNAME == "BP_HBP"]
    attach(selected)
    selected$BP_HBP[!eval(filters)] <- irr
    selected$BP_HBP[eval(filters) & ((BP_SYST >= 140 | BP_DIAST >= 90 |
MEDICINE_HBP1 == "1") & MEDICINE != "9" & MEDICINE_HBP1 != "9" & !is.
na(BP_SYST) & !is.na(BP_DIAST)) ] <- "1"
    selected$BP_HBP[eval(filters) &  (BP_SYST < 140 & BP_DIAST < 90 ) &
(MEDICINE_HBP1 != "1" & MEDICINE_HBP1 != "9")  ] <- "2"
    selected$BP_HBP[eval(filters) &  (is.na(BP_SYST) | is.na(BP_DIAST) )
| (MEDICINE_HBP1 == "9" |   MEDICINE == "9") ] <- mis
    detach(selected)
    #### Actual or potential hypertension BP_HBP2, higher limits
    mis = variables$MISSING[variables$VARNAME == "BP_HBP2"]
    attach(selected)
    selected$BP_HBP2[!eval(filters)] <- irr
    selected$BP_HBP2[eval(filters) & ((BP_SYST >= 160 | BP_DIAST >= 100 |
MEDICINE_HBP1 == "1") & !is.na(BP_SYST) & !is.na(BP_DIAST) & MEDICINE_
HBP1 != "9" & MEDICINE != "9") ] <- "1"
    selected$BP_HBP2[eval(filters) &  ((BP_SYST < 160 & BP_DIAST < 100 )
& MEDICINE_HBP1 != "1" & MEDICINE_HBP1 != "9") ] <- "2"
    selected$BP_HBP2[eval(filters) &  (is.na(BP_SYST) | is.na(BP_DIAST) )
| MEDICINE_HBP1 == "9" |   MEDICINE == "9"] <- mis
    detach(selected)
    #### Effective antihypertensive drug treatment BP_MEDEFF_HBP
```

```
    mis = variables$MISSING[variables$VARNAME == "BP_MEDEFF_HBP"]
    attach(selected)
    selected$BP_MEDEFF_HBP[!eval(filters)] <- irr
    selected$BP_MEDEFF_HBP[eval(filters) & MEDICINE_HBP1 != "1" ] <- "."
    selected$BP_MEDEFF_HBP[eval(filters) & BP_SYST < 140 & BP_DIAST < 90
& MEDICINE_HBP1 == "1" ] <- "1"
    selected$BP_MEDEFF_HBP[eval(filters) & ((BP_SYST >= 140 | BP_DIAST >=
90 ) & MEDICINE_HBP1 == "1") ] <- "2"
    selected$BP_MEDEFF_HBP[eval(filters) & (is.na(BP_SYST) | is.na(BP_
DIAST)) & MEDICINE_HBP1 == "1"] <- mis
    ###       Medicines for high blood pressure among population
MEDICINE_HBP_POP
    mis = variables$MISSING[variables$VARNAME == "MEDICINE_HBP_POP"]
    selected$MEDICINE_HBP_POP[!eval(filters)] <- irr
    selected$MEDICINE_HBP_POP[eval(filters) & MEDICINE_HBP1 == "1" ] <-
"1"
    selected$MEDICINE_HBP_POP[eval(filters) & (MEDICINE_HBP1 == "2" |
MEDICINE == "2") ] <- "2"
    selected$MEDICINE_HBP_POP[eval(filters) & ((MEDICINE == "1" &
MEDICINE_HBP1 == "9") | (MEDICINE != "1" & MEDICINE != "2" ) ) ] <- mis
    ###       Medicines for high blood pressure among actual or
potential   MEDICINE_HBP_ILL
    mis = variables$MISSING[variables$VARNAME == "MEDICINE_HBP_ILL"]
    selected$MEDICINE_HBP_ILL[!eval(filters)] <- irr
    selected$MEDICINE_HBP_ILL[eval(filters) & BP_HBP != "1" ] <- "."
    selected$MEDICINE_HBP_ILL[eval(filters) & BP_HBP == "1" & MEDICINE_
HBP1 == "1" ] <- "1"
    selected$MEDICINE_HBP_ILL[eval(filters) & ((BP_HBP == "1" & MEDICINE_
HBP1 == "2") | (BP_HBP == "1" & MEDICINE == "2")) ] <- "2"
    selected$MEDICINE_HBP_ILL[eval(filters) & (BP_HBP == "1" &  MEDICINE_
HBP1 == "9") | (BP_HBP == "1" &  MEDICINE == "9")  ] <- mis
    detach(selected)
    attach(selected)
    #### Systolic hypertension BP_HSBP
    mis = variables$MISSING[variables$VARNAME == "BP_HSBP"]
    selected$BP_HSBP[!eval(filters)] <- irr
    selected$BP_HSBP[eval(filters) & BP_SYST >= 140 ] <- "1"
    selected$BP_HSBP[eval(filters) & BP_SYST < 140  ] <- "2"
    selected$BP_HSBP[eval(filters) &  is.na(BP_SYST) ] <- mis
    #### Elevated blood pressure BP_EBP
    mis = variables$MISSING[variables$VARNAME == "BP_EBP"]
    selected$BP_EBP[!eval(filters)] <- irr
    selected$BP_EBP[eval(filters) &  (BP_SYST >= 140 | BP_DIAST >= 90)  ]
<- "1"
    selected$BP_EBP[eval(filters) & BP_SYST < 140 & BP_DIAST < 90  ] <-
"2"
    selected$BP_EBP[eval(filters) & (is.na(BP_SYST) | is.na(BP_DIAST)) ]
<- mis
    detach(selected)
    attach(selected)
     ###      Optimal blood pressure among those using anti-hypertensive
drugs   BP_OPTIMAL
    mis = variables$MISSING[variables$VARNAME == "BP_OPTIMAL"]
    selected$BP_OPTIMAL[!eval(filters)] <- irr
    selected$BP_OPTIMAL[eval(filters) & MEDICINE_HBP1 != "1" ] <- "."
```

```
    selected$BP_OPTIMAL[eval(filters) & MEDICINE_HBP1 == "1" & BP_EBP ==
"2" ] <- "1"
    selected$BP_OPTIMAL[eval(filters) & MEDICINE_HBP1 == "1" & BP_EBP ==
"1"  ] <- "2"
    selected$BP_OPTIMAL[eval(filters) & MEDICINE_HBP1 == "1" &   BP_EBP
!= "1"  &  BP_EBP != "2"  ] <- mis
    #### Categories of blood pressure BP_CATEGORY
    mis = variables$MISSING[variables$VARNAME == "BP_CATEGORY"]
    selected$BP_CATEGORY[!eval(filters)] <- irr
    selected$BP_CATEGORY[eval(filters) & BP_SYST < 120 & BP_DIAST < 80]
<- "1"
    selected$BP_CATEGORY[eval(filters) & ( (BP_SYST >=120 & BP_SYST < 130
&  BP_DIAST < 85 ) | (BP_DIAST >= 80 & BP_DIAST < 85 &  BP_DIAST < 130
)) ] <- "2"
    selected$BP_CATEGORY[eval(filters) & ( (BP_SYST >=130 & BP_SYST < 140
&  BP_DIAST < 90 ) | (BP_DIAST >= 85 & BP_DIAST < 90 &  BP_DIAST < 140
)) ] <- "3"
    selected$BP_CATEGORY[eval(filters) & ( (BP_SYST >=140 & BP_SYST < 160
&  BP_DIAST < 100 ) | (BP_DIAST >= 90 & BP_DIAST < 100 &  BP_DIAST < 160
)) ] <- "4"
    selected$BP_CATEGORY[eval(filters) & ( (BP_SYST >=160 & BP_SYST < 180
&  BP_DIAST < 110 ) | (BP_DIAST >= 100 & BP_DIAST < 110 &  BP_DIAST <
180 )) ] <- "5"
    selected$BP_CATEGORY[eval(filters) &  (BP_SYST >= 180 | BP_DIAST >=
110) ] <- "6"
    selected$BP_CATEGORY[eval(filters) & (is.na(BP_SYST) | is.na(BP_
DIAST))] <- mis
    detach(selected)
    ####     Awareness of hypertension BP_AWARE
    attach(selected)
    mis = variables$MISSING[variables$VARNAME == "BP_AWARE"]
    selected$BP_AWARE[!eval(filters)] <- irr
    selected$BP_AWARE[eval(filters) & BP_HBP != "1" ] <- "."
    selected$BP_AWARE[eval(filters) &  BP_HBP == "1" & DIAGN_HBP == "1" ]
<- "1"
    selected$BP_AWARE[eval(filters) &  BP_HBP == "1" & (DIAGN_HBP == "2"
| DIAGN_HBP == "8" )] <- "2"
    selected$BP_AWARE[eval(filters) & BP_HBP == "1" & DIAGN_HBP != "1" &
DIAGN_HBP != "2"  & DIAGN_HBP != "8"] <- mis
     ####    Awareness of hypertension BP_AWARE2, higher limits
    mis = variables$MISSING[variables$VARNAME == "BP_AWARE2"]
    selected$BP_AWARE2[!eval(filters)] <- irr
    selected$BP_AWARE2[eval(filters) & BP_HBP2 != "1" ] <- "."
    selected$BP_AWARE2[eval(filters) &  BP_HBP2 == "1" & DIAGN_HBP == "1"
] <- "1"
    selected$BP_AWARE2[eval(filters) &  BP_HBP2 == "1" & (DIAGN_HBP ==
"2" | DIAGN_HBP == "8" )] <- "2"
    selected$BP_AWARE2[eval(filters) & BP_HBP2 == "1" & DIAGN_HBP != "1"
& DIAGN_HBP != "2"  & DIAGN_HBP != "8"] <- mis
    detach(selected)
    selected <- try(selected[, c('PI', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'MEDICINE_HBP1', 'BP_HBP', 'BP_HBP2', 'BP_HSBP',
'BP_EBP', 'BP_CATEGORY', 'BP_MEDEFF_HBP', 'BP_AWARE', 'BP_AWARE2',
'MEDICINE_HBP_POP', 'MEDICINE_HBP_ILL', 'BP_OPTIMAL')], TRUE)
    if (class( selected ) == "try-error") {
```

```
        stop(paste('Selected variables not found from output dataset.',
'\n', sep=""))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
      outdata <- selected
    }
    selected <- NULL
    tstop <- Sys.time()
    cat('\nEHES_derived_variables_bp function call completed in
', format(round(difftime(tstop, tstart, units='mins'), digits=3),
nsmall=3), '.', '\n', sep='')
    return(outdata)
}
```

derived_variables_bs - to calculate derived variables for blood samples

```
derived_variables_bs <- function(data=NULL, variables) {
    ### data = dataset, variables = definition table for variables
    outdata=NULL
    tstart <- Sys.time()
    if (is.null(data) ) {
        stop("\n***ERROR:***\n No data given. Execution halted.")
    }
    cat('Creating variables:\nBS_CHOL_MMOL, BS_HDL_MMOL, BS_CHOLHDL_
RATIO, BS_NONHDL \nBS_HBA1C_MMOL, BS_FGLC_MMOL', '\n\n', sep='')
    selected <- try(data[,  c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SEX',
                      'BS_CHOL', 'BS_UNIT_CHOL', 'BS_HDL', 'BS_UNIT_
HDL', 'BS_HBA1C', 'BS_UNIT_HBA1C', 'BS_GLC', 'BS_UNIT_GLC', 'BS_FAST')]
, TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset \n',
sep=""))
    }
    ## to filter the participants
    filters <-  expression(EXAM == '1' | QUEST == '1')
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
    irr <- ""
    ## coding missing values and convert to numeric
    mis = variables$MISSING[variables$VARNAME == "BS_CHOL"]
    selected$BS_CHOL = code_miss_numeric(selected, 'BS_CHOL', mis, irr,
filter=filters )
     mis = variables$MISSING[variables$VARNAME == "BS_HDL"]
    selected$BS_HDL = code_miss_numeric(selected, 'BS_HDL', mis, irr,
filter=filters )
     mis = variables$MISSING[variables$VARNAME == "BS_HBA1C"]
    selected$BS_HBA1C = code_miss_numeric(selected, 'BS_HBA1C', irr,
mis, filter=filters )
     mis = variables$MISSING[variables$VARNAME == "BS_GLC"]
    selected$BS_GLC = code_miss_numeric(selected, 'BS_GLC', irr, mis,
```

```
filter=filters )

    # ***********************************************************************
********** #
    ############ Unit conversions to mmol/l: cholesterols, glucose,
HbA1c ##########
    # ***********************************************************************
********** #
    ### BS_CHOL_MMOL
    attach(selected)
    mis = variables$MISSING[variables$VARNAME == "BS_CHOL_MMOL"]
    selected$BS_CHOL_MMOL[!eval(filters)] <- ""
    selected$BS_CHOL_MMOL[eval(filters) & BS_UNIT_CHOL=='1'
]<-format(round( selected[eval(filters) & BS_UNIT_CHOL=='1',c('BS_
CHOL')], 4), nsmall=4, trim=T)
    selected$BS_CHOL_MMOL[eval(filters) & BS_UNIT_CHOL=='2'
]<-format(round( selected[eval(filters) & BS_UNIT_CHOL=='2',c('BS_
CHOL')]/38.67, 4), nsmall=4, trim=T)
    selected$BS_CHOL_MMOL[eval(filters) &  is.na(BS_CHOL) | (BS_UNIT_
CHOL!='1'  & BS_UNIT_CHOL!='2')  ] <- mis
    ### BS_HDL_MMOL
    mis = variables$MISSING[variables$VARNAME == "BS_HDL_MMOL"]
    selected$BS_HDL_MMOL[!eval(filters)] <- ""
    selected$BS_HDL_MMOL[eval(filters) & BS_UNIT_HDL=='1'
]<-format(round( selected[eval(filters) & BS_UNIT_HDL=='1',c('BS_HDL')] ,
4), nsmall=4, trim=T)
    selected$BS_HDL_MMOL[eval(filters) & BS_UNIT_HDL=='2'
]<-format(round( selected[eval(filters) & BS_UNIT_HDL=='2',c('BS_
HDL')]/38.67 , 4), nsmall=4, trim=T)
    selected$BS_HDL_MMOL[eval(filters) &  is.na(BS_HDL) | (BS_UNIT_
HDL!='1'  & BS_UNIT_HDL!='2') ] <- mis
    ### BS_HBA1C_MMOL
    mis = variables$MISSING[variables$VARNAME == "BS_HBA1C_MMOL"]
    selected$BS_HBA1C_MMOL[!eval(filters)] <- ""
    selected$BS_HBA1C_MMOL[eval(filters)  & BS_UNIT_HBA1C=='1' ]
<-selected[eval(filters)  & BS_UNIT_HBA1C=='1',c('BS_HBA1C')]
    selected$BS_HBA1C_MMOL[eval(filters)  & BS_UNIT_HBA1C=='2' ]
<- format(round((as.numeric(selected[eval(filters)  & BS_UNIT_
HBA1C=='2',c('BS_HBA1C')]) - 2.152) / 0.09148   , 4), nsmall=4, trim=T)
    selected$BS_HBA1C_MMOL[eval(filters) & is.na(BS_HBA1C) | (BS_UNIT_
HBA1C!='1'  & BS_UNIT_HBA1C!='2') ] <- mis
    ### BS_FGLC_MMOL
    mis = variables$MISSING[variables$VARNAME == "BS_FGLC_MMOL"]
    selected$BS_FGLC_MMOL[!eval(filters)] <- ""
    selected$BS_FGLC_MMOL[eval(filters) & BS_FAST != "1" ] <- "."
    selected$BS_FGLC_MMOL[eval(filters)  & BS_UNIT_GLC=='1' & BS_
FAST=='1' ] <-selected[eval(filters)  & BS_UNIT_GLC=='1' & BS_FAST=='1',
c('BS_GLC')]
    selected$BS_FGLC_MMOL[eval(filters)  & BS_UNIT_GLC=='2' & BS_
FAST=='1' ] <-format(round(as.numeric(selected[eval(filters)  & BS_UNIT_
GLC=='2' & BS_FAST=='1', c('BS_GLC')])*0.0555  , 4), nsmall=4, trim=T)
    selected$BS_FGLC_MMOL[eval(filters) & BS_FAST=='1'  & (is.na(BS_GLC)
| ( BS_UNIT_GLC!='1' & BS_UNIT_GLC!='2' ) )] <- mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
```

```
'EXAM', 'QUEST', 'SEX', 'BS_CHOL_MMOL', 'BS_HDL_MMOL', 'BS_HBA1C_MMOL',
'BS_FGLC_MMOL')], TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
    ## creating output data
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
      outdata <- selected
    }
    ### coding missing and converting to numeric
    mis = variables$MISSING[variables$VARNAME == "BS_CHOL_MMOL"]
    selected$BS_CHOL_MMOL = code_miss_numeric(selected, 'BS_CHOL_MMOL',
mis, irr, filter=filters )
     mis = variables$MISSING[variables$VARNAME == "BS_HDL_MMOL"]
    selected$BS_HDL_MMOL = code_miss_numeric(selected, 'BS_HDL_MMOL',
mis, irr, filter=filters )
     mis = variables$MISSING[variables$VARNAME == "BS_HBA1C_MMOL"]
     selected$BS_HBA1C_MMOL = code_miss_numeric(selected, 'BS_HBA1C_
MMOL', mis, irr, filter=filters )
    mis = variables$MISSING[variables$VARNAME == "BS_FGLC_MMOL"]
    irr = "."
    selected$BS_FGLC_MMOL = code_miss_numeric(selected, 'BS_FGLC_MMOL',
mis, irr, filter=filters )
    irr=""

    # ******************************************************************
********** #
    ############ Non-HDL and cholesterol to HDL ratio
    # ******************************************************************
********** #
    ### BS_NONHDL
    attach(selected)
    mis = variables$MISSING[variables$VARNAME == "BS_NONHDL"]
    selected$BS_FGLC_MMOL[!eval(filters)] <- ""
    selected$BS_NONHDL[eval(filters) & !is.na(BS_HDL_MMOL) & !is.na(BS_
CHOL_MMOL)] <- format(round( selected[eval(filters) & !is.na(BS_HDL_MMOL)
& !is.na(BS_CHOL_MMOL), c('BS_CHOL_MMOL')] - selected[eval(filters) &
!is.na(BS_HDL_MMOL) & !is.na(BS_CHOL_MMOL), c('BS_HDL_MMOL')] , 4),
nsmall=4, trim=T)
    selected$BS_NONHDL[eval(filters) & ( is.na(BS_HDL_MMOL) | is.na(BS_
CHOL_MMOL)) ] <- mis
    ### BS_CHOLHDL_RATIO
    mis = variables$MISSING[variables$VARNAME == "BS_CHOLHDL_RATIO"]
    selected$BS_CHOLHDL_RATIO[!eval(filters)] <- ""
    selected$BS_CHOLHDL_RATIO[eval(filters) & !is.na(BS_HDL_MMOL)
& !is.na(BS_CHOL_MMOL)]<- format(round( selected[ eval(filters)
&  !is.na(BS_HDL_MMOL) & !is.na(BS_CHOL_MMOL), c('BS_CHOL_MMOL')]/
selected[eval(filters) & !is.na(BS_HDL_MMOL) & !is.na(BS_CHOL_MMOL),
c('BS_HDL_MMOL')] , 4), nsmall=4, trim=T)
    selected$BS_CHOLHDL_RATIO[eval(filters) & (is.na(BS_HDL_MMOL)|is.
na(BS_CHOL_MMOL))]<-mis
```

```
    detach(selected)
    selected1 <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL', 
'EXAM', 'QUEST', 'SEX', 'BS_NONHDL', 'BS_CHOLHDL_RATIO')], TRUE)
    if (class( selected1 ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n', 
sep=""))
    }
    ### creating output data
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected1, all=T)
    }
    if (is.null(outdata)) {
      outdata <- selected1
    }
    selected1=NULL
    cat('Creating variables:\nBS_ECHOL, BS_ECHOL6, BS_ECHOL7, BS_ECHOL8, 
BS_LHDL, BS_HRATIO, BS_ENONHDL, BS_GLYCEMIA, BS_PROVDIAB_GLC, BS_
PROVDIAB_HBA1C, BS_DIAB_MSR, BS_DIAB_MSR2.', '\n\n', sep='')
    # coding missing values and converting to numeric
    mis = variables$MISSING[variables$VARNAME == "BS_CHOLHDL_RATIO"]
    selected$BS_CHOLHDL_RATIO = code_miss_numeric(selected, 'BS_CHOLHDL_
RATIO', mis, irr, filter=filters )
     mis = variables$MISSING[variables$VARNAME == "BS_NONHDL"]
    selected$BS_NONHDL = code_miss_numeric(selected, 'BS_NONHDL', mis, 
irr, filter=filters )

    # *********************************************************************
********** #
    ############ Elevated cholesterol, glucose, HbA1c levels 
###############
    # *********************************************************************
********** #
    ### BS_ECHOL
    attach(selected)
    mis = variables$MISSING[variables$VARNAME == "BS_ECHOL"]
    selected$BS_ECHOL[!eval(filters)] <- ""
    selected$BS_ECHOL[eval(filters) & !is.na(BS_CHOL_MMOL) & BS_CHOL_
MMOL>=5]<- '1'
    selected$BS_ECHOL[eval(filters) & !is.na(BS_CHOL_MMOL) & BS_CHOL_
MMOL<5]<- '2'
    selected$BS_ECHOL[eval(filters) & is.na(BS_CHOL_MMOL)]<- mis
    ### BS_ECHOL6
    mis = variables$MISSING[variables$VARNAME == "BS_ECHOL6"]
    selected$BS_ECHOL6[!eval(filters)] <- ""
    selected$BS_ECHOL6[eval(filters) & !is.na(BS_CHOL_MMOL) & BS_CHOL_
MMOL>=6]<- '1'
    selected$BS_ECHOL6[eval(filters) & !is.na(BS_CHOL_MMOL) & BS_CHOL_
MMOL<6]<- '2'
    selected$BS_ECHOL6[eval(filters) & is.na(BS_CHOL_MMOL)]<- mis
    ### BS_ECHOL7
    mis = variables$MISSING[variables$VARNAME == "BS_ECHOL7"]
    selected$BS_ECHOL7[!eval(filters)] <- ""
    selected$BS_ECHOL7[eval(filters) & !is.na(BS_CHOL_MMOL) & BS_CHOL_
MMOL>=7]<-'1'
    selected$BS_ECHOL7[eval(filters) & !is.na(BS_CHOL_MMOL) & BS_CHOL_
```

```
MMOL<7]<-'2'
    selected$BS_ECHOL7[eval(filters) & is.na(BS_CHOL_MMOL)]<-mis
    ### BS_ECHOL8
    mis = variables$MISSING[variables$VARNAME == "BS_ECHOL8"]
    selected$BS_ECHOL8[!eval(filters)] <- ""
    selected$BS_ECHOL8[eval(filters) & !is.na(BS_CHOL_MMOL) & BS_CHOL_
MMOL>=8]<-'1'
    selected$BS_ECHOL8[eval(filters) & !is.na(BS_CHOL_MMOL) & BS_CHOL_
MMOL<8]<-'2'
    selected$BS_ECHOL8[eval(filters) & is.na(BS_CHOL_MMOL)]<-  mis
    ### BS_LHDL
    mis = variables$MISSING[variables$VARNAME == "BS_LHDL"]
    selected$BS_LHDL[!eval(filters)] <- ""
    selected$BS_LHDL[eval(filters) & !is.na(BS_HDL_MMOL) & ((SEX=='1' &
BS_HDL_MMOL<1) | (SEX=='2' & BS_HDL_MMOL<1.2))] <- '1'
    selected$BS_LHDL[eval(filters) & !is.na(BS_HDL_MMOL) & ((SEX=='1' &
BS_HDL_MMOL>=1)| (SEX=='2' & BS_HDL_MMOL>=1.2))] <- '2'
    selected$BS_LHDL[eval(filters) &  (is.na(BS_HDL_MMOL) & SEX!='1' &
SEX != '2')] <- mis
    ## BS_HRATIO
    mis = variables$MISSING[variables$VARNAME == "BS_HRATIO"]
    selected$BS_HRATIO[!eval(filters)] <- ""
    selected$BS_HRATIO[eval(filters) &  !is.na(BS_CHOLHDL_RATIO) & BS_
CHOLHDL_RATIO > 5 ] <- '1'
    selected$BS_HRATIO[eval(filters) &  !is.na(BS_CHOLHDL_RATIO) & BS_
CHOLHDL_RATIO <= 5 ] <- '2'
    selected$BS_HRATIO[eval(filters) &  is.na(BS_CHOLHDL_RATIO)] <- mis
    ### BS_ENONHDL
    mis = variables$MISSING[variables$VARNAME == "BS_ENONHDL"]
    selected$BS_ENONHDL[!eval(filters)] <- ""
    selected$BS_ENONHDL[eval(filters) &  !is.na(BS_NONHDL) & BS_
NONHDL>3.5]<-'1'
    selected$BS_ENONHDL[eval(filters) & !is.na(BS_NONHDL) & BS_
NONHDL<=3.5]<-'2'
    selected$BS_ENONHDL[eval(filters) &  is.na(BS_NONHDL) ] <- mis
    ### BS_PROVDIAB_HBA1C
    mis = variables$MISSING[variables$VARNAME == "BS_PROVDIAB_HBA1C"]
    selected$BS_PROVDIAB_HBA1C[!eval(filters)] <- ""
    selected$BS_PROVDIAB_HBA1C[eval(filters)  & !is.na(BS_HBA1C_MMOL) &
BS_HBA1C_MMOL>=48]<- '1'
    selected$BS_PROVDIAB_HBA1C[eval(filters)  & !is.na(BS_HBA1C_MMOL) &
BS_HBA1C_MMOL<48]<- '2'
    selected$BS_PROVDIAB_HBA1C[eval(filters)  & is.na(BS_HBA1C_MMOL)]<-
mis
    ### BS_DIAB_MSR
    mis = variables$MISSING[variables$VARNAME == "BS_DIAB_MSR"]
    selected$BS_DIAB_MSR[!eval(filters)] <- ""
    selected$BS_DIAB_MSR[eval(filters)  & is.na(BS_HBA1C_MMOL) & !is.
na(BS_FGLC_MMOL) & BS_FGLC_MMOL>=7]<- '1'
    selected$BS_DIAB_MSR[eval(filters)  & !is.na(BS_HBA1C_MMOL) & BS_
HBA1C_MMOL>=48]<- '1'
    selected$BS_DIAB_MSR[eval(filters)  & is.na(BS_HBA1C_MMOL) & !is.
na(BS_FGLC_MMOL) & BS_FGLC_MMOL<7]<- '2'
    selected$BS_DIAB_MSR[eval(filters)  & !is.na(BS_HBA1C_MMOL) & BS_
HBA1C_MMOL<48]<- '2'
```

```
    selected$BS_DIAB_MSR[eval(filters) & is.na(BS_FGLC_MMOL) & is.na(BS_
HBA1C_MMOL)]<- mis
     ### BS_DIAB_MSR2, lower limits
    mis = variables$MISSING[variables$VARNAME == "BS_DIAB_MSR2"]
    selected$BS_DIAB_MSR2[!eval(filters)] <- ""
    selected$BS_DIAB_MSR2[eval(filters)  & is.na(BS_HBA1C_MMOL) & !is.
na(BS_FGLC_MMOL) & BS_FGLC_MMOL>=6]<- '1'
    selected$BS_DIAB_MSR2[eval(filters)  & !is.na(BS_HBA1C_MMOL) & BS_
HBA1C_MMOL>=42]<- '1'
    selected$BS_DIAB_MSR2[eval(filters)  & is.na(BS_HBA1C_MMOL) & !is.
na(BS_FGLC_MMOL) & BS_FGLC_MMOL<6]<- '2'
    selected$BS_DIAB_MSR2[eval(filters)  & !is.na(BS_HBA1C_MMOL) & BS_
HBA1C_MMOL<42]<- '2'
    selected$BS_DIAB_MSR2[eval(filters) & is.na(BS_FGLC_MMOL) & is.na(BS_
HBA1C_MMOL)]<- mis
    ### BS_GLYCEMIA
    mis = variables$MISSING[variables$VARNAME == "BS_GLYCEMIA"]
    selected$BS_GLYCEMIA[!eval(filters)] <- ""
    selected$BS_GLYCEMIA[eval(filters) & is.na(BS_FGLC_MMOL)] <- '9'
    selected$BS_GLYCEMIA[eval(filters) & !is.na(BS_FGLC_MMOL)  & BS_FGLC_
MMOL > 6 & BS_FGLC_MMOL < 7 ]<-'1'
    selected$BS_GLYCEMIA[eval(filters) & !is.na(BS_FGLC_MMOL)  & (BS_
FGLC_MMOL<=6 | BS_FGLC_MMOL>=7)] <- '2'
    selected$BS_GLYCEMIA[eval(filters) & is.na(BS_FGLC_MMOL)] <- mis
    ### BS_PROVDIAB_GLC
    mis = variables$MISSING[variables$VARNAME == "BS_PROVDIAB_GLC"]
    selected$BS_PROVDIAB_GLC[!eval(filters)] <- ""
    selected$BS_PROVDIAB_GLC[eval(filters)  & !is.na(BS_FGLC_MMOL) & BS_
FGLC_MMOL>=7] <- '1'
    selected$BS_PROVDIAB_GLC[eval(filters)  & !is.na(BS_FGLC_MMOL) & BS_
FGLC_MMOL<7] <- '2'
    selected$BS_PROVDIAB_GLC[eval(filters)  & is.na(BS_FGLC_MMOL)]<-mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SEX',
                                 'BS_ECHOL', 'BS_ECHOL6', 'BS_ECHOL7',
'BS_ECHOL8', 'BS_LHDL', 'BS_HRATIO', 'BS_ENONHDL',
                                 'BS_GLYCEMIA', 'BS_PROVDIAB_GLC', 'BS_
PROVDIAB_HBA1C' , 'BS_DIAB_MSR', 'BS_DIAB_MSR2')], TRUE)
    if (class( selected1 ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
    }
    selected = NULL

    # ******************************************************************
********** #
    ############ Treatment for lowering cholesterol levels, diabetes
###############
```

```
    # *******************************************************************
********** #
    cat('Creating variables:\nMEDICINE_CHOL1, MEDICINE_DIAB1, BS_HCHOL,
BS_HNONHDL, BS_DIAB_SR, BS_MEDEFF_CHOL, BS_MEDEFF_DIAB, BS_DIAB_SR, BS_
DIAB', '\n', sep='')
    selected1 <- try(data[ , c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SEX',
                      'MEDICINE', 'MEDICINE_CHOL', 'MEDICINE_DIAB',
'DIAGN_DIAB', 'DIAGN_CHOL')] , TRUE)
    if (class( selected1 ) == "try-error") {
        stop(paste('Selected variables not found from dataset\n',
sep=""))
    }
    selected <- merge(selected1, outdata , all=T)
    ### coding missing and converting to numeric
    mis = variables$MISSING[variables$VARNAME == "BS_CHOL_MMOL"]
    selected$BS_CHOL_MMOL = code_miss_numeric(selected, 'BS_CHOL_MMOL',
mis, irr, filter=filters )
    mis = variables$MISSING[variables$VARNAME == "BS_HDL_MMOL"]
    selected$BS_HDL_MMOL = code_miss_numeric(selected, 'BS_HDL_MMOL',
mis, irr, filter=filters )
    mis = variables$MISSING[variables$VARNAME == "BS_CHOLHDL_RATIO"]
    selected$BS_CHOLHDL_RATIO = code_miss_numeric(selected, 'BS_CHOLHDL_
RATIO', mis, irr, filter=filters )
    mis = variables$MISSING[variables$VARNAME == "BS_NONHDL"]
    selected$BS_NONHDL = code_miss_numeric(selected, 'BS_NONHDL', mis,
irr, filter=filters )
    mis = variables$MISSING[variables$VARNAME == "BS_HBA1C_MMOL"]
    selected$BS_HBA1C_MMOL = code_miss_numeric(selected, 'BS_HBA1C_
MMOL', mis, irr, filter=filters )
    mis = variables$MISSING[variables$VARNAME == "BS_FGLC_MMOL"]
    irr = "."
    selected$BS_FGLC_MMOL = code_miss_numeric(selected, 'BS_FGLC_MMOL',
mis, irr, filter=filters )
    irr = ""
    attach(selected)
    ### MEDICINE_CHOL1
    mis = variables$MISSING[variables$VARNAME == "MEDICINE_CHOL1"]
    selected$MEDICINE_CHOL1[!eval(filters)] <- irr
    selected$MEDICINE_CHOL1[eval(filters) &  MEDICINE != "1" ] <- "."
    selected$MEDICINE_CHOL1[eval(filters) & MEDICINE == "1" & MEDICINE_
CHOL == "1" ] <- "1"
    selected$MEDICINE_CHOL1[eval(filters) & ((MEDICINE == "1" & MEDICINE_
CHOL == "2") ) ] <- "2"
    selected$MEDICINE_CHOL1[eval(filters) & (MEDICINE == "1"  & MEDICINE_
CHOL != "1"  & MEDICINE_CHOL != "2")  ] <- mis
    ###MEDICINE_DIAB1
    mis = variables$MISSING[variables$VARNAME == "MEDICINE_DIAB1"]
    selected$MEDICINE_DIAB1[!eval(filters)] <- irr
    selected$MEDICINE_DIAB1[eval(filters) &  MEDICINE != "1" ] <- "."
    selected$MEDICINE_DIAB1[eval(filters) & MEDICINE == "1" & MEDICINE_
DIAB == "1" ] <- "1"
    selected$MEDICINE_DIAB1[eval(filters) & ((MEDICINE == "1" & MEDICINE_
DIAB == "2") ) ] <- "2"
    selected$MEDICINE_DIAB1[eval(filters) & (MEDICINE == "1"  & MEDICINE_
```

```
DIAB != ”1”  & MEDICINE_DIAB != ”2”)  ] <- mis
    detach(selected)
    ## BS_HCHOL
    mis = variables$MISSING[variables$VARNAME == ”BS_HCHOL”]
    attach(selected)
    selected$BS_HCHOL[!eval(filters)] <- irr
    selected$BS_HCHOL[eval(filters) & (BS_CHOL_MMOL>=5 | MEDICINE_
CHOL1==’1’)  & MEDICINE != ”9” & MEDICINE_CHOL1 != ”9” &  !is.na(BS_
CHOL_MMOL)  ] <- ”1”
    selected$BS_HCHOL[eval(filters) &   BS_CHOL_MMOL<5 & (MEDICINE_CHOL1
!= ”1” & MEDICINE_CHOL1 != ”9”)  ] <- ”2”
    selected$BS_HCHOL[eval(filters) &  is.na(BS_CHOL_MMOL) | MEDICINE_
CHOL1 == ”9” &   MEDICINE == ”9”] <- mis
    detach(selected)
    ## BS_MEDEFF_CHOL
    mis = variables$MISSING[variables$VARNAME == ”BS_MEDEFF_CHOL”]
    attach(selected)
    selected$BS_MEDEFF_CHOL[!eval(filters)] <- irr
    selected$BS_MEDEFF_CHOL[eval(filters) & MEDICINE_CHOL1 != ”1” ] <-
”.”
    selected$BS_MEDEFF_CHOL[eval(filters) & BS_CHOL_MMOL < 5 & MEDICINE_
CHOL1 == ”1” ] <- ”1”
    selected$BS_MEDEFF_CHOL[eval(filters) & BS_CHOL_MMOL >= 5 & MEDICINE_
CHOL1 == ”1” ] <- ”2”
    selected$BS_MEDEFF_CHOL[eval(filters) & is.na(BS_CHOL_MMOL) &
MEDICINE_CHOL1 == ”1”] <- mis
    ## BS_MEDEFF_DIAB
    mis = variables$MISSING[variables$VARNAME == ”BS_MEDEFF_DIAB”]
    selected$BS_MEDEFF_DIAB[!eval(filters)] <- irr
    selected$BS_MEDEFF_DIAB[eval(filters) & MEDICINE_DIAB1 != ”1” ] <-
”.”
    selected$BS_MEDEFF_DIAB[eval(filters) &  is.na(BS_HBA1C_MMOL) & !is.
na(BS_FGLC_MMOL) & BS_FGLC_MMOL<6 & MEDICINE_DIAB1 == ”1” ] <- ”1”
    selected$BS_MEDEFF_DIAB[eval(filters) &  !is.na(BS_HBA1C_MMOL) &  BS_
HBA1C_MMOL<48 & MEDICINE_DIAB1 == ”1” ] <- ”1”
    selected$BS_MEDEFF_DIAB[eval(filters) &  is.na(BS_HBA1C_MMOL) & !is.
na(BS_FGLC_MMOL) & BS_FGLC_MMOL>=6 & MEDICINE_DIAB1 == ”1” ] <- ”2”
    selected$BS_MEDEFF_DIAB[eval(filters) &  !is.na(BS_HBA1C_MMOL) &  BS_
HBA1C_MMOL>=48 & MEDICINE_DIAB1 == ”1” ] <- ”2”
    selected$BS_MEDEFF_DIAB[eval(filters) & is.na(BS_FGLC_MMOL)  &
is.na(BS_HBA1C_MMOL) & MEDICINE_DIAB1 == ”1”] <- mis
    ### BS_HNONHDL
    mis = variables$MISSING[variables$VARNAME == ”BS_HNONHDL”]
    selected$BS_HNONHDL[!eval(filters)] <- irr
    selected$BS_HNONHDL[eval(filters) &    (BS_NONHDL>3.5 | MEDICINE_
CHOL1==’1’)]<-’1’
    selected$BS_HNONHDL[eval(filters) & BS_NONHDL<=3.5 & MEDICINE_
CHOL1==’2’]<-’2’
    selected$BS_HNONHDL[eval(filters) & is.na(BS_NONHDL) & MEDICINE_CHOL1
!= ”1” &   MEDICINE_CHOL1 != ”2”] <-mis
    ### BS_DIAB_SR
    mis = variables$MISSING[variables$VARNAME == ”BS_DIAB_SR”]
    selected$BS_DIAB_SR[!eval(filters)] <- irr
    selected$BS_DIAB_SR[eval(filters) &  (BS_DIAB_MSR==’1’ | DIAGN_
DIAB==’1’)] <- ”1”
```

```
    selected$BS_DIAB_SR[eval(filters) &  BS_DIAB_MSR=='2' & DIAGN_
DIAB=='2'] <- "2"
    selected$BS_DIAB_SR[eval(filters) & BS_DIAB_MSR != "1" & BS_DIAB_MSR
!= "2" & DIAGN_DIAB != "1" & DIAGN_DIAB != "2"] <- mis
    ### BS_DIAB
    mis = variables$MISSING[variables$VARNAME == "BS_DIAB"]
    selected$BS_DIAB[!eval(filters)] <- irr
    selected$BS_DIAB[eval(filters) & (BS_DIAB_MSR=='1' | MEDICINE_
DIAB1=='1') ]<-'1'
    selected$BS_DIAB[eval(filters) & BS_DIAB_MSR=='2' & MEDICINE_
DIAB1=='2']<-'2'
    selected$BS_DIAB[eval(filters) & BS_DIAB_MSR != "1" & BS_DIAB_MSR !=
"2" & MEDICINE_DIAB1 != "1" & MEDICINE_DIAB1 != "2"] <- mis
    detach(selected)
    attach(selected)
    ###       Medicines among population   MEDICINE_DIAB_POP
    mis = variables$MISSING[variables$VARNAME == "MEDICINE_DIAB_POP"]
    selected$MEDICINE_DIAB_POP[!eval(filters)] <- irr
    selected$MEDICINE_DIAB_POP[eval(filters) & MEDICINE_DIAB1 == "1" ] <-
"1"
    selected$MEDICINE_DIAB_POP[eval(filters) & (MEDICINE_DIAB1 == "2" |
MEDICINE == "2") ] <- "2"
    selected$MEDICINE_DIAB_POP[eval(filters) & ((MEDICINE == "1" &
MEDICINE_DIAB1 == "9") | (MEDICINE != "1" & MEDICINE != "2" ) ) ] <- mis
    ###       Medicines  among actual or potential   MEDICINE_DIAB_ILL
    mis = variables$MISSING[variables$VARNAME == "MEDICINE_DIAB_ILL"]
    selected$MEDICINE_DIAB_ILL[!eval(filters)] <- irr
    selected$MEDICINE_DIAB_ILL[eval(filters) & BS_DIAB != "1" ] <- "."
    selected$MEDICINE_DIAB_ILL[eval(filters) & BS_DIAB == "1" & MEDICINE_
DIAB1 == "1" ] <- "1"
    selected$MEDICINE_DIAB_ILL[eval(filters) & ((BS_DIAB == "1" &
MEDICINE_DIAB1 == "2") | (BS_DIAB == "1" & MEDICINE == "2")) ] <- "2"
    selected$MEDICINE_DIAB_ILL[eval(filters) & (BS_DIAB == "1" &
MEDICINE_DIAB1 == "9")  ] <- mis
    selected$MEDICINE_DIAB_ILL[eval(filters) & (BS_DIAB == "1" &
MEDICINE_DIAB1 == "9" ) | (BS_DIAB == "1" & MEDICINE == "9") ] <- "."
    ###       Medicines among population   MEDICINE_CHOL_POP
    mis = variables$MISSING[variables$VARNAME == "MEDICINE_CHOL_POP"]
    selected$MEDICINE_CHOL_POP[!eval(filters)] <- irr
    selected$MEDICINE_CHOL_POP[eval(filters) & MEDICINE_CHOL1 == "1" ] <-
"1"
    selected$MEDICINE_CHOL_POP[eval(filters) & (MEDICINE_CHOL1 == "2" |
MEDICINE == "2") ] <- "2"
    selected$MEDICINE_CHOL_POP[eval(filters) & ((MEDICINE == "1" &
MEDICINE_CHOL1 == "9") | (MEDICINE != "1" & MEDICINE != "2" ) ) ] <- mis
    ###       Medicines  among actual or potential   MEDICINE_CHOL_ILL
    mis = variables$MISSING[variables$VARNAME == "MEDICINE_CHOL_ILL"]
    selected$MEDICINE_CHOL_ILL[!eval(filters)] <- irr
    selected$MEDICINE_CHOL_ILL[eval(filters) & BS_HCHOL != "1" ] <- "."
    selected$MEDICINE_CHOL_ILL[eval(filters) & BS_HCHOL == "1" &
MEDICINE_CHOL1 == "1" ] <- "1"
    selected$MEDICINE_CHOL_ILL[eval(filters) & ((BS_HCHOL == "1" &
MEDICINE_CHOL1 == "2") | (BS_HCHOL == "1" & MEDICINE == "2")) ] <- "2"
    selected$MEDICINE_CHOL_ILL[eval(filters) & (BS_HCHOL == "1" &
MEDICINE_CHOL1 == "9")  ] <- mis
```

```
    selected$MEDICINE_CHOL_ILL[eval(filters) & (BS_HCHOL == "1" &
MEDICINE_CHOL1 == "9" ) | (BS_HCHOL == "1" & MEDICINE == "9") ] <- "."
      ####    Awareness of elevated cholestero BS_CHOL_AWARE
    mis = variables$MISSING[variables$VARNAME == "BS_CHOL_AWARE"]
    selected$BS_CHOL_AWARE[!eval(filters)] <- irr
    selected$BS_CHOL_AWARE[eval(filters) & BS_HCHOL != "1" ] <- "."
    selected$BS_CHOL_AWARE[eval(filters) &  BS_HCHOL == "1" & DIAGN_CHOL
== "1" ] <- "1"
    selected$BS_CHOL_AWARE[eval(filters) &  BS_HCHOL == "1" & (DIAGN_CHOL
== "2" | DIAGN_CHOL == "8" )] <- "2"
    selected$BS_CHOL_AWARE[eval(filters) & BS_HCHOL == "1" & DIAGN_CHOL
!= "1" & DIAGN_CHOL != "2"  & DIAGN_CHOL != "8"] <- mis
    detach(selected)
    ####       BS_DIAB_AWARE
    attach(selected)
    mis = variables$MISSING[variables$VARNAME == "BS_DIAB_AWARE"]
    selected$BS_DIAB_AWARE[!eval(filters)] <- irr
    selected$BS_DIAB_AWARE[eval(filters) & BS_DIAB != "1" ] <- "."
    selected$BS_DIAB_AWARE[eval(filters) &  BS_DIAB == "1" & DIAGN_DIAB
== "1" ] <- "1"
    selected$BS_DIAB_AWARE[eval(filters) &  BS_DIAB == "1" & (DIAGN_DIAB
== "2" | DIAGN_DIAB == "8" )] <- "2"
    selected$BS_DIAB_AWARE[eval(filters) & BS_DIAB == "1" & DIAGN_DIAB !=
"1" & DIAGN_DIAB != "2"  & DIAGN_DIAB != "8"] <- mis
    ####       BS_DIAB_SR_AWARE
    mis = variables$MISSING[variables$VARNAME == "BS_DIAB_SR_AWARE"]
    selected$BS_DIAB_SR_AWARE[!eval(filters)] <- irr
    selected$BS_DIAB_SR_AWARE[eval(filters) & BS_DIAB_SR != "1" ] <- "."
    selected$BS_DIAB_SR_AWARE[eval(filters) &  BS_DIAB_SR == "1" & DIAGN_
DIAB == "1" ] <- "1"
    selected$BS_DIAB_SR_AWARE[eval(filters) &  BS_DIAB_SR == "1" &
(DIAGN_DIAB == "2" | DIAGN_DIAB == "8" )] <- "2"
    selected$BS_DIAB_SR_AWARE[eval(filters) & BS_DIAB_SR == "1" & DIAGN_
DIAB != "1" & DIAGN_DIAB != "2"  & DIAGN_DIAB != "8"] <- mis
    ###       Optimal cholestero among those using  drugs   BS_CHOL_
OPTIMAL
    mis = variables$MISSING[variables$VARNAME == "BS_CHOL_OPTIMAL"]
    selected$BS_CHOL_OPTIMAL[!eval(filters)] <- irr
    selected$BS_CHOL_OPTIMAL[eval(filters) & MEDICINE_CHOL1 != "1" ] <-
"."
    selected$BS_CHOL_OPTIMAL[eval(filters) & MEDICINE_CHOL1 == "1" & BS_
ECHOL == "2" ] <- "1"
    selected$BS_CHOL_OPTIMAL[eval(filters) & MEDICINE_CHOL1 == "1" & BS_
ECHOL == "1"  ] <- "2"
    selected$BS_CHOL_OPTIMAL[eval(filters) & MEDICINE_CHOL1 == "1" &
BS_ECHOL != "1"  &  BS_ECHOL != "2"  ] <- mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY',
'SERIAL','EXAM', 'QUEST', 'SEX',  'BS_HCHOL', 'BS_HNONHDL', 'MEDICINE_
CHOL1', 'BS_MEDEFF_CHOL', 'MEDICINE_DIAB1', 'BS_MEDEFF_DIAB',
                       'BS_DIAB_SR', 'BS_DIAB', 'BS_CHOL_AWARE', 'BS_
DIAB_AWARE', 'BS_DIAB_SR_AWARE', 'MEDICINE_DIAB_POP', 'MEDICINE_DIAB_
ILL', 'MEDICINE_CHOL_POP', 'MEDICINE_CHOL_ILL', 'BS_CHOL_OPTIMAL', 'BS_
GLC_OPTIMAL')] , TRUE)
    if (class( selected ) == "try-error") {
```

```
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
    }
    selected <- NULL
    tstop <- Sys.time()
    cat('\nEHES_derived_variables_bs function call completed in
', format(round(difftime(tstop, tstart, units='mins'), digits=3),
nsmall=3), '.', '\n', sep='')
    return(outdata)
}
```

derived_variables_anthropometrics - to calculate derived variables for anthropometrics

```
derived_variables_anthropometrics <- function(data=NA, variables) {
    ### data = dataset, variables = definition table for variables
    tstart <- Sys.time()
    outdata=NULL
    if (is.null(data) ) {
        stop("\n***ERROR:***\n No data given. Execution halted.")
    }
    cat('Creating variables:\nBMI, WAIST_LARGE, BMI_OVERWEIGHT, BMI_
OBESE, BMI_CATEGORY', '\n\n', sep='')
    selected <- try(data[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SEX',
                      'WEIGHT', 'HEIGHT', 'WAIST')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset \n',
sep=""))
    }
    # to filter the participants
    filters <-  expression(EXAM == '1' | QUEST == '1')
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
    irr = ""
    ## Coding missing values and converting to numeric
    mis = variables$MISSING[variables$VARNAME == "WEIGHT"]
    selected$WEIGHT = code_miss_numeric(selected, 'WEIGHT', mis, irr,
filter=filters )
    mis = variables$MISSING[variables$VARNAME == "HEIGHT"]
    selected$HEIGHT = code_miss_numeric(selected, 'HEIGHT', mis, irr,
filter=filters )
    mis = variables$MISSING[variables$VARNAME == "WAIST"]
    selected$WAIST = code_miss_numeric(selected, 'WAIST', mis, irr,
filter=filters )
    attach(selected)
    ### BMI
```

```
    mis = variables$MISSING[variables$VARNAME == "BMI"]
    selected$BMI[eval(filters) ] <- format(round(selected[eval(filters),
c('WEIGHT')] / (( selected[eval(filters) , c('HEIGHT')]/100)^2) , 2),
nsmall=2, trim=T)
    selected$BMI[!eval(filters)] <- irr
    selected$BMI[eval(filters) & (is.na(WEIGHT) | is.na(HEIGHT))] <- mis
    ## WAIST_LARGE
    mis = variables$MISSING[variables$VARNAME == "WAIST_LARGE"]
    selected$WAIST_LARGE[!eval(filters)] = irr
    selected$WAIST_LARGE[eval(filters) & SEX=='1' & WAIST>102]<- '1'
    selected$WAIST_LARGE[eval(filters)  & SEX=='2' & WAIST>88]<- '1'
    selected$WAIST_LARGE[eval(filters) &  SEX=='1' & WAIST<=102]<- '2'
    selected$WAIST_LARGE[eval(filters) &  SEX=='2' & WAIST<=88]<- '2'
    selected$WAIST_LARGE[eval(filters) & is.na(WAIST) & SEX != "1" & SEX
!= "2" ]<- mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SEX', 'BMI', 'WAIST_LARGE')], TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
    ## creating output data
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
      outdata <- selected
    }
    ## Coding missing values and converting to numeric
    mis = variables$MISSING[variables$VARNAME == "BMI"]
    selected$BMI = code_miss_numeric(selected, 'BMI', mis, irr,
filter=filters )
    attach(selected)
    ## BMI_OVERWEIGHT
    mis = variables$MISSING[variables$VARNAME == "BMI_OVERWEIGHT"]
    selected$BMI_OVERWEIGHT[!eval(filters)] <- irr
    selected$BMI_OVERWEIGHT[eval(filters) & BMI>=25]<- '1'
    selected$BMI_OVERWEIGHT[eval(filters) &   BMI<25]<- '2'
    selected$BMI_OVERWEIGHT[eval(filters) & is.na(BMI)]<- mis
    ## BMI_OBESE
    mis = variables$MISSING[variables$VARNAME == "BMI_OBESE"]
    selected$BMI_OBESE[!eval(filters)] <- mis
    selected$BMI_OBESE[eval(filters) &  BMI>=30]<- '1'
    selected$BMI_OBESE[eval(filters) &  BMI<30]<- '2'
    selected$BMI_OBESE[eval(filters) & is.na(BMI)]<- mis
    ## BMI_CATEGORY
     mis = variables$MISSING[variables$VARNAME == "BMI_CATEGORY"]
    selected$BMI_CATEGORY[!eval(filters)] <- irr
    selected$BMI_CATEGORY[eval(filters)  & BMI<18.5]<- '1'
    selected$BMI_CATEGORY[eval(filters)  & BMI>=18.5 & BMI<25]<- '2'
    selected$BMI_CATEGORY[eval(filters) & BMI>=25 & BMI<30]<- '3'
    selected$BMI_CATEGORY[eval(filters) & BMI>=30 & BMI<35]<- '4'
    selected$BMI_CATEGORY[eval(filters)  & BMI>=35 & BMI<40]<- '5'
    selected$BMI_CATEGORY[eval(filters)  & BMI>=40]<- '6'
```

```
    selected$BMI_CATEGORY[eval(filters) & is.na(BMI)]<- mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SEX',
                                 'BMI_OVERWEIGHT', 'BMI_OBESE', 'BMI_
CATEGORY')], TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
    }
    selected = NULL

    # ****************************************************************
********** #
    ############ self reported height, weight, BMI  ###############
    # ****************************************************************
********** #
    cat('Creating variables:\nBMI_SR, HEIGHT_DIFF, WEIGHT_DIFF, BMI_
DIFF, BMI_OVERWEIGHT_SR, BMI_OBESE_SR, BMI_CATEGORY_SR, PREGN1', '\n',
sep='')
    selected1 <- try(data[   , c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SEX',
                      'SHEIGHT', 'SWEIGHT', 'PREGN')] , TRUE)
    if (class( selected1 ) == "try-error") {
        stop(paste('Selected variables not found from dataset\n',
sep=""))
    }
    selected2 <- try(data[ , c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'ELIG', 'EXAM', 'QUEST', 'SEX',
                      'HEIGHT', 'WEIGHT')] , TRUE)
    if (class( selected2 ) == "try-error") {
        stop(paste('Selected variables not found from dataset\n',
sep=""))
    }
    selected <- merge(selected1, outdata , all=T)
    selected <- merge(selected, selected2 , all=T)
    selected1=NULL
    selected2=NULL
    ### coding missing and converting to numeric
    mis = variables$MISSING[variables$VARNAME == "WEIGHT"]
    selected$WEIGHT = code_miss_numeric(selected, 'WEIGHT', mis, irr,
filter=filters )
    mis = variables$MISSING[variables$VARNAME == "HEIGHT"]
    selected$HEIGHT = code_miss_numeric(selected, 'HEIGHT', mis, irr,
filter=filters )
    mis = variables$MISSING[variables$VARNAME == "BMI"]
    selected$BMI = code_miss_numeric(selected, 'BMI', mis, irr,
filter=filters )
    mis = c(variables$MISSING[variables$VARNAME == "SWEIGHT"], variables
```

```
$DONTKNOW[variables$VARNAME == "SWEIGHT"])
    selected$SWEIGHT = code_miss_numeric(selected, 'SWEIGHT', mis, irr, 
filter=filters )
    mis = c(variables$MISSING[variables$VARNAME == "SHEIGHT"], variables
$DONTKNOW[variables$VARNAME == "SHEIGHT"])
    selected$SHEIGHT = code_miss_numeric(selected, 'SHEIGHT', mis, irr, 
filter=filters )
    attach(selected)
    ### BMI_SR
    mis = variables$MISSING[variables$VARNAME == "BMI_SR"]
    selected$BMI_SR[eval(filters) ] <- format(round(selected[eval(filte
rs) , c('SWEIGHT')]/((selected[eval(filters), c('SHEIGHT')]/100)^2 ), 2), 
nsmall=2, trim=T)
    selected$BMI_SR[!eval(filters)] <- irr
    selected$BMI_SR[eval(filters) & (is.na(SWEIGHT)|is.na(SHEIGHT))]<- 
mis
    ### HEIGHT_DIFF
    mis = variables$MISSING[variables$VARNAME == "HEIGHT_DIFF"]
    selected$HEIGHT_DIFF[eval(filters) ]<- format(round(selected[eval(filt
ers) ,c('SHEIGHT')] - selected[eval(filters) ,c('HEIGHT')], 2), nsmall=2, 
trim=T)
    selected$HEIGHT_DIFF[!eval(filters)] <- irr
    selected$HEIGHT_DIFF[eval(filters) & (is.na(HEIGHT)|is.
na(SHEIGHT))]<- mis
    ### WEIGHT_DIFF
    mis = variables$MISSING[variables$VARNAME == "WEIGHT_DIFF"]
    selected$WEIGHT_DIFF[eval(filters)]<- format(round(selected[eval(filte
rs) ,c('SWEIGHT')] - selected[eval(filters) ,c('WEIGHT')] , 2), nsmall=2, 
trim=T)
    selected$WEIGHT_DIFF[!eval(filters)] <- irr
    selected$WEIGHT_DIFF[eval(filters) & (is.na(WEIGHT)|is.
na(SWEIGHT))]<- mis
     ### PREGN1
    mis = variables$MISSING[variables$VARNAME == "PREGN1"]
    selected$PREGN1[!eval(filters)] <- irr
    selected$PREGN1[eval(filters) & SEX != "2"]<- "."
    selected$PREGN1[eval(filters) & SEX == "2"]<- 
selected$PREGN[eval(filters) & SEX == "2"]
    selected$PREGN1[eval(filters) & SEX == "2" & !(PREGN %in% 
c("1","2"))] <- mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL', 
'EXAM', 'QUEST', 'SEX',
                                'BMI_SR', 'HEIGHT_DIFF', 'WEIGHT_DIFF', 
'PREGN1')], TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n', 
sep=""))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
    }
```

```
    selected = outdata
    ### coding missing and converting to numeric
    mis = variables$MISSING[variables$VARNAME == "BMI"]
    selected$BMI = code_miss_numeric(selected, 'BMI', mis, irr,
filter=filters )
    mis = variables$MISSING[variables$VARNAME == "BMI_SR"]
    selected$BMI_SR = code_miss_numeric(selected, 'BMI_SR', mis, irr,
filter=filters )
    attach(selected)
    ### BMI_DIFF
    mis = variables$MISSING[variables$VARNAME == "BMI_DIFF"]
    selected$BMI_DIFF[eval(filters) ]<-  format(round(selected[eval(fil
ters),c('BMI_SR')] - selected[eval(filters),c('BMI')] , 2), nsmall=2,
trim=T)
    selected$BMI_DIFF[!eval(filters)] <- irr
    selected$BMI_DIFF[eval(filters) &  (is.na(BMI)|is.na(BMI_SR))]<- mis
    ## BMI_OVERWEIGHT_SR
    mis = variables$MISSING[variables$VARNAME == "BMI_OVERWEIGHT_SR"]
    selected$BMI_OVERWEIGHT_SR[!eval(filters)] <- irr
    selected$BMI_OVERWEIGHT_SR[eval(filters) & BMI_SR>=25]<- '1'
    selected$BMI_OVERWEIGHT_SR[eval(filters) & BMI_SR<25]<- '2'
    selected$BMI_OVERWEIGHT_SR[eval(filters) & is.na(BMI_SR)]<- mis
    ## BMI_OBESE_SR
     mis = variables$MISSING[variables$VARNAME == "BMI_OBESE_SR"]
    selected$BMI_OBESE_SR[!eval(filters)] <- irr
    selected$BMI_OBESE_SR[eval(filters)  & BMI_SR>=30]<- '1'
    selected$BMI_OBESE_SR[eval(filters) & BMI_SR<30]<- '2'
    selected$BMI_OBESE_SR[eval(filters) & is.na(BMI_SR)]<- mis
    ## BMI_CATEGORY_SR
    mis = variables$MISSING[variables$VARNAME == "BMI_CATEGORY_SR"]
    selected$BMI_OBESE_SR[!eval(filters)] <- irr
    selected$BMI_CATEGORY_SR[eval(filters)  & BMI_SR<18.5]<- '1'
    selected$BMI_CATEGORY_SR[ eval(filters) & BMI_SR>=18.5 & BMI_SR<25]<-
'2'
    selected$BMI_CATEGORY_SR[eval(filters) & BMI_SR>=25 & BMI_SR<30]<-
'3'
    selected$BMI_CATEGORY_SR[ eval(filters) & BMI_SR>=30 & BMI_SR<35]<-
'4'
    selected$BMI_CATEGORY_SR[eval(filters) & BMI_SR>=35 & BMI_SR<40]<-
'5'
    selected$BMI_CATEGORY_SR[eval(filters)  & BMI_SR>=40]<- '6'
    selected$BMI_CATEGORY_SR[eval(filters) & is.na(BMI_SR)]<- mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SEX', 'BMI_DIFF', 'BMI_OVERWEIGHT_SR', 'BMI_OBESE_SR',
'BMI_CATEGORY_SR', 'PREGN1')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
```

```
    }
    selected <- NULL
    tstop <- Sys.time()
    cat('\nEHES_derived_variables_anthropometrics function call
completed in ', format(round(difftime(tstop, tstart, units='mins'),
digits=3), nsmall=3), '.', '\n', sep='')
    return(outdata)
}
```

derived_variables_smoking - to calculate derived variables for smoking

```
derived_variables_smoking <- function(data=NA, variables) {
    ### data = dataset, variables = definition table for variables
    tstart <- Sys.time()
    outdata=NULL
    if (is.null(data) ) {
        stop("\n***ERROR:***\n No data given. Execution halted.")
    }
    cat('Creating variables:\nSMOKEY, NUM_... and SMOKE_... variables',
'\n\n', sep='')
    selected <- try(data[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SMOKE', 'SMOKE_EVER', 'SMOKE_YEARS',
                              'SMOKE_CIGSM', 'NUM_CIGSM', 'SMOKE_CIGSH',
'NUM_CIGSH', 'SMOKE_CIGARS', 'NUM_CIGARS', 'SMOKE_PIPE', 'NUM_PIPE',
'SMOKE_OTHER', 'NUM_OTHER')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset \n',
sep=""))
    }
     # to filter the participants
    filters <-  expression(EXAM == '1' | QUEST == '1')
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
    irr = ""
    mis = variables$MISSING[variables$VARNAME == "SMOKEY"]
    attach(selected)
    selected$SMOKEY[!eval(filters)] <- irr
    selected$SMOKEY[eval(filters) & (SMOKE=='1'|SMOKE_EVER=='1')] <-
selected[eval(filters) &  (SMOKE=='1'|SMOKE_EVER=='1'), c('SMOKE_YEARS')]
    selected$SMOKEY[eval(filters) & (SMOKE_EVER=='2' & (SMOKE=='2' |
SMOKE=='3'))] <- '0'
    selected$SMOKEY[eval(filters) & SMOKE_YEARS =='99'  | (SMOKE != '1' &
SMOKE != '2' & SMOKE != '3' & SMOKE_EVER != '1' & SMOKE_EVER != '2') ]<-
mis
    ## SMOKE_... variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == "SMOKE_CIGSM1"]
    mis = variables$MISSING[variables$VARNAME == "SMOKE_CIGSM1"]
    selected$SMOKE_CIGSM1[!eval(filters)] <- ""
    selected$SMOKE_CIGSM1[eval(filters) & SMOKE != '1'] <- irr
    selected$SMOKE_CIGSM1[eval(filters) & SMOKE == '1'] <-
selected$SMOKE_CIGSM[eval(filters) & SMOKE == '1']
    selected$SMOKE_CIGSM1[eval(filters) & SMOKE == '1' & SMOKE_CIGSM !=
```

```
'1' & SMOKE_CIGSM != "2"]<-mis
    irr = variables$IRRELEVANT[variables$VARNAME == "SMOKE_CIGSH1"]
    mis = variables$MISSING[variables$VARNAME == "SMOKE_CIGSH1"]
    selected$SMOKE_CIGSH1[!eval(filters)] <- ""
    selected$SMOKE_CIGSH1[eval(filters) & SMOKE != '1'] <- irr
    selected$SMOKE_CIGSH1[eval(filters) & SMOKE == '1'] <-
selected$SMOKE_CIGSH[eval(filters) & SMOKE == '1']
    selected$SMOKE_CIGSH1[eval(filters) & SMOKE == '1' & SMOKE_CIGSH !=
'1' & SMOKE_CIGSH != "2"]<-mis
    irr = variables$IRRELEVANT[variables$VARNAME == "SMOKE_CIGARS1"]
    mis = variables$MISSING[variables$VARNAME == "SMOKE_CIGARS1"]
    selected$SMOKE_CIGARS1[!eval(filters)] <- ""
    selected$SMOKE_CIGARS1[eval(filters) & SMOKE != '1'] <- irr
    selected$SMOKE_CIGARS1[eval(filters) & SMOKE == '1'] <-
selected$SMOKE_CIGARS[eval(filters) & SMOKE == '1']
    selected$SMOKE_CIGARS1[eval(filters) & SMOKE == '1' & SMOKE_CIGARS !=
'1' & SMOKE_CIGARS != "2"]<-mis
    irr = variables$IRRELEVANT[variables$VARNAME == "SMOKE_PIPE1"]
    mis = variables$MISSING[variables$VARNAME == "SMOKE_PIPE1"]
    selected$SMOKE_PIPE1[!eval(filters)] <- ""
    selected$SMOKE_PIPE1[eval(filters) & SMOKE != '1'] <- irr
    selected$SMOKE_PIPE1[eval(filters) & SMOKE == '1'] <- selected$SMOKE_
PIPE[eval(filters) & SMOKE == '1']
    selected$SMOKE_PIPE1[eval(filters) & SMOKE == '1' & SMOKE_PIPE != '1'
& SMOKE_PIPE != "2"]<-mis
    irr = variables$IRRELEVANT[variables$VARNAME == "SMOKE_OTHER1"]
    mis = variables$MISSING[variables$VARNAME == "SMOKE_OTHER1"]
    selected$SMOKE_OTHER1[!eval(filters)] <- ""
    selected$SMOKE_OTHER1[eval(filters) & SMOKE != '1'] <- irr
    selected$SMOKE_OTHER1[eval(filters) & SMOKE == '1'] <-
selected$SMOKE_OTHER[eval(filters) & SMOKE == '1']
    selected$SMOKE_OTHER1[eval(filters) & SMOKE == '1' & SMOKE_OTHER !=
'1' & SMOKE_OTHER != "2"]<-mis
    detach(selected)
    ## NUM_... variables using filters
    attach(selected)
    irr = variables$IRRELEVANT[variables$VARNAME == "NUM_CIGSM1"]
    mis = variables$MISSING[variables$VARNAME == "NUM_CIGSM1"]
    selected$NUM_CIGSM1[!eval(filters)] <- ""
    selected$NUM_CIGSM1[eval(filters) & SMOKE_CIGSM1 != '1'] <- irr
    selected$NUM_CIGSM1[eval(filters) & SMOKE_CIGSM1 == '1'] <-
selected$NUM_CIGSM[eval(filters) & SMOKE_CIGSM1 == '1']
    selected$NUM_CIGSM1[eval(filters) & SMOKE_CIGSM1 == '1' & NUM_CIGSM
== mis]<-mis
    irr = variables$IRRELEVANT[variables$VARNAME == "NUM_CIGSH1"]
    mis = variables$MISSING[variables$VARNAME == "NUM_CIGSH1"]
    selected$NUM_CIGSH1[!eval(filters)] <- ""
    selected$NUM_CIGSH1[eval(filters) & SMOKE_CIGSH1 != '1'] <- irr
    selected$NUM_CIGSH1[eval(filters) & SMOKE_CIGSH1 == '1'] <-
selected$NUM_CIGSH[eval(filters) & SMOKE_CIGSH1 == '1']
    selected$NUM_CIGSH1[eval(filters) & SMOKE_CIGSH1 == '1' & NUM_CIGSH
== mis]<-mis
    irr = variables$IRRELEVANT[variables$VARNAME == "NUM_CIGARS1"]
    mis = variables$MISSING[variables$VARNAME == "NUM_CIGARS1"]
    selected$NUM_CIGARS1[!eval(filters)] <- ""
```

```
    selected$NUM_CIGARS1[eval(filters) & SMOKE_CIGARS1 != '1'] <- irr
    selected$NUM_CIGARS1[eval(filters) & SMOKE_CIGARS1 == '1'] <-
selected$NUM_CIGARS[eval(filters) & SMOKE_CIGARS1 == '1']
    selected$NUM_CIGARS1[eval(filters) & SMOKE_CIGARS1 == '1' & NUM_
CIGARS == mis]<-mis
     irr = variables$IRRELEVANT[variables$VARNAME == "NUM_PIPE1"]
    mis = variables$MISSING[variables$VARNAME == "NUM_PIPE1"]
    selected$NUM_PIPE1[!eval(filters)] <- ""
    selected$NUM_PIPE1[eval(filters) & SMOKE_PIPE1 != '1'] <- irr
    selected$NUM_PIPE1[eval(filters) & SMOKE_PIPE1 == '1'] <-
selected$NUM_PIPE[eval(filters) & SMOKE_PIPE1 == '1']
    selected$NUM_PIPE1[eval(filters) & SMOKE_PIPE1 == '1' & NUM_PIPE ==
mis]<-mis
     irr = variables$IRRELEVANT[variables$VARNAME == "NUM_OTHER1"]
    mis = variables$MISSING[variables$VARNAME == "NUM_OTHER1"]
    selected$NUM_OTHER1[!eval(filters)] <- ""
    selected$NUM_OTHER1[eval(filters) & SMOKE_OTHER1 != '1'] <- irr
    selected$NUM_OTHER1[eval(filters) & SMOKE_OTHER1 == '1'] <-
selected$NUM_OTHER[eval(filters) & SMOKE_OTHER1 == '1']
    selected$NUM_OTHER1[eval(filters) & SMOKE_OTHER1 == '1' & NUM_OTHER
== mis]<-mis
    ## SMOKE_EVER1
    irr = variables$IRRELEVANT[variables$VARNAME == "SMOKE_EVER1"]
    mis = variables$MISSING[variables$VARNAME == "SMOKE_EVER1"]
    selected$SMOKE_EVER1[!eval(filters)] <- ""
    selected$SMOKE_EVER1[eval(filters) & (SMOKE == '1' | SMOKE == '')] <-
irr
    selected$SMOKE_EVER1[eval(filters) &  SMOKE != '1' &  SMOKE != ''] <-
selected$SMOKE_EVER[eval(filters) &  SMOKE != '1' &  SMOKE != '']
    selected$SMOKE_EVER1[eval(filters) & SMOKE != '1' & SMOKE_EVER ==
mis] <-mis
    detach(selected)
    attach(selected)
    ### SMOKE_YEARS1
     irr = variables$IRRELEVANT[variables$VARNAME == "SMOKE_YEARS1"]
    mis = variables$MISSING[variables$VARNAME == "SMOKE_YEARS1"]
    selected$SMOKE_YEARS1[!eval(filters)] <- ""
    selected$SMOKE_YEARS1[eval(filters) & SMOKE != '1' & SMOKE_EVER1 !=
'1'] <- irr
    selected$SMOKE_YEARS1[eval(filters) &  (SMOKE == '1' | SMOKE_EVER1
== '1')] <- selected$SMOKE_YEARS[eval(filters) &  (SMOKE == '1' | SMOKE_
EVER1 == '1')]
    selected$SMOKE_YEARS1[eval(filters) & (SMOKE == '1' | SMOKE_EVER1 ==
'1') & SMOKE_YEARS == mis] <-mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SMOKEY', 'SMOKE_CIGSM1', 'NUM_CIGSM1', 'SMOKE_CIGSH1',
'NUM_CIGSH1',
                             'SMOKE_CIGARS1', 'NUM_CIGARS1', 'SMOKE_
PIPE1', 'NUM_PIPE1', 'SMOKE_OTHER1', 'NUM_OTHER1', 'SMOKE_EVER1',
'SMOKE_YEARS1')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
```

```
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
    }
    selected <- NULL
    tstop <- Sys.time()
    cat('EHES_derived_variables_smoking function call completed in
', format(round(difftime(tstop, tstart, units='mins'), digits=3),
nsmall=3), '.', '\n\n', sep='')
    return(outdata)
}
```

derived_variables_background - to calculate derived variables for background and self-perceived health items

```
derived_variables_background <- function(data=NA, variables) {
    ### data = dataset, variables = definition table for variables
    tstart <- Sys.time()
    outdata=NULL
    if (is.null(data) ) {
        stop("\n***ERROR:***\n No data given. Execution halted.")
    }
    cat('Creating variables:\nEMPLOYMENT1, EDUCATION_CAT, LABOUR_CAT,
HEALTH_GOOD, LIMITED_YES variables', '\n\n', sep='')
    selected <- try(data[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'PROFIT', 'EMPLOYMENT', 'EDUCATION', 'LABOUR_STATUS',
'HEALTH', 'LIMITED')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset \n',
sep=""))
    }
     # to filter the participants
    filters <-  expression(EXAM == '1' | QUEST == '1')
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
     attach(selected)
    ## EMPLOYMENT1 variable using filters
    irr = variables$IRRELEVANT[variables$VARNAME == "EMPLOYMENT1"]
    mis = variables$MISSING[variables$VARNAME == "EMPLOYMENT1"]
    selected$EMPLOYMENT1[!eval(filters)] <- ""
    selected$EMPLOYMENT1[eval(filters) & PROFIT != '1'] <- irr
    selected$EMPLOYMENT1[eval(filters) & PROFIT == '1'] <- selected$EMPLO
YMENT[eval(filters) & PROFIT == '1']
    selected$EMPLOYMENT1[eval(filters) & PROFIT == '1' & EMPLOYMENT !=
'1' & EMPLOYMENT != "2" & EMPLOYMENT != "3"]<-mis
    detach(selected)
      attach(selected)
    ## EDUCATION_CAT variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == "EDUCATION_CAT"]
    mis = variables$MISSING[variables$VARNAME == "EDUCATION_CAT"]
    selected$EDUCATION_CAT[!eval(filters)] <- ""
    selected$EDUCATION_CAT[eval(filters) & (EDUCATION == '1' | EDUCATION
```

```
== '2' | EDUCATION == '3' )] <- '1'
    selected$EDUCATION_CAT[eval(filters) & (EDUCATION == '4' | EDUCATION
== '5' )] <- '2'
    selected$EDUCATION_CAT[eval(filters) & (EDUCATION == '6' | EDUCATION
== '7' )] <- '3'
    selected$EDUCATION_CAT[eval(filters) & EDUCATION == '9']<-mis
    detach(selected)
    attach(selected)
    ## LABOUR_CAT variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == "LABOUR_CAT"]
    mis = variables$MISSING[variables$VARNAME == "LABOUR_CAT"]
    selected$LABOUR_CAT[!eval(filters)] <- ""
    selected$LABOUR_CAT[eval(filters) & (LABOUR_STATUS == '1' )] <- '1'
    selected$LABOUR_CAT[eval(filters) & (LABOUR_STATUS == '2'  )] <- '2'
    selected$LABOUR_CAT[eval(filters) & (LABOUR_STATUS == '3' | LABOUR_
STATUS == '4' | LABOUR_STATUS == '5' | LABOUR_STATUS == '6'| LABOUR_
STATUS == '7' | LABOUR_STATUS == '8'  )] <- '3'
    selected$LABOUR_CAT[eval(filters) & LABOUR_STATUS == '9']<-mis
    ## HEALTH_GOOD
    notknown = variables$DONTKNOW[variables$VARNAME == "HEALTH_GOOD"]
    mis = variables$MISSING[variables$VARNAME == "HEALTH_GOOD"]
    selected$HEALTH_GOOD[!eval(filters)] <- ""
    selected$HEALTH_GOOD[eval(filters) & (HEALTH == '1' | HEALTH == '2')]
<- '1'
    selected$HEALTH_GOOD[eval(filters) & (HEALTH != '1' & HEALTH != '2' &
HEALTH != notknown & HEALTH != mis)] <- '2'
    selected$HEALTH_GOOD[eval(filters) & (HEALTH == notknown)] <-
notknown
    selected$HEALTH_GOOD[eval(filters) & HEALTH == mis]<-mis
    ## LIMITED_YES
    notknown = variables$DONTKNOW[variables$VARNAME == "LIMITED_YES"]
    mis = variables$MISSING[variables$VARNAME == "LIMITED_YES"]
    selected$LIMITED_YES[!eval(filters)] <- ""
    selected$LIMITED_YES[eval(filters) & (LIMITED == '1' | LIMITED ==
'2')] <- '1'
    selected$LIMITED_YES[eval(filters) & (LIMITED != '1' & LIMITED != '2'
& LIMITED != notknown & LIMITED != mis)] <- '2'
    selected$LIMITED_YES[eval(filters) & (LIMITED == notknown)] <-
notknown
    selected$LIMITED_YES[eval(filters) & LIMITED == mis]<-mis
    detach(selected)
    selected <- try(selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'EMPLOYMENT1', 'EDUCATION_CAT', 'LABOUR_CAT', 'HEALTH_
GOOD', 'LIMITED_YES')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset.', '\n',
sep=""))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
    }
    selected <- NULL
```

```
    tstop <- Sys.time()
    cat('EHES_derived_variables_background function call completed
in ', format(round(difftime(tstop, tstart, units='mins'), digits=3),
nsmall=3), '.', '\n\n', sep='')
    return(outdata)
}
```

derived_variables_elig - to calculate derived variables for eligibility and consent items

```
derived_variables_elig <- function(data=NA, variables) {
    ### data = dataset, variables = definition table for variables
    tstart <- Sys.time()
    outdata=NULL
    if (is.null(data) ) {
        stop("\n***ERROR:***\n No data given. Execution halted.")
    }
    cat('Creating variables:\nEXAM1, QUEST1, REASON1, EXAM_REF1, EXAM_
NOAPP1, QUEST_REF1, CONSENT_...1,  variables', '\n\n', sep='')
    selected <- try(data[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'ELIG', 'EXAM', 'QUEST', 'REASON', 'EXAM_REF', 'EXAM_NOAPP', 'QUEST_
REF', 'CONSENT_ANTR', 'CONSENT_BP', 'CONSENT_BS', 'CONSENT_LIPID',
'CONSENT_GLC',  'CONSENT_STORE', 'CONSENT_DNA')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset \n',
sep=""))
    }
     # to filter the participants
    filters <-  TRUE
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
     attach(selected)
    ## EXAM1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == "EXAM1"]
    mis = variables$MISSING[variables$VARNAME == "EXAM1"]
    selected$EXAM1[!eval(filters)] <- ""
    selected$EXAM1[eval(filters) & ELIG != '1'] <- irr
    selected$EXAM1[eval(filters) & ELIG == '1'] <-
selected$EXAM[eval(filters) & ELIG == '1']
    ## REASON1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == "REASON1"]
    mis = variables$MISSING[variables$VARNAME == "REASON1"]
    selected$REASON1[!eval(filters)] <- ""
    selected$REASON1[eval(filters) & ELIG != '2'] <- irr
    selected$REASON1[eval(filters) & ELIG == '2'] <-
selected$REASON[eval(filters) & ELIG == '2']
    ## EXAM_REF1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == "EXAM_REF1"]
    mis = variables$MISSING[variables$VARNAME == "EXAM_REF1"]
    selected$EXAM_REF1[!eval(filters)] <- ""
    selected$EXAM_REF1[eval(filters) & ELIG != '1' | EXAM != "2"] <- irr
    selected$EXAM_REF1[eval(filters) & ELIG == '1' & EXAM == "2"] <-
selected$EXAM_REF[eval(filters) & ELIG == '1' & EXAM == "2"]
```

```
    ## EXAM_NOAPP1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ”EXAM_NOAPP1”]
    mis = variables$MISSING[variables$VARNAME == ”EXAM_NOAPP1”]
    selected$EXAM_NOAPP1[!eval(filters)] <- ””
    selected$EXAM_NOAPP1[eval(filters) & ELIG != ’1’ | EXAM != ”2” |
EXAM_REF == ”1”] <- irr
    selected$EXAM_NOAPP1[eval(filters) & ELIG == ’1’ & EXAM == ”2” &
EXAM_REF != ”1”] <- selected$EXAM_NOAPP[eval(filters) & ELIG == ’1’ &
EXAM == ”2” & EXAM_REF != ”1”]
    ## QUEST1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ”QUEST1”]
    mis = variables$MISSING[variables$VARNAME == ”QUEST1”]
    selected$QUEST1[!eval(filters)] <- ””
    selected$QUEST1[eval(filters) & ELIG != ’1’] <- irr
    selected$QUEST1[eval(filters) & ELIG == ’1’] <-
selected$QUEST[eval(filters) & ELIG == ’1’]
    ## QUEST_REF1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ”QUEST_REF1”]
    mis = variables$MISSING[variables$VARNAME == ”QUEST_REF1”]
    selected$QUEST_REF1[!eval(filters)] <- ””
    selected$QUEST_REF1[eval(filters) & ELIG != ’1’ | QUEST != ”2”] <-
irr
    selected$QUEST_REF1[eval(filters) & ELIG == ’1’ & QUEST == ”2”] <-
selected$QUEST_REF[eval(filters) & ELIG == ’1’ & QUEST == ”2”]
     ## CONSENT_ANTR1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ”CONSENT_ANTR1”]
    mis = variables$MISSING[variables$VARNAME == ”CONSENT_ANTR1”]
    selected$CONSENT_ANTR1[!eval(filters)] <- ””
    selected$CONSENT_ANTR1[eval(filters) & ELIG != ’1’ | EXAM != ”1”] <-
irr
    selected$CONSENT_ANTR1[eval(filters) & ELIG == ’1’ & EXAM == ”1”] <-
selected$CONSENT_ANTR[eval(filters) & ELIG == ’1’ & EXAM == ”1”]
     ## CONSENT_BP1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ”CONSENT_BP1”]
    mis = variables$MISSING[variables$VARNAME == ”CONSENT_BP1”]
    selected$CONSENT_BP1[!eval(filters)] <- ””
    selected$CONSENT_BP1[eval(filters) & ELIG != ’1’ | EXAM != ”1”] <-
irr
    selected$CONSENT_BP1[eval(filters) & ELIG == ’1’ & EXAM == ”1”] <-
selected$CONSENT_BP[eval(filters) & ELIG == ’1’ & EXAM == ”1”]
    ## CONSENT_BS1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ”CONSENT_BS1”]
    mis = variables$MISSING[variables$VARNAME == ”CONSENT_BS1”]
    selected$CONSENT_BS1[!eval(filters)] <- ””
    selected$CONSENT_BS1[eval(filters) & ELIG != ’1’ | EXAM != ”1”] <-
irr
    selected$CONSENT_BS1[eval(filters) & ELIG == ’1’ & EXAM == ”1”] <-
selected$CONSENT_BS[eval(filters) & ELIG == ’1’ & EXAM == ”1”]
    ## CONSENT_LIPID1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ”CONSENT_LIPID1”]
    mis = variables$MISSING[variables$VARNAME == ”CONSENT_LIPID1”]
    selected$CONSENT_LIPID1[!eval(filters)] <- ””
    selected$CONSENT_LIPID1[eval(filters) & ELIG != ’1’ | EXAM != ”1” |
CONSENT_BS != ”1”] <- irr
    selected$CONSENT_LIPID1[eval(filters) & ELIG == ’1’ & EXAM == ”1” &
```

```
CONSENT_BS == ″1″] <- selected$CONSENT_LIPID[eval(filters) & ELIG == ′1′
& EXAM == ″1″ & CONSENT_BS == ″1″]
    ## CONSENT_GLC1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ″CONSENT_GLC1″]
    mis = variables$MISSING[variables$VARNAME == ″CONSENT_GLC1″]
    selected$CONSENT_GLC1[!eval(filters)] <- ″″
    selected$CONSENT_GLC1[eval(filters) & ELIG != ′1′ | EXAM != ″1″ |
CONSENT_BS != ″1″] <- irr
    selected$CONSENT_GLC1[eval(filters) & ELIG == ′1′ & EXAM == ″1″ &
CONSENT_BS == ″1″] <- selected$CONSENT_GLC[eval(filters) & ELIG == ′1′ &
EXAM == ″1″ & CONSENT_BS == ″1″]
     ## CONSENT_STORE1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ″CONSENT_STORE1″]
    mis = variables$MISSING[variables$VARNAME == ″CONSENT_STORE1″]
    selected$CONSENT_STORE1[!eval(filters)] <- ″″
    selected$CONSENT_STORE1[eval(filters) & ELIG != ′1′ | EXAM != ″1″ |
CONSENT_BS != ″1″] <- irr
    selected$CONSENT_STORE1[eval(filters) & ELIG == ′1′ & EXAM == ″1″ &
CONSENT_BS == ″1″] <- selected$CONSENT_STORE[eval(filters) & ELIG == ′1′
& EXAM == ″1″ & CONSENT_BS == ″1″]
    ## CONSENT_DNA1 variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ″CONSENT_DNA1″]
    mis = variables$MISSING[variables$VARNAME == ″CONSENT_DNA1″]
    selected$CONSENT_DNA1[!eval(filters)] <- ″″
    selected$CONSENT_DNA1[eval(filters) & ELIG != ′1′ | EXAM != ″1″ |
CONSENT_BS != ″1″] <- irr
    selected$CONSENT_DNA1[eval(filters) & ELIG == ′1′ & EXAM == ″1″ &
CONSENT_BS == ″1″] <- selected$CONSENT_DNA[eval(filters) & ELIG == ′1′ &
EXAM == ″1″ & CONSENT_BS == ″1″]
    ## EXAMQUEST variables using filters
    irr = variables$IRRELEVANT[variables$VARNAME == ″EXAMQUEST″]
    mis = variables$MISSING[variables$VARNAME == ″EXAMQUEST″]
    selected$EXAMQUEST[!eval(filters)] <- ″″
    selected$EXAMQUEST[eval(filters) & ELIG != ′1′] <- irr
    selected$EXAMQUEST[eval(filters) & ELIG == ′1′ & EXAM == ″1″ & QUEST
== ″1″] <- ″1″
    selected$EXAMQUEST[eval(filters) & ELIG == ′1′ & (EXAM != ″1″ | QUEST
!= ″1″)] <- ″2″
    detach(selected)
    selected <- try(selected[, c(′PID′, ′COUNTRY′, ′SURVEY′, ′SERIAL′,
′ELIG′, ′EXAM1′, ′REASON1′, ′EXAM_REF1′, ′EXAM_NOAPP1′, ′QUEST1′,
′QUEST_REF1′,
                               ′CONSENT_ANTR1′,′CONSENT_BP1′,′CONSENT_
BS1′,′CONSENT_LIPID1′,′CONSENT_GLC1′,′CONSENT_DNA1′,′CONSENT_STORE1′,
′EXAMQUEST′)] , TRUE)
    if (class( selected ) == ″try-error″) {
        stop(paste(′Selected variables not found from dataset.′, ′\n′,
sep=″″))
    }
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
    }
```

```
    selected <- NULL
    tstop <- Sys.time()
    cat('EHES_derived_variables_elig function call completed in 
', format(round(difftime(tstop, tstart, units='mins'), digits=3), 
nsmall=3), '.', '\n\n', sep='')
    return(outdata)
}
```

derived_variables_schooling - to calculate third of years of education variable

```
derived_variables_schooling <- function(data=NA, variables) {
    ### data = dataset, variables = definition table for variables
    tstart <- Sys.time()
    outdata=NULL
    if (is.null(data) ) {
        stop("\n***ERROR:***\n No data given. Execution halted.")
    }
    cat('Creating variables:\nSCHOOL_THIRDS', '\n\n', sep='')
    selected <- try(data[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL', 
'EXAM', 'QUEST', 'BIRTH_YEAR', 'SCHOOL_YEARS', 'SEX')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset \n', 
sep=""))
    }
    filters <-  expression(EXAM == '1' | QUEST == '1')
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
    irr = ""
    mis = variables$MISSING[variables$VARNAME == "SCHOOL_YEARS"]
    selected$SCHOOL_YEARS = code_miss_numeric(selected, 'SCHOOL_YEARS', 
mis, irr, filter=filters )
    ### removing missing observations
    selected1 <- selected[, c("PID", "COUNTRY", "SURVEY",  "SERIAL", 
"EXAM", "QUEST")]
    selected  = selected[!is.na(selected$SCHOOL_YEARS) & !is.
na(selected$BIRTH_YEAR) ,]
    #Create key variables
    selected$GROUP<-paste(selected$COUNTRY, selected$SEX, sep="")
    selected$GROUPY<-paste(selected$GROUP, selected$BIRTH_YEAR, sep="")
    GROUPYS<-sort(unique(selected$GROUPY))
    #Create data frame for cutpoints
    tert1<-data.frame(SERIAL=GROUPYS,GROUP=substr(GROUPYS,1,3),COUNTRY=s
ubstr(GROUPYS,1,2),SEX=substr(GROUPYS,3,3),YEAR=substr(GROUPYS,4,7))
    tert1$YEAR<-as.numeric(levels(tert1$YEAR))[tert1$YEAR]
    #Add starting values
    tert2<-tert1
    tert2$YEAR<-tert1$YEAR-3
    tert2$SERIAL<-paste(tert2$GROUP,tert2$YEAR,sep="")
    tert<-merge(tert1,tert2,all=TRUE)
    #The function which calculate tertiles
    tertiles<-function(var){
        l<-quantile(var, probs=0.3333, type=1,na.rm=TRUE)
        u<-quantile(var, probs=0.6667, type=1,na.rm=TRUE)
```

```
        n<-length(var[!is.na(var)])
        c1<-sum(var<l,na.rm=TRUE)/n
        c2<-sum(var==l,na.rm=TRUE)/n
        c3<-sum(var>l & var<u,na.rm=TRUE)/n
        c4<-sum(var==u,na.rm=TRUE)/n
        c5<-sum(var>u,na.rm=TRUE)/n
        return(c(l,u,c1,c2,c3,c4,c5))
    }
    #Calculate tertiles
    for (i in 1:nrow(tert)) {
        group<-tert$GROUP[i]
        year<-tert$YEAR[i]
        x<-selected$SCHOOL_YEARS[selected$GROUP==group & abs(as.
numeric(selected$BIRTH_YEAR)-year)<4]
        y<-tertiles(as.numeric(x))
        y<-as.numeric(y)
        tert$L[i]<-y[1]
        tert$U[i]<-y[2]
        tert$C1[i]<-y[3]
        tert$C2[i]<-y[4]
        tert$C3[i]<-y[5]
        tert$C4[i]<-y[6]
        tert$C5[i]<-y[7]
    }
    #Define first year
    for (i in 1:nrow(tert)) {
        if (i==1) {
            tert$cond[i]<-1
        }
        else if (tert$GROUP[i]!=tert$GROUP[i-1]) {
            tert$cond[i]<-1
        }
        else {
            tert$cond[i]<-0
        }
    }
    #Calculate cutpoints
    for (i in 1:nrow(tert)){
        l<-tert$L[i]
        u<-tert$U[i]
        c1<-tert$C1[i]
        c2<-tert$C2[i]
        c3<-tert$C3[i]
        c4<-tert$C4[i]
        c5<-tert$C5[i]
        if (tert$cond[i]==1){
            absdiff1<-abs(1/3-c1)+abs(1/3-c2-c3-c4)+abs(1/3-c5)+10*(c1<0
.15)+10*(c5<0.15)+10*(c1<0.25)*(0.25-c1)+10*(c5<0.25)*(0.25-c5)
            absdiff<-absdiff1
            LOW<-l-0.5
            UP<-u+0.5
            absdiff2<-abs(1/3-c1-c2)+abs(1/3-c3-c4)+abs(1/3-c5
)+10*((c1+c2)<0.15)+10*(c5<0.15)+10*((c1+c2)<0.25)*(0.25-c1-
c2)+10*(c5<0.25)*(0.25-c5)+10*(u==l)
            if (absdiff2<absdiff) {
```

```
                absdiff<-absdiff2
                LOW<-l+0.5
                UP<-u+0.5
            }
            absdiff3<-abs(1/3-c1)+abs(1/3-c2-c3)+abs(1/3-c4-c5)+10*(c1<
0.15)+10*((c4+c5)<0.15)+10*(c1<0.25)*(0.25-c1)+10*((c4+c5)<0.25)*(0.25-
c4-c5)+10*(u==l)
            if (absdiff3<absdiff) {
                absdiff<-absdiff3
                LOW<-l-0.5
                UP<-u-0.5
            }
            absdiff4=abs(1/3-c1-c2)+abs(1/3-c3)+abs(1/3-c4-c5)+
10*((c1+c2)<0.15)+10*((c4+c5)<0.15)+10*((c1+c2)<0.25)*(0.25-c1-
c2)+10*((c4+c5)<0.25)*(0.25-c4-c5)+10*((u-l)==1)+100*(u==l);
            if (absdiff4<absdiff) {
                absdiff<-absdiff4
                LOW<-l+0.5
                UP<-u-0.5
            }
        }
        else {
            LOW1<-tert$LOW[i-1]
            UP1<-tert$UP[i-1]
            absdiff1<-abs(1/3-c1)+abs(1/3-c2-c3-c4)+abs(1/3-c5)+10*(c1<0
.15)+10*(c5<0.15)+10*(c1<0.25)*(0.25-c1)+10*(c5<0.25)*(0.25-c5)+0.2*((l-
0.5)!=LOW1)+0.2*((u+0.5)!=UP1)
            absdiff<-absdiff1
            LOW<-l-0.5
            UP<-u+0.5
            absdiff2<-abs(1/3-c1-c2)+abs(1/3-c3-c4)+abs(1/3-c5
)+10*((c1+c2)<0.15)+10*(c5<0.15)+10*((c1+c2)<0.25)*(0.25-c1-
c2)+10*(c5<0.25)*(0.25-c5)+10*(u==l)+0.2*((l+0.5)!=LOW1)+0.2*((u+0.5)!=
UP1)
            if (absdiff2<absdiff) {
                absdiff<-absdiff2
                LOW<-l+0.5
                UP<-u+0.5
            }
            absdiff3<-abs(1/3-c1)+abs(1/3-c2-c3)+abs(1/3-c4-c5)+10*(c1<
0.15)+10*((c4+c5)<0.15)+10*(c1<0.25)*(0.25-c1)+10*((c4+c5)<0.25)*(0.25-
c4-c5)+10*(u==l)+0.2*((l-0.5)!=LOW1)+0.2*((u-0.5)!=UP1)
            if (absdiff3<absdiff) {
                absdiff<-absdiff3
                LOW<-l-0.5
                UP<-u-0.5
            }
            absdiff4=abs(1/3-c1-c2)+abs(1/3-c3)+abs(1/3-c4-c5)+
10*((c1+c2)<0.15)+10*((c4+c5)<0.15)+10*((c1+c2)<0.25)*(0.25-c1-
c2)+10*((c4+c5)<0.25)*(0.25-c4-c5)+10*((u-l)==1)+100*(u==l)+0.2*((l+0.5)
!=LOW1)+0.2*((u-0.5)!=UP1);
            if (absdiff4<absdiff) {
                absdiff<-absdiff4
                LOW<-l+0.5
                UP<-u-0.5
```

```
            }
        }
                tert$LOW[i]<-LOW
                tert$UP[i]<-UP
    }
    #Drop starting values
    tert<-merge(tert,tert1,all.y=TRUE)
    selected<-merge(selected,tert[,c("SERIAL","LOW","UP")],by.
x="GROUPY",by.y="SERIAL")
    selected <- selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST', 'SCHOOL_YEARS', 'LOW', 'UP')]
    selected<-merge(selected,selected1, all=T)
    #Classify data by cutpoints
    mis = variables$MISSING[variables$VARNAME == "SCHOOL_THIRDS"]
    irr = variables$IRRELEVANT[variables$VARNAME == "SCHOOL_THIRDS"]
    attach(selected)
    selected$SCHOOL_THIRDS <- irr
    selected$SCHOOL_THIRDS[eval(filters) & !is.na(selected$SCHOOL_YEARS)&
selected$SCHOOL_YEARS<selected$LOW]<-'1'
    selected$SCHOOL_THIRDS[eval(filters) & !is.na(selected$SCHOOL_
YEARS)& selected$SCHOOL_YEARS<selected$UP & selected$SCHOOL_
YEARS>selected$LOW]<-'2'
    selected$SCHOOL_THIRDS[eval(filters) & !is.na(selected$SCHOOL_YEARS)
& selected$SCHOOL_YEARS>selected$UP]<-'3'
    selected$SCHOOL_THIRDS[eval(filters) & is.na(selected$SCHOOL_
YEARS)]<-mis
    detach(selected)
    selected <- selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST','SCHOOL_THIRDS')]
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
    }
    selected <- NULL
    tstop <- Sys.time()
    cat('EHES_derived_variables_schooling function call completed in
', format(round(difftime(tstop, tstart, units='mins'), digits=3),
nsmall=3), '.', '\n\n', sep='')
    return(outdata)
}
```

derived_variables_income - to calculate family income and household's equivalised disposable income

```
derived_variables_income <- function(data=NA, variables, dlimits) {
    ### data = dataset, variables = definition table for variables
    tstart <- Sys.time()
    outdata=NULL
    if (is.null(data) ) {
        stop("\n***ERROR:***\n No data given. Execution halted.")
    }
    cat('Creating variables: FINCOME, INCOME_CLASS.\n', '\n\n', sep='')
    selected <- try(data[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
```

```
'EXAM', 'QUEST', 'TOTAL_INCOME', 'PERSONS')] , TRUE)
    if (class( selected ) == "try-error") {
        stop(paste('Selected variables not found from dataset \n',
sep=""))
    }
    filters <-  expression(EXAM == '1' | QUEST == '1')
    attach(selected)
    selected = selected[eval(filters),]
    detach(selected)
    irr = ""
    mis1 = variables$MISSING[variables$VARNAME == "PERSONS"]
    mis2 = variables$MISSING[variables$VARNAME == "TOTAL_INCOME"]
    notknown2 = variables$DONTKNOW[variables$VARNAME == "TOTAL_INCOME"]
    ## converting to numeric
    selected$PERSONS = code_miss_numeric(selected, 'PERSONS', mis1, irr,
filter=filters )
    selected$KEY<-paste(selected$COUNTRY,selected$SURVEY,sep="")
    dlimits$KEY<-paste(dlimits$COUNTRY, dlimits$SURVEY,sep="")
    dlimits$L1<-as.numeric(dlimits$L1)
    dlimits$L2<-as.numeric(dlimits$L2)
    dlimits$L3<-as.numeric(dlimits$L3)
    dlimits$L4<-as.numeric(dlimits$L4)
    dlimits$L5<-as.numeric(dlimits$L5)
    dlimits$L6<-as.numeric(dlimits$L6)
    dlimits$L7<-as.numeric(dlimits$L7)
    dlimits$L8<-as.numeric(dlimits$L8)
    dlimits$L9<-as.numeric(dlimits$L9)
    dlimits$NCLASS<-as.numeric(dlimits$NCLASS)
    for (j in 1:nrow(dlimits))    {
        selected1<-selected[selected$KEY==dlimits$KEY[j],]
        selected1<-selected1[selected1$TOTAL_INCOME!=mis2 &
selected1$TOTAL_INCOME!=notknown2 & !is.na(selected1$TOTAL_INCOME)
& selected1$TOTAL_INCOME!=irr & selected1$PERSONS>0 &  !is.
na(selected1$PERSONS),]
        if (nrow(selected1)>0) {
            nclass<-as.numeric(dlimits$NCLASS[j])
            limits<-dlimits[j,4:(nclass+2)]
            limits<-as.numeric(limits)
            ######Define limits of income classes, weighting variable
            limits<-log(limits)
            selected1$CWEIGHT[1:nrow(selected1)]<-1
            selected1$CWEIGHT<-as.numeric(selected1$CWEIGHT)
            ######Calculate class frequencies
            n<-sum(as.numeric(selected1$CWEIGHT))
            frek<-c(1:nclass)
            for (i in 1:nclass) {
                frek[i]<-sum(selected1$CWEIGHT[selected1$TOTAL_
INCOME==i])/n
            }
            ######A log-linear approximation for lowest and highest
income classes
            p1<-frek[1]
            p10<-frek[nclass]
            z1<-qnorm(p1)
            z10<-qnorm(1-p10)
```

```
            x1<-as.numeric(limits[1])
            x10<-as.numeric(limits[length(limits)])
            sigma<-(x10-x1)/(z10-z1)
            mu<-x1-z1*sigma
            m1<-exp(mu+sigma*qnorm(p1/2))
            m10<-exp(mu+sigma*qnorm(1-0.5*p10))
            ######Calculate densities for limitspoin of lowest and
highest income classes
            dens<-c(1:nclass)
            dens[1]<-dnorm(x1,mean=mu,sd=sigma)
            dens[nclass]<-dnorm(x10,mean=mu,sd=sigma)
            ######Calculate densities for midpoints of other classes
            for (i in 2:(nclass-1)) {
                dens[i]<-frek[i]/(limits[i]-limits[i-1])
            }
            points<-c(1:nclass)
            points[1]<-limits[1]
            points[nclass]<-limits[nclass-1]
            for (i in 2:(nclass-1)) {
                points[i]<-0.5*(limits[i-1]+limits[i])
            }
            #####A piecewise linear approximation of the density
function
            alpha<-c(1:(nclass-2))
            beta<-c(1:(nclass-2))
            for (i in 1:(nclass-2)) {
                beta[i]<-(dens[i+2]-dens[i])/(points[i+2]-points[i])
                if (abs(beta[i])>dens[i+1]/(points[i+1]-limits[i])) {
                    beta[i]<-sign(dens[i+2]-dens[i])*dens[i+1]/
(points[i+1]-limits[i])
                }
                alpha[i]<-dens[i+1]-beta[i]*points[i+1]
            }
            ######Calculate median for each class
            mpoint<-c(1:nclass)
            mpoint[1]<-exp(mu+sigma*qnorm(p1/2))
            mpoint[nclass]<-exp(mu+sigma*qnorm(1-0.5*p10))
            for (i in 2:(nclass-1)) {
                a<-beta[i-1]/2
                b<-alpha[i-1]
                c<--0.5*beta[i-1]*limits[i-1]^2-alpha[i-1]*limits[i-1]-
0.5*frek[i]
                if (a==0) {
                    mpoint[i]<-exp(points[i])
                }
                else  {
                mpoint[i]<-exp((-b+sqrt(b^2-4*a*c))/(2*a))
                }
            }
            ######Calculate family income
            for (i in 1:nclass) {
                selected1$INCOME[selected1$TOTAL_INCOME==i]<-mpoint[i]
            }
            selected1$FINCOME<-selected1$INCOME/sqrt(as.
numeric(selected1$PERSONS))
```

```
            ######Calculate quantiles for the family income
            q<-quantile(selected1$FINCOME,probs=seq(0,1,0.2),na.rm=TRUE,
names=FALSE,type=1,weigth=selected1$CWEIGHT)
            ######Initial grouping of the family income
            q[1]<-0
            q[6]<-q[6]+1
            for (i in 1:5) {
                selected1$CLASS[selected1$FINCOME>q[i] &
selected1$FINCOME<q[i+1] & !is.na(selected1$FINCOME)]<-i
                ######Observations which equals a class limit
                selected1$CLASS[selected1$FINCOME==q[i+1]]<-i+0.5
            }
            ######Classify limit observations
            n<-sum(selected1$CWEIGHT[!is.na(selected1$CLASS)])
            selected1$CLASS<-as.numeric(selected1$CLASS)
            for (i in 1:4) {
                l<-0.2*i*n-sum(selected1$CWEIGHT[!is.na(selected1$CLASS)
& selected1$CLASS<=i])
                u<-sum(selected1$CWEIGHT[!is.na(selected1$CLASS) &
selected1$CLASS<i+1])-0.2*i*n
                selected1$CLASS[!is.na(selected1$CLASS) &
selected1$CLASS==i+0.5]<-selected1$CLASS[!is.na(selected1$CLASS) &
selected1$CLASS==i+0.5]+sign(u-l)*0.5
            }
            ######Classify remaining limit observations
            if (sum(!is.na(selected1$CLASS) & (selected1$CLASS-floor(sele
cted1$CLASS)==0.5))>0) {
                for (i in 1:4) {
                    if (sum(!is.na(selected1$CLASS) &
selected1$CLASS==i+0.5)>0) {
                        sum1<-sum(selected1$CWEIGHT[!is.
na(selected1$CLASS) & selected1$CLASS==i])
                        sum2<-sum(selected1$CWEIGHT[!is.
na(selected1$CLASS) & selected1$CLASS==i+1])
                        if (sum1<sum2) {
                            selected1$CLASS[selected1$CLASS==i+0.5]<-i
                        }
                        else if (sum1<sum2) {
                            selected1$CLASS[selected1$CLASS==i+0.5]<-i+1
                        }
                        else if (i==1|i==2) {
                            selected1$CLASS[selected1$CLASS==i+0.5]<-i
                        }
                        else {
                            selected1$CLASS[selected1$CLASS==i+0.5]<-i+1
                        }
                    }
                }
            }
            selected1$INCOME_CLASS<-selected1$CLASS
            selected1<-selected1[,c('KEY','SERIAL','FINCOME','INCOME_
CLASS')]
            if (j==1)   {
                incomes<-selected1
            }
```

```
            if (j>1)   {
                incomes<-merge(incomes,selected1,all=TRUE)
            }
        }
        selected1=NULL
    }
    selected<-merge(incomes,selected,all=TRUE)
    #Missing values
    mis1 = variables$MISSING[variables$VARNAME == "FINCOME"]
    mis2 = variables$MISSING[variables$VARNAME == "INCOME_CLASS"]
    attach(selected)
    ### rounding to integer
    selected$FINCOME[eval(filters)]<- format(round(selected[eval(filters)
,c('FINCOME')] , 0), nsmall=0, trim=T)
    selected$FINCOME[eval(filters) & selected$FINCOME == "NA"]<-mis1
    selected$INCOME_CLASS[eval(filters) & is.na(selected$INCOME_CLASS)]<-
mis2
    detach(selected)
    selected <- selected[, c('PID', 'COUNTRY', 'SURVEY', 'SERIAL',
'EXAM', 'QUEST','FINCOME', 'INCOME_CLASS')]
    if (!is.null(outdata)  ) {
        outdata <- merge(outdata, selected, all=T)
    }
    if (is.null(outdata)) {
        outdata <- selected
    }
    selected <- NULL
    tstop <- Sys.time()
    cat('EHES_derived_variables_income function call completed in
', format(round(difftime(tstop, tstart, units='mins'), digits=3),
nsmall=3), '.', '\n\n', sep='')
    return(outdata)
}
```

Estimation of indicators

drawmeans - to calculate means, standard errors and 95 % confidence intervals for continuous varibles by survey, sex and 10-year age-groups and age-standardisation for 25-64-year-olds by sex

```
drawmeans  <- function(data, varname, mis=NA, irr=NA, filters=TRUE) {
    ### data = dataset, varname = name of the variable, mis = code for
missing value, irr = code for irrelevants, filters = indicators for
participation
    data <- data[, c("COUNTRY", "SURVEY", "EXAM", "QUEST", "SEX", "AGE",
varname)]
    attach(data)
    data = data[eval(filters),]
    detach(data)
    countries <- unique(data$COUNTRY)
    ns_header = c("25-64_n_nonmissing", "men_25-64_n_nonmissing",
"women_25-64_n_nonmissing", "men_25-64_nmissing", "women_25-64_
nmissing",
                    "men_25-34_n", "men_35-44_n", "men_45-54_n", "men_55-
```

```
64_n”,
                    ”men_25-34_mean”, ”men_35-44_mean”, ”men_45-54_mean”,
”men_55-64_mean”, ”men_25-64_mean”,
                    ”men_25-34_se”, ”men_35-44_se”, ”men_45-54_se”,
”men_55-64_se”, ”men_25-64_se”,
                    ”women_25-34_n”, ”women_35-44_n”, ”women_45-54_n”,
”women_55-64_n”,
                    ”women_25-34_mean”, ”women_35-44_mean”, ”women_45-54_
mean”, ”women_55-64_mean”, ”women_25-64_mean”,
                     ”women_25-34_se”, ”women_35-44_se”, ”women_45-54_se”,
”women_55-64_se”, ”women_25-64_se” ,
                    ”men_25-34_cil”, ”men_35-44_cil”, ”men_45-54_cil”,
”men_55-64_cil”, ”men_25-64_cil”,      ”men_25-34_ciu”, ”men_35-44_ciu”,
”men_45-54_ciu”, ”men_55-64_ciu”, ”men_25-64_ciu”,
                      ”women_25-34_cil”, ”women_35-44_cil”, ”women_45-54_
cil”, ”women_55-64_cil”, ”women_25-64_cil”,      ”women_25-34_ciu”,
”women_35-44_ciu”, ”women_45-54_ciu”, ”women_55-64_ciu”, ”women_25-64_
ciu”)
    rows <- as.data.frame(cbind(””,””,”” , ””), stringsAsFactors  = F )
    names(rows) = c(”country”, ”survey”, ”ns_header”, ”ns”)
    firstrow=T
    for (country in countries) {
        surveys <- sort(unique(data[data$COUNTRY == country,]$SURVEY))
        for (survey in surveys) {
            selected <- data[data$COUNTRY == country & data$SURVEY ==
survey & data$AGE >= 25 & data$AGE < 65, ]
            selected <- selected[ !(selected[, varname] %in% c(irr) ) &
selected[, varname] != ” ” ,]
            nall1 <- nrow(selected)
            if (nall1 != 0) {
                nall1=NULL
                nall <- nrow(selected[!(selected[, varname] %in% c(mis)
), ])
                nmen <- nrow(selected[!(selected[, varname] %in% c(mis)
) & selected$SEX == ”1”, ])
                nwomen <- nrow(selected[!(selected[, varname] %in%
c(mis) ) & selected$SEX == ”2”, ])
                nmism <- nrow(selected[ selected[, varname] %in% mis &
selected$SEX == ”1”,])
                nmisw <- nrow(selected[ selected[, varname] %in% mis &
selected$SEX == ”2”,])
                if ( (nmism + nmisw) == nall) {
                    mean11 = ””; mean12 = ””; mean21 = ””; mean22 = ””;
mean31 = ””; mean32 = ””; mean41 = ””; mean42 = ””; mean1 = ””; mean2 =
””
                    n11 = ””; n12 = ””;  n21 = ””; n22 = ””; n31 = ””;
n32 = ””; n41 = ””; n42 = ””; n1 = ””; n2 = ””
                    se11 = ””; se12 = ””; se21 = ””; se22 = ””; se31 =
””; se32 = ””; se41 = ””; se42 = ””; se1 = ””; se2 = ””
                    cil11 = ””; cil12 = ””; cil21 = ””; cil22 = ””;
cil31 = ””; cil32 = ””; cil41 = ””; cil42 = ””; cil1 = ””; cil2 = ””
                    ciu11 = ””; ciu12 = ””; ciu21 = ””; ciu22 = ””;
ciu31 = ””; ciu32 = ””; ciu41 = ””; ciu42 = ””; ciu1 = ””; ciu2 = ””
                }
```

```
                else {
                    selected[ selected[, varname] %in% c(mis,irr) ,
varname] <- ""
                    selected[, varname] <- as.numeric(selected[,
varname])
                    age1 <-  selected[selected$AGE >= 25 & selected$AGE
< 35 , ]
                    age2 <-  selected[selected$AGE >= 35 & selected$AGE
< 45 , ]
                    age3 <-  selected[selected$AGE >= 45 & selected$AGE
< 55 , ]
                    age4 <-  selected[selected$AGE >= 55 & selected$AGE
< 65 , ]
                    if (nmen !=0 ) {
                        mean11 <-  mean(age1[age1$SEX == "1", varname],
na.rm=T)
                        mean21 <-  mean(age2[age2$SEX == "1", varname],
na.rm=T)
                        mean31 <-  mean(age3[age3$SEX == "1", varname],
na.rm=T)
                        mean41 <- mean(age4[age4$SEX == "1", varname],
na.rm=T)
                        n11 <- length(na.omit(age1[age1$SEX == "1",
varname]))
                        n21 <- length(na.omit(age2[age2$SEX == "1",
varname]))
                        n31 <- length(na.omit(age3[age3$SEX == "1",
varname]))
                        n41 <- length(na.omit(age4[age4$SEX == "1",
varname]))
                        se11 <- sqrt(var(age1[age1$SEX == "1", varname],
na.rm=T) / n11)
                        se21 <- sqrt(var(age2[age2$SEX == "1", varname],
na.rm=T) / n21)
                        se31 <- sqrt(var(age3[age3$SEX == "1", varname],
na.rm=T) / n31)
                        se41 <- sqrt(var(age4[age4$SEX == "1", varname],
na.rm=T) / n41)
                        se1 <- sqrt( sum( ((1/length(c(se11, se21, se31,
se41)) ) * c(se11, se21, se31, se41))^2 ))
                        mean1 <- mean(c(mean11, mean21, mean31, mean41))
                        cil11 <- mean11 - 1.96*se11
                        cil21 <- mean21 - 1.96*se21
                        cil31 <- mean31 - 1.96*se31
                        cil41 <- mean41 - 1.96*se41
                        cil1 <- mean1 - 1.96*se1
                        ciu11 <- mean11 + 1.96*se11
                        ciu21 <- mean21 + 1.96*se21
                        ciu31 <- mean31 + 1.96*se31
                        ciu41 <- mean41 + 1.96*se41
                        ciu1 <- mean1 + 1.96*se1
                        se1 =  format(round(se1, 3 ), nsmall=3)
                        mean1 = format(round(mean1, 2), nsmall=2)
                        mean11 =  format(round(mean11, 2), nsmall=2)
                        mean21 =  format(round(mean21, 2), nsmall=2)
```

```
            mean31 =  format(round(mean31, 2), nsmall=2)
            mean41 =  format(round(mean41, 2), nsmall=2)
            se11 =  format(round(se11, 3), nsmall=3)
            se21 =  format(round(se21, 3), nsmall=3)
            se31 =  format(round(se31, 3), nsmall=3)
            se41 =  format(round(se41, 3), nsmall=3)
            cil11 =  format(round(cil11, 1), nsmall=1)
            cil21 =  format(round(cil21, 1), nsmall=1)
            cil31 =  format(round(cil31, 1), nsmall=1)
            cil41 =  format(round(cil41, 1), nsmall=1)
            cil1 =  format(round(cil1, 1), nsmall=1)
            ciu11 =  format(round(ciu11, 1), nsmall=1)
            ciu21 =  format(round(ciu21, 1), nsmall=1)
            ciu31 =  format(round(ciu31, 1), nsmall=1)
            ciu41 =  format(round(ciu41, 1), nsmall=1)
            ciu1 =  format(round(ciu1, 1), nsmall=1)
        }
        else {
            mean11 = "" ; mean21 = ""; mean31 = ""; mean41 =
"" ; mean1 = ""

            n11 = ""; n21 = "";  n31 = ""; n41 = ""
            se11 = "" ; se21 = "" ; se31 = ""; se41 = "";

se1 = ""

            cil11 = ""; cil21 = ""; cil31 = ""; cil41 = "" ;

cil1 = ""

            ciu11 = ""; ciu21 = ""; ciu31 = ""; ciu41 = "";

ciu1 = ""

        }

        if (nwomen !=0 ) {
            mean12 <-   mean(age1[age1$SEX == "2", varname],
na.rm=T)

            mean22 <-   mean(age2[age2$SEX == "2", varname],
na.rm=T)

            mean32 <-   mean(age3[age3$SEX == "2", varname],
na.rm=T)

            mean42 <-   mean(age4[age4$SEX == "2", varname],
na.rm=T)

            n12 <- length(na.omit(age1[age1$SEX == "2",
varname]))

            n22 <-  length(na.omit(age2[age2$SEX == "2",
varname]))

            n32 <-  length(na.omit(age3[age3$SEX == "2",
varname]))

            n42 <-    length(na.omit(age4[age4$SEX == "2",
varname]))

            se12 <- sqrt(var(age1[age1$SEX == "2", varname],
na.rm=T) / n12)

            se22 <- sqrt(var(age2[age2$SEX == "2", varname],
na.rm=T) / n22)

            se32 <- sqrt(var(age3[age3$SEX == "2", varname],
na.rm=T) / n32)

            se42 <- sqrt(var(age4[age4$SEX == "2", varname],
na.rm=T) / n42)

            mean2 <- mean(c(mean12, mean22, mean32, mean42))
```

```
                        se2 <-  sqrt( sum( ((1/length(c(se12, se22,
se32, se42)) ) * c(se12, se22, se32, se42))^2 ))
                        cil12 <- mean12 - 1.96*se12
                        cil22 <- mean22 - 1.96*se22
                        cil32 <- mean32 - 1.96*se32
                        cil42 <- mean42 - 1.96*se42
                        cil2 <- mean2 - 1.96*se2
                        ciu12 <- mean12 + 1.96*se12
                        ciu22 <- mean22 + 1.96*se22
                        ciu32 <- mean32 + 1.96*se32
                        ciu42 <- mean42 + 1.96*se42
                        ciu2 <- mean2 + 1.96*se2
                        mean2 = format(round(mean2, 2), nsmall=2)
                        se2 <- format(round(se2 , 3), nsmall=3)
                        mean12 =  format(round(mean12, 2), nsmall=2)
                        mean22 =  format(round(mean22, 2), nsmall=2)
                        mean32 =  format(round(mean32, 2), nsmall=2)
                        mean42 =  format(round(mean42, 2), nsmall=2)
                        se12 =  format(round(se12, 3), nsmall=3)
                        se22 =  format(round(se22, 3), nsmall=3)
                        se32 =  format(round(se32, 3), nsmall=3)
                        se42 =  format(round(se42, 3), nsmall=3)
                        cil12 =  format(round(cil12, 1), nsmall=1)
                        cil22 =  format(round(cil22, 1), nsmall=1)
                        cil32 =  format(round(cil32, 1), nsmall=1)
                        cil42 =  format(round(cil42, 1), nsmall=1)
                        cil2 =  format(round(cil2, 1), nsmall=1)
                        ciu12 =  format(round(ciu12, 1), nsmall=1)
                        ciu22 =  format(round(ciu22, 1), nsmall=1)
                        ciu32 =  format(round(ciu32, 1), nsmall=1)
                        ciu42 =  format(round(ciu42, 1), nsmall=1)
                        ciu2 =  format(round(ciu2, 1), nsmall=1)
                    }
                    else {
                        mean12 = "" ; mean22 = ""; mean32 = ""; mean42 =
"" ; mean2 = ""
                        n12 = ""; n22 = ""; n32 = "" ; n42 = ""
                        se12 = ""; se22 = "" ; se32 = ""; se42 = "" ;
se2 = ""
                         cil12 = ""; cil22 = ""; cil32 = ""; cil42 = "";
cil2 = ""
                         ciu12 = ""; ciu22 = ""; ciu32 = ""; ciu42 = "";
ciu2 = ""
                    }
                }
            }
            else {
                nall = 0
                nmen = 0
                nwomen = 0
                nmism = ""
                nmisw = ""
                n11 = ""; n21 = ""; n31 = ""; n41 = ""; mean11 = "";
mean21 = ""; mean31 = ""; mean41 = ""; mean1 = ""; se11 = ""; se21 = "";
se31 = ""; se41 = ""; se1 = ""
```

```
                n12 = ""; n22 = ""; n32 = ""; n42 = ""; mean12 = "";
mean22 = ""; mean32 = ""; mean42 = ""; mean2 = ""; se12 = ""; se22 = "";
se32 = ""; se42 = ""; se2 = ""
                cil11 = ""; cil21 = ""; cil31 = ""; cil41 = ""; cil1 =
""; ciu11 = ""; ciu21 = ""; ciu31 = ""; ciu41 = ""; ciu1 = ""
                 cil12 = ""; cil22 = ""; cil32 = ""; cil42 = ""; cil2 =
""; ciu12 = ""; ciu22 = ""; ciu32 = ""; ciu42 = ""; ciu2 = ""
                nall1=NULL
            }
            ns = c(nall, nmen, nwomen, nmism, nmisw, n11, n21, n31, n41,
mean11, mean21, mean31, mean41, mean1, se11, se21, se31, se41, se1,
                n12, n22, n32, n42, mean12, mean22, mean32, mean42,
mean2, se12, se22, se32, se42, se2,
                cil11, cil21, cil31, cil41, cil1,  ciu11, ciu21, ciu31,
ciu41, ciu1,
                 cil12, cil22, cil32, cil42, cil2,  ciu12, ciu22, ciu32,
ciu42, ciu2)
            rows1 = cbind(country, survey, ns_header, ns)
            rows = rbind(rows, rows1)
            if (firstrow) {
                rows = rows[-c(1),]
                firstrow=F
            }
        }
    }
    ### TOTAL line
    country = "ALL"
    survey = "ALL"
    selected <- data[data$AGE >= 25 & data$AGE < 65, ]
    selected <- selected[ !(selected[, varname] %in% c(irr) ) &
selected[, varname] != " " ,]
    nall1 <- nrow(selected)
    if (nall1 != 0) {
        nall <- nrow(selected[!(selected[, varname] %in% c(mis) ), ])
        nmen <- nrow(selected[!(selected[, varname] %in% c(mis) ) &
selected$SEX == "1", ])
        nwomen <- nrow(selected[!(selected[, varname] %in% c(mis) ) &
selected$SEX == "2", ])
        nmism <- nrow(selected[ selected[, varname] %in% mis &
selected$SEX == "1",])
        nmisw <- nrow(selected[ selected[, varname] %in% mis &
selected$SEX == "2",])
        if (( nmism + nmisw) == nall) {
            mean11 = ""; mean12 = ""; mean21 = ""; mean22 = ""; mean31 =
""; mean32 = ""; mean41 = ""; mean42 = ""; mean1 = ""; mean2 = ""
                    n11 = ""; n12 = "";  n21 = ""; n22 = ""; n31 = "";
n32 = ""; n41 = ""; n42 = ""; n1 = ""; n2 = ""
                    se11 = ""; se12 = ""; se21 = ""; se22 = ""; se31 =
""; se32 = ""; se41 = ""; se42 = ""; se1 = ""; se2 = ""
                    cil11 = ""; cil12 = ""; cil21 = ""; cil22 = "";
cil31 = ""; cil32 = ""; cil41 = ""; cil42 = ""; cil1 = ""; cil2 = ""
                    ciu11 = ""; ciu12 = ""; ciu21 = ""; ciu22 = "";
ciu31 = ""; ciu32 = ""; ciu41 = ""; ciu42 = ""; ciu1 = ""; ciu2 = ""
        }
        else {
```

```
            selected[ selected[, varname] %in% c(mis,irr) , varname] <-
""
            selected[, varname] <- as.numeric(selected[, varname])
            age1 <-  selected[selected$AGE >= 25 & selected$AGE < 35 , ]
            age2 <-  selected[selected$AGE >= 35 & selected$AGE < 45 , ]
            age3 <-  selected[selected$AGE >= 45 & selected$AGE < 55 , ]
            age4 <-  selected[selected$AGE >= 55 & selected$AGE < 65 , ]
            if (nmen !=0 ) {
                mean11 <-  mean(age1[age1$SEX == "1", varname], na.rm=T)
                mean21 <-  mean(age2[age2$SEX == "1", varname], na.rm=T)
                mean31 <-  mean(age3[age3$SEX == "1", varname], na.rm=T)
                mean41 <- mean(age4[age4$SEX == "1", varname], na.rm=T)
                n11 <- length(na.omit(age1[age1$SEX == "1", varname]))
                n21 <- length(na.omit(age2[age2$SEX == "1", varname]))
                n31 <- length(na.omit(age3[age3$SEX == "1", varname]))
                n41 <- length(na.omit(age4[age4$SEX == "1", varname]))
                se11 <- sqrt(var(age1[age1$SEX == "1", varname],
na.rm=T) / n11)
                se21 <- sqrt(var(age2[age2$SEX == "1", varname],
na.rm=T) / n21)
                se31 <- sqrt(var(age3[age3$SEX == "1", varname],
na.rm=T) / n31)
                se41 <- sqrt(var(age4[age4$SEX == "1", varname],
na.rm=T) / n41)
                se1 <- sqrt( sum( ((1/length(c(se11, se21, se31, se41))
) * c(se11, se21, se31, se41))^2 ))
                mean1 <- mean(c(mean11, mean21, mean31, mean41))
                cil11 <- mean11 - 1.96*se11
                cil21 <- mean21 - 1.96*se21
                cil31 <- mean31 - 1.96*se31
                cil41 <- mean41 - 1.96*se41
                cil1 <- mean1 - 1.96*se1
                ciu11 <- mean11 + 1.96*se11
                ciu21 <- mean21 + 1.96*se21
                ciu31 <- mean31 + 1.96*se31
                ciu41 <- mean41 + 1.96*se41
                ciu1 <- mean1 + 1.96*se1
                se1 =  format(round(se1, 3 ), nsmall=3)
                mean1 = format(round(mean1, 2), nsmall=2)
                mean11 =  format(round(mean11, 2), nsmall=2)
                mean21 =  format(round(mean21, 2), nsmall=2)
                mean31 =  format(round(mean31, 2), nsmall=2)
                mean41 =  format(round(mean41, 2), nsmall=2)
                se11 =  format(round(se11, 3), nsmall=3)
                se21 =  format(round(se21, 3), nsmall=3)
                se31 =  format(round(se31, 3), nsmall=3)
                se41 =  format(round(se41, 3), nsmall=3)
               cil11 =  format(round(cil11, 1), nsmall=1)
                cil21 =  format(round(cil21, 1), nsmall=1)
                cil31 =  format(round(cil31, 1), nsmall=1)
                cil41 =  format(round(cil41, 1), nsmall=1)
                cil1 =  format(round(cil1, 1), nsmall=1)
                ciu11 =  format(round(ciu11, 1), nsmall=1)
                ciu21 =  format(round(ciu21, 1), nsmall=1)
                ciu31 =  format(round(ciu31, 1), nsmall=1)
```

```
            ciu41 =  format(round(ciu41, 1), nsmall=1)
            ciu1 =  format(round(ciu1, 1), nsmall=1)
        }
        else {
            mean11 = "" ; mean21 = ""; mean31 = ""; mean41 = "" ;
mean1 = ""
            n11 = ""; n21 = "";  n31 = ""; n41 = ""
            se11 = "" ; se21 = "" ; se31 = ""; se41 = ""; se1 = ""
            cil11 = ""; cil21 = ""; cil31 = ""; cil41 = "" ; cil1 =
""
            ciu11 = ""; ciu21 = ""; ciu31 = ""; ciu41 = ""; ciu1 =
""
        }
        if (nwomen !=0 ) {
            mean12 <-   mean(age1[age1$SEX == "2", varname],
na.rm=T)
            mean22 <-   mean(age2[age2$SEX == "2", varname],
na.rm=T)
            mean32 <-   mean(age3[age3$SEX == "2", varname],
na.rm=T)
            mean42 <-   mean(age4[age4$SEX == "2", varname],
na.rm=T)
            n12 <- length(na.omit(age1[age1$SEX == "2", varname]))
            n22 <-  length(na.omit(age2[age2$SEX == "2", varname]))
            n32 <-  length(na.omit(age3[age3$SEX == "2", varname]))
            n42 <-    length(na.omit(age4[age4$SEX == "2",
varname]))
            se12 <- sqrt(var(age1[age1$SEX == "2", varname],
na.rm=T) / n12)
            se22 <- sqrt(var(age2[age2$SEX == "2", varname],
na.rm=T) / n22)
            se32 <- sqrt(var(age3[age3$SEX == "2", varname],
na.rm=T) / n32)
            se42 <- sqrt(var(age4[age4$SEX == "2", varname],
na.rm=T) / n42)
            mean2 <- mean(c(mean12, mean22, mean32, mean42))
            se2 <-  sqrt( sum( ((1/length(c(se12, se22, se32, se42))
) * c(se12, se22, se32, se42))^2 ))
            cil12 <- mean12 - 1.96*se12
            cil22 <- mean22 - 1.96*se22
            cil32 <- mean32 - 1.96*se32
            cil42 <- mean42 - 1.96*se42
            cil2 <- mean2 - 1.96*se2
            ciu12 <- mean12 + 1.96*se12
            ciu22 <- mean22 + 1.96*se22
            ciu32 <- mean32 + 1.96*se32
            ciu42 <- mean42 + 1.96*se42
            ciu2 <- mean2 + 1.96*se2
            mean2 = format(round(mean2, 2), nsmall=2)
            se2 <- format(round(se2 , 3), nsmall=3)
            mean12 =  format(round(mean12, 2), nsmall=2)
            mean22 =  format(round(mean22, 2), nsmall=2)
            mean32 =  format(round(mean32, 2), nsmall=2)
            mean42 =  format(round(mean42, 2), nsmall=2)
            se12 =  format(round(se12, 3), nsmall=3)
```

```
                se22 =  format(round(se22, 3), nsmall=3)
                se32 =  format(round(se32, 3), nsmall=3)
                se42 =  format(round(se42, 3), nsmall=3)
                cil12 =  format(round(cil12, 1), nsmall=1)
                cil22 =  format(round(cil22, 1), nsmall=1)
                cil32 =  format(round(cil32, 1), nsmall=1)
                cil42 =  format(round(cil42, 1), nsmall=1)
                cil2 =  format(round(cil2, 1), nsmall=1)
                ciu12 =  format(round(ciu12, 1), nsmall=1)
                ciu22 =  format(round(ciu22, 1), nsmall=1)
                ciu32 =  format(round(ciu32, 1), nsmall=1)
                ciu42 =  format(round(ciu42, 1), nsmall=1)
                ciu2 =  format(round(ciu2, 1), nsmall=1)
            }
            else {
                mean12 = "" ; mean22 = ""; mean32 = ""; mean42 = "" ;
mean2 = ""
                n12 = ""; n22 = ""; n32 = "" ; n42 = ""
                se12 = ""; se22 = "" ; se32 = ""; se42 = "" ; se2 = ""
                cil12 = ""; cil22 = ""; cil32 = ""; cil42 = ""; cil2 =
""
                ciu12 = ""; ciu22 = ""; ciu32 = ""; ciu42 = ""; ciu2 =
""
            }
        }
    }
    else {
        nall = 0
        nmen = 0
        nwomen = 0
        nmism = ""
        nmisw = ""
        n11 = ""; n21 = ""; n31 = ""; n41 = ""; mean11 = ""; mean21 =
""; mean31 = ""; mean41 = ""; mean1 = ""; se11 = ""; se21 = ""; se31 =
""; se41 = ""; se1 = ""
        n12 = ""; n22 = ""; n32 = ""; n42 = ""; mean12 = ""; mean22 =
""; mean32 = ""; mean42 = ""; mean2 = ""; se12 = ""; se22 = ""; se32 =
""; se42 = ""; se2 = ""
        cil21 = ""; cil31 = ""; cil41 = ""; cil1 = "";  ciu21 = "";
ciu31 = ""; ciu41 = ""; ciu1 = ""
         cil12 = ""; cil22 = ""; cil32 = ""; cil42 = ""; cil2 = "" ;
ciu12 = ""; ciu22 = ""; ciu32 = ""; ciu42 = ""; ciu2 = ""
    }
    ns = c(nall, nmen, nwomen, nmism, nmisw, n11, n21, n31, n41, mean11,
mean21, mean31, mean41, mean1, se11, se21, se31, se41, se1,
                n12, n22, n32, n42, mean12, mean22, mean32, mean42,
mean2, se12, se22, se32, se42, se2,
                cil11, cil21, cil31, cil41, cil1,  ciu11, ciu21, ciu31,
ciu41, ciu1,
                 cil12, cil22, cil32, cil42, cil2,  ciu12, ciu22, ciu32,
ciu42, ciu2)
    rows1 = cbind(country, survey, ns_header, ns)
    rows = rbind(rows, rows1)
    cat('Means for ', varname, ' done.\n', sep='')
    return(rows)
```

```
}
```

drawprops - to calculate proportions and number of observations for categorical varibles by survey, sex and 10-year age-groups and age-standardisation for 25-64-year-olds by sex

```
drawprops  <- function(data, varname, mis=NA, irr=NA, filters=TRUE) {
    ### data = dataset, varname= name of the variable, mis = code for
missing value, irr = code for irrelevants, filters = indicators for
participation
    if (varname == "ELIG" | varname == "EXAM1" | varname == "QUEST1" |
varname == "EXAMQUEST") {
        data <- data[, c("COUNTRY", "SURVEY",  "SEX", "AGE",   varname)]
    }
    else {
         data <- data[, c("COUNTRY", "SURVEY", "EXAM", "QUEST",  "SEX",
"AGE", varname)]
    }
    attach(data)
    data = data[eval(filters),]
    detach(data)
    values <- sort(unique(data[!(data[,varname] %in% c(mis, irr, "")),
varname]))
    countries <- unique(data$COUNTRY)
    ns_header1 = c("25-64_n_nonmissing", "25-64_nmissing", "men_25-64_
nmissing", "women_25-64_nmissing" )
    for (country in countries) {
        surveys <- sort(unique(data[data$COUNTRY == country,]$SURVEY))
        for (survey in surveys) {
            rowss <- as.data.frame(cbind("", "", "" , "", ""),
stringsAsFactors  = F )
            names(rowss) = c("COUNTRY", "SURVEY", "FIELD", "FIELD2",
varname)
            firstrow=T
            if (varname != "ELIG") {
                selected <- data[data$COUNTRY == country & data$SURVEY
== survey & data$AGE >= 25 & data$AGE < 65, ]
                selected <- selected[ !(selected[, varname] %in% c(irr)
) & selected[, varname] != "" ,]
            }
            if (varname == "ELIG" ) {
                selected <- data[data$COUNTRY == country & data$SURVEY
== survey , ]
            }
             nall1 <- nrow(selected)
            n1=NULL; n2=NULL; prop1=NULL; prop2=NULL
            n11=NULL; n21=NULL; n31=NULL; n41=NULL
            n12=NULL; n22=NULL; n32=NULL; n42=NULL
            prop11=NULL; prop21=NULL; prop31=NULL; prop41=NULL
            prop12=NULL; prop22=NULL; prop32=NULL; prop42=NULL
            nmen  <-   nrow(selected[ selected[, varname] %in% values &
selected$SEX == "1",])
            nwomen <-   nrow(selected[ selected[, varname] %in% values &
selected$SEX == "2",])
            if (varname == "ELIG"  | varname == "EXAM1" | varname ==
```

```
"QUEST1" | varname == "EXAMQUEST") {
                 nall <- nrow(selected[(selected$SEX %in% c("1","2")) &
selected$AGE != "999" & selected$AGE != ""  &  selected[, varname] %in%
values  & selected$AGE >= 25 & selected$AGE < 65, ])
                nmis <-  nrow(selected[ !(selected$SEX %in% c("1","2"))
| selected$AGE == ""| selected$AGE == "999"  |  !(selected[, varname]
%in% values)  | selected$AGE < 25 | selected$AGE >= 65 , ])
                nmism = ""
                nmisw  =""
                selected = selected[selected$SEX %in% c("1","2") &
selected$AGE >= 25 & selected$AGE < 65,]
            }
            else {
                nall <- nrow(selected[!(selected[, varname] %in% c(mis)
), ])
                nmis <-  nrow(selected[ selected[, varname] %in% mis ,])
                 nmism <- nrow(selected[ selected[, varname] %in% mis &
selected$SEX == "1",])
                nmisw <- nrow(selected[ selected[, varname] %in% mis &
selected$SEX == "2",])
            }
            ### converting to numeric, missing values
            selected = selected[ !(selected[, varname] %in% c(mis)) , ]
            selected[, varname] <- as.numeric(selected[, varname])
            age1 <-  selected[selected$AGE >= 25 & selected$AGE < 35 , ]
            age2 <-  selected[selected$AGE >= 35 & selected$AGE < 45 , ]
            age3 <-  selected[selected$AGE >= 45 & selected$AGE < 55 , ]
            age4 <-  selected[selected$AGE >= 55 & selected$AGE < 65 , ]
            if (nmen !=0 ) {
                n1["all"] <- length(na.omit(selected[selected$SEX ==
"1", varname]))
                n11["all"] <- length(na.omit(age1[age1$SEX == "1",
varname]))
                n21["all"] <- length(na.omit(age2[age2$SEX == "1",
varname]))
                n31["all"] <- length(na.omit(age3[age3$SEX == "1",
varname]))
                n41["all"] <- length(na.omit(age4[age4$SEX == "1",
varname]))
            }
            if (nwomen !=0 ) {
                n2["all"] <- length(na.omit(selected[selected$SEX ==
"2", varname]))
                n12["all"] <- length(na.omit(age1[age1$SEX == "2",
varname]))
                n22["all"] <- length(na.omit(age2[age2$SEX == "2",
varname]))
                n32["all"] <- length(na.omit(age3[age3$SEX == "2",
varname]))
                n42["all"] <- length(na.omit(age4[age4$SEX == "2",
varname]))
            }
            ns1 = c(nall, nmis, nmism, nmisw)
            rows1 = as.data.frame(cbind(country, survey, ns_header1,
"all", ns1) , stringsAsFactors=F)
```

```
            names(rows1) <- c("COUNTRY", "SURVEY", "FIELD", "FIELD2", 
varname)
            ns1=NULL
            rowss = merge(rowss,rows1, all=T)
            if (nall!=0) {
                for (value in values) {
                    if ( nall == 0) {
                        n1[value] <- ""; n2[value] <- ""
                        n11[value] <- "";  n21[value] <- ""; n31[value] 
<- ""; n41[value] <- ""; n12[value] <- ""; n22[value] <- ""; n32[value] 
<- ""; n42[value] <- ""
                        prop1[value] <- ""; prop2[value] <- ""; 
prop11[value] <- ""; prop21[value] <- ""; prop31[value] <- ""; 
prop41[value] <- ""
                        prop12[value] <- ""; prop22[value] <- ""; 
prop32[value] <- ""; prop42[value] <- ""
                    }
                    else {
                        if (nmen !=0 ) {
                            n1[value] <- length(na.
omit(selected[selected$SEX == "1" & selected[,varname] == value, 
varname]))
                            n11[value] <- length(na.omit(age1[age1$SEX 
== "1" & age1[,varname] == value, varname]))
                            n21[value] <- length(na.omit(age2[age2$SEX 
== "1" & age2[,varname] == value, varname]))
                            n31[value] <- length(na.omit(age3[age3$SEX 
== "1" & age3[,varname] == value, varname]))
                            n41[value] <- length(na.omit(age4[age4$SEX 
== "1" & age4[,varname] == value, varname]))
                            prop11[value] <- 100*(n11[value] / 
n11["all"])

                            prop21[value] <- 100*(n21[value] / 
n21["all"] )

                            prop31[value] <-  100*(n31[value] / 
n31["all"] )

                            prop41[value] <-  100*(n41[value] / 
n41["all"] )

                            prop1[value] <- mean(c(prop11[value], 
prop21[value], prop31[value], prop41[value]))
                            prop1[value] =  format(round(as.
numeric(prop1[value]), 1), nsmall=1)
                        }
                        else {
                            prop1[value] = ""
                            prop11[value] = ""
                            prop21[value] = ""
                            prop31[value] = ""
                            prop41[value] = ""
                            n1[value] = "0"
                            n11[value] = "0"
                            n21[value] = "0"
                            n31[value] = "0"
                            n41[value] = "0"
                        }
```

```
                        if (nwomen !=0 ) {
                            n2[value] <- length(na.
omit(selected[selected$SEX == "2" & selected[,varname] == value,
varname]))
                            n12[value] <- length(na.omit(age1[age1$SEX
== "2" & age1[,varname] == value, varname]))
                            n22[value] <- length(na.omit(age2[age2$SEX
== "2" & age2[,varname] == value, varname]))
                            n32[value] <- length(na.omit(age3[age3$SEX
== "2" & age3[,varname] == value, varname]))
                            n42[value] <- length(na.omit(age4[age4$SEX
== "2" & age4[,varname] == value, varname]))
                            prop12[value] <- 100*(n12[value] /
n12["all"])

                            prop22[value] <- 100*(n22[value] /
n22["all"] )

                            prop32[value] <-  100*(n32[value] /
n32["all"] )

                            prop42[value] <-  100*(n42[value] /
n42["all"] )

                            prop2[value] <- mean(c(prop12[value],
prop22[value], prop32[value], prop42[value]))
                            prop2[value] =  format(round(as.
numeric(prop2[value]), 1), nsmall=1)
                        }
                        else {
                            prop2[value] = ""
                            prop12[value] = ""
                            prop22[value] = ""
                            prop32[value] = ""
                            prop42[value] = ""
                            n2[value] = "0"
                            n12[value] = "0"
                            n22[value] = "0"
                            n32[value] = "0"
                            n42[value] = "0"
                        }
                    }
                 }
                ns2 = rbind( cbind("men_25-34_n",n11), cbind("men_35-
44_n",n21), cbind("men_45-54_n",n31), cbind("men_55-64_n",n41),
cbind("men_25-64_n",n1))
                ns2 = cbind(rownames(ns2), ns2)
                rownames(ns2) <-NULL
                rows2 = as.data.frame(cbind(country, survey,  ns2) ,
stringsAsFactors=F)
                names(rows2) <- c("COUNTRY", "SURVEY", "FIELD2",
"FIELD", varname)
                ns2=NULL
                ns3 = rbind( cbind("women_25-34_n",n12),
cbind("women_35-44_n",n22), cbind("women_45-54_n",n32), cbind("women_55-
64_n",n42), cbind("women_25-64_n",n2))
                ns3 = cbind(rownames(ns3), ns3)
                rownames(ns3) <-NULL
                rows3 = as.data.frame(cbind(country, survey,  ns3) ,
```

```
stringsAsFactors=F)
                names(rows3) <- c("COUNTRY", "SURVEY", "FIELD2",
"FIELD", varname)
                ns3=NULL
                ns4 = rbind( cbind("men_25-34_percent",prop11),
cbind("men_35-44_percent",prop21), cbind("men_45-54_percent",prop31),
cbind("men_55-64_percent",prop41), cbind("men_25-64_percent",prop1))
                ns4 = cbind(rownames(ns4), ns4)
                rownames(ns4) <-NULL
                rows4 = as.data.frame(cbind(country, survey,  ns4) ,
stringsAsFactors=F)
                names(rows4) <- c("COUNTRY", "SURVEY", "FIELD2",
"FIELD", varname)
                rows4[, varname] <- format(round( as.numeric(rows4[,
varname]) , 1), nsmall=1, trim=T)
                ns4=NULL
                ns5 = rbind( cbind("women_25-34_percent",prop12),
cbind("women_35-44_percent",prop22), cbind("women_45-54_
percent",prop32), cbind("women_55-64_percent",prop42),
cbind("women_25-64_percent",prop2))
                ns5 = cbind(rownames(ns5), ns5)
                rownames(ns5) <-NULL
                rows5 = as.data.frame(cbind(country, survey,  ns5) ,
stringsAsFactors=F)
                names(rows5) <- c("COUNTRY", "SURVEY", "FIELD2",
"FIELD", varname)
                rows5[, varname] <- format(round( as.numeric(rows5[,
varname]) , 1), nsmall=1, trim=T)
                ns5=NULL
                rowss = merge(rowss,rows2, all=T)
                rowss = merge(rowss,rows3, all=T)
                rowss = merge(rowss,rows4, all=T)
                rowss = merge(rowss,rows5, all=T)
            }
            rows1=NULL
            rows2=NULL
            rows3=NULL
            rows4=NULL
            rows5=NULL
             if (firstrow) {
                    rowss = rowss[-c(1),]
                    firstrow=F
             }
            if (country==countries[1] & survey==surveys[1]) {
                rows = rowss
            }
            else {
                rows= merge(rows,rowss, all=T)
            }
            rowss=NULL
        }
    }

    ## TOTAL line
    country = "ALL"
```

```
    survey = "ALL"
    if (varname == "ELIG" | varname == "EXAM1" | varname == "QUEST1" |
varname == "EXAMQUEST") {
        selected <- data
    }
    else {
        selected <- data[data$AGE >= 25 & data$AGE < 65, ]
        selected <- selected[ !(selected[, varname] %in% c(irr) ) &
selected[, varname] != " " ,]
    }
    nall1 <- nrow(selected)
    n1=NULL; n2=NULL; prop1=NULL; prop2=NULL
    n11=NULL; n21=NULL; n31=NULL; n41=NULL
    n12=NULL; n22=NULL; n32=NULL; n42=NULL
    prop11=NULL; prop21=NULL; prop31=NULL; prop41=NULL
    prop12=NULL; prop22=NULL; prop32=NULL; prop42=NULL
    nmen  <-   nrow(selected[ selected[, varname] %in% values &
selected$SEX == "1",])
    nwomen <-   nrow(selected[ selected[, varname] %in% values &
selected$SEX == "2",])
    if (varname == "ELIG" | varname == "EXAM1" | varname == "QUEST1" |
varname == "EXAMQUEST") {
        nall <- nrow(selected[(selected$SEX %in% c("1","2")) &
selected$AGE != "999" & selected$AGE != ""  &  selected[, varname] %in%
values  & selected$AGE >= 25 & selected$AGE < 65, ])
        nmis <-  nrow(selected[ !(selected$SEX %in% c("1","2")) |
selected$AGE == ""| selected$AGE == "999"   |  !(selected[, varname] %in%
values)  | selected$AGE < 25 | selected$AGE >= 65 , ])
       nmen  <-   nrow(selected[ selected[, varname] %in% values &
selected$SEX == "1" & selected$AGE != "999"  & selected$AGE != "" &
selected$AGE >= 25 & selected$AGE < 65,])
       nwomen <-   nrow(selected[ selected[, varname] %in% values &
selected$SEX == "2" & selected$AGE != "999"  & selected$AGE != "" &
selected$AGE >= 25 & selected$AGE < 65,])
       nmism = ""
       nmisw  =""
       selected = selected[selected$SEX %in% c("1","2") & selected$AGE
>= 25 & selected$AGE < 65,]
    }
    else {
       nall  <-   nrow(selected[ selected[, varname] %in% values,])
       nmis <- nrow(selected[ selected[, varname] %in% mis,])
       nmism <- nrow(selected[ selected[, varname] %in% mis &
selected$SEX == "1",])
       nmisw <- nrow(selected[ selected[, varname] %in% mis &
selected$SEX == "2",])
    }
    selected = selected[ !(selected[, varname] %in% c(mis)) , ]
    selected[, varname] <- as.numeric(selected[, varname])
    age1 <-  selected[selected$AGE >= 25 & selected$AGE < 35 , ]
    age2 <-  selected[selected$AGE >= 35 & selected$AGE < 45 , ]
    age3 <-  selected[selected$AGE >= 45 & selected$AGE < 55 , ]
    age4 <-  selected[selected$AGE >= 55 & selected$AGE < 65 , ]
    if (nmen !=0 ) {
        n1["all"] <- length(na.omit(selected[selected$SEX == "1",
```

```
varname]))
        n11["all"] <- length(na.omit(age1[age1$SEX == "1", varname]))
        n21["all"] <- length(na.omit(age2[age2$SEX == "1", varname]))
        n31["all"] <- length(na.omit(age3[age3$SEX == "1", varname]))
        n41["all"] <- length(na.omit(age4[age4$SEX == "1", varname]))
    }
    if (nwomen !=0 ) {
        n2["all"] <- length(na.omit(selected[selected$SEX == "2",
varname]))
        n12["all"] <- length(na.omit(age1[age1$SEX == "2", varname]))
        n22["all"] <- length(na.omit(age2[age2$SEX == "2", varname]))
        n32["all"] <- length(na.omit(age3[age3$SEX == "2", varname]))
        n42["all"] <- length(na.omit(age4[age4$SEX == "2", varname]))
    }
    ns1 = c(nall, nmis, nmism, nmisw)
    rows1 = as.data.frame(cbind(country, survey, ns_header1, "all", ns1)
, stringsAsFactors=F)
    names(rows1) <- c("COUNTRY", "SURVEY", "FIELD", "FIELD2", varname)
    ns1=NULL
    rows = merge(rows,rows1, all=T)
    rows1=NULL

    if (nall != 0) {
        for (value in values) {
            if ( nall == 0) {
                n1[value] <- ""; n2[value] <- ""
                n11[value] <- "";  n21[value] <- ""; n31[value] <- "";
n41[value] <- ""; n12[value] <- ""; n22[value] <- ""; n32[value] <- "";
n42[value] <- ""
                prop1[value] <- ""; prop2[value] <- ""; prop11[value] <-
""; prop21[value] <- ""; prop31[value] <- ""; prop41[value] <- ""
                prop12[value] <- ""; prop22[value] <- ""; prop32[value]
<- ""; prop42[value] <- ""
            }
            else {
                if (nmen !=0 ) {
                    n1[value] <- length(na.omit(selected[selected$SEX ==
"1" & selected[,varname] == value, varname]))
                    n11[value] <- length(na.omit(age1[age1$SEX == "1" &
age1[,varname] == value, varname]))
                    n21[value] <- length(na.omit(age2[age2$SEX == "1" &
age2[,varname] == value, varname]))
                    n31[value] <- length(na.omit(age3[age3$SEX == "1" &
age3[,varname] == value, varname]))
                    n41[value] <- length(na.omit(age4[age4$SEX == "1" &
age4[,varname] == value, varname]))
                    prop11[value] <- 100*(n11[value] / n11["all"])
                    prop21[value] <- 100*(n21[value] / n21["all"] )
                    prop31[value] <-  100*(n31[value] / n31["all"] )
                    prop41[value] <-  100*(n41[value] / n41["all"] )
                    prop1[value] <- mean(c(prop11[value], prop21[value],
prop31[value], prop41[value]))
                    prop1[value] =  format(round(as.
numeric(prop1[value]), 1), nsmall=1)
                }
```

```
                else {
                    prop1[value] = ””
                    prop11[value] = ””
                    prop21[value] = ””
                    prop31[value] = ””
                    prop41[value] = ””
                    n1[value] = ”0”
                    n11[value] = ”0”
                    n21[value] = ”0”
                    n31[value] = ”0”
                    n41[value] = ”0”
                }
                if (nwomen !=0 ) {
                    n2[value] <- length(na.omit(selected[selected$SEX ==
”2” & selected[,varname] == value, varname]))
                    n12[value] <- length(na.omit(age1[age1$SEX == ”2” &
age1[,varname] == value, varname]))
                    n22[value] <- length(na.omit(age2[age2$SEX == ”2” &
age2[,varname] == value, varname]))
                    n32[value] <- length(na.omit(age3[age3$SEX == ”2” &
age3[,varname] == value, varname]))
                    n42[value] <- length(na.omit(age4[age4$SEX == ”2” &
age4[,varname] == value, varname]))
                    prop12[value] <- 100*(n12[value] / n12[”all”])
                    prop22[value] <- 100*(n22[value] / n22[”all”] )
                    prop32[value] <-  100*(n32[value] / n32[”all”] )
                    prop42[value] <-  100*(n42[value] / n42[”all”] )
                    prop2[value] <- mean(c(prop12[value], prop22[value],
prop32[value], prop42[value]))
                    prop2[value] =  format(round(as.
numeric(prop2[value]), 1), nsmall=1)
                }
                else {
                    prop2[value] = ””
                    prop12[value] = ””
                    prop22[value] = ””
                    prop32[value] = ””
                    prop42[value] = ””
                    n2[value] = ”0”
                    n12[value] = ”0”
                    n22[value] = ”0”
                    n32[value] = ”0”
                    n42[value] = ”0”
                }
            }
        }
        ns2 = rbind( cbind(”men_25-34_n”,n11), cbind(”men_35-44_n”,n21),
cbind(”men_45-54_n”,n31), cbind(”men_55-64_n”,n41), cbind(”men_25-
64_n”,n1))
        ns2 = cbind(rownames(ns2), ns2)
        rownames(ns2) <-NULL
        rows2 = as.data.frame(cbind(country, survey,  ns2) ,
stringsAsFactors=F)
        names(rows2) <- c(”COUNTRY”, ”SURVEY”, ”FIELD2”, ”FIELD”,
varname)
```

```
        ns2=NULL
        ns3 = rbind( cbind("women_25-34_n",n12), cbind("women_35-
44_n",n22), cbind("women_45-54_n",n32), cbind("women_55-64_n",n42),
cbind("women_25-64_n",n2))
        ns3 = cbind(rownames(ns3), ns3)
        rownames(ns3) <-NULL
        rows3 = as.data.frame(cbind(country, survey,  ns3) ,
stringsAsFactors=F)
        names(rows3) <- c("COUNTRY", "SURVEY", "FIELD2", "FIELD",
varname)
        ns3=NULL
        ns4 = rbind( cbind("men_25-34_percent",prop11),
cbind("men_35-44_percent",prop21), cbind("men_45-54_percent",prop31),
cbind("men_55-64_percent",prop41), cbind("men_25-64_percent",prop1))
        ns4 = cbind(rownames(ns4), ns4)
        rownames(ns4) <-NULL
        rows4 = as.data.frame(cbind(country, survey,  ns4) ,
stringsAsFactors=F)
        names(rows4) <- c("COUNTRY", "SURVEY", "FIELD2", "FIELD",
varname)
        rows4[, varname] <- format(round( as.numeric(rows4[, varname]) ,
1), nsmall=1, trim=T)
        ns4=NULL
        ns5 = rbind( cbind("women_25-34_percent",prop12),
cbind("women_35-44_percent",prop22), cbind("women_45-54_
percent",prop32), cbind("women_55-64_percent",prop42),
cbind("women_25-64_percent",prop2))
        ns5 = cbind(rownames(ns5), ns5)
        rownames(ns5) <-NULL
        rows5 = as.data.frame(cbind(country, survey,  ns5) ,
stringsAsFactors=F)
        names(rows5) <- c("COUNTRY", "SURVEY", "FIELD2", "FIELD",
varname)
        rows5[, varname] <- format(round( as.numeric(rows5[, varname]) ,
1), nsmall=1, trim=T)
        ns5=NULL
        rows = merge(rows,rows2, all=T)
        rows = merge(rows,rows3, all=T)
        rows = merge(rows,rows4, all=T)
        rows = merge(rows,rows5, all=T)
    }
    rows2=NULL
    rows3=NULL
    rows4=NULL
    rows5=NULL
    cat('Prevalences for ', varname, ' done.\n', sep='')
    return(rows)
}
```

plotdist - to plot simple histogram of the variable

```
plotdist <- function(data, varname, outpath, div, minim, maxim, mis=NA,
irr=NA, axmin=minim, axmax=maxim) {
    ### data = dataset, varname = name of the variable, outpath = path
to save the file, minim = lower limit of the variable, maxim = upper
```

```
limit of the variable,
    ### mis = code for missing value, irr = code for irrelevants, axmin
= minimum of the x-axis, axmax = minimum of the x-axis
    return=T
    variable <-data[, varname]
    novalue <- (variable %in% mis)
    totallymis <-  (variable %in% irr)
    if (sum(totallymis) + sum(novalue) == length(variable)) {
        return = FALSE
    }
    variable <- as.numeric(variable[!novalue & !totallymis])
    if (sum(variable >= minim & variable <= maxim) == 0) {
        return = FALSE
    }
    counts <- table(variable[variable >= minim & variable <= maxim])
    classes <- as.numeric(names(counts))
    divpoints <- ((classes %% div) == 0)
    if (return) {
        png(file = paste(outpath,  tolower(data$COUNTRY[1]),
data$SURVEY[1], '_', tolower(varname), '.png', sep=''), width=800,
height=600)
        plot(classes, counts, type='h', axes=FALSE, main=paste(varname,
' (', data$COUNTRY[1], 'survey', data$SURVEY[1], ')'), xlab=varname,
ylab='N', xlim=c(minim, maxim), ylim=c(0, max(counts)))
        points(classes, counts, cex=0.6)
        # Number of outliers in the right:
        outl <- sum(variable > maxim)
        if (outl > 0)
            mtext(paste(outl, ' obs.\n>', maxim, sep=''), side=1,
line=3, at=maxim, cex=0.9)
        # Number of outliers in the left:
        outl <- sum(variable < minim)
        if (outl > 0)
            mtext(paste(outl, ' obs.\n<', minim, sep=''), side=1,
line=3, at=minim, cex=0.9)
        ## Number of observations and missing values
        mtext(paste('n_relevant = ',length(variable) + sum(novalue,
na.rm=T), sep=''), side=3, line=-2, cex=0.9, adj=1)
        mtext(paste(sum(novalue), ' missing', sep=''), side=3, line=-3,
cex=0.9, adj=1)
        box()
        axis(1, at=seq(axmin, axmax, by=div), labels=seq(axmin, axmax,
by=div), cex.axis=0.9)
        axis(2, cex.axis=0.9)
        mtext(paste('Updated on', Sys.Date()), side=3, line=0.3, adj=1,
cex=0.9)
        dev.off()
    }
    variable = NULL
}
```

drawtable_pr - to derive participation rates by survey, sex and 10-year age-group

```
drawtable_pr <- function(data)  {
    ### data = dataset
    # tab, output table
    tab <- NULL
    countries <- sort(unique(data$COUNTRY))
    exclude <- ""
    for (country in countries) {
        surveys <- sort(unique(data[data$COUNTRY == country,]$SURVEY))
        for (survey in surveys) {
            #### both sex
            selected <- data[data$COUNTRY == country & data$SURVEY ==
survey & data$AGE >= 25 & data$AGE < 65 & data$SEX %in% c("1", "2"),]
            #### 25-64 years
            nall <- nrow(selected)
            if (is.null(tab)) {
                tab <- as.data.frame(cbind(country, survey, "all", "25-
64"), stringsAsFactors  = F )
                tab[,dim(tab)[2]+1] <- nall
                names(tab) <- c("COUNTRY", "SURVEY", "SEX","AGE", "N_
NONMISSING")
            }
            else {
                if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="25-64",]) != 0 ) {
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all" & tab$AGE=="25-64", "N_NONMISSING"] <- nall
                }
                else {
                    tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, "all", "25-64")
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="25-64", "N_NONMISSING"] <- nall
                }
            }
            nelig <-  nrow(selected[ selected$ELIG == "1"  ,])
            nnonelig <-  nrow(selected[ selected$ELIG == "2"  ,])
            nunres <-  nrow(selected[ selected$ELIG == "3"  ,])

            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="25-64", "N_ELIGIBLE"] <- as.integer(nelig)
            tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="25-64", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
            tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="25-64", "N_UNRESOLVED"] <-
as.integer(nunres)

            nexam <-  nrow(selected[ selected$EXAM1 == "1"  ,])
            nquest <-  nrow(selected[  selected$QUEST1 == "1"  ,])
            nexamquest <-  nrow(selected[  selected$EXAMQUEST == "1"
,])
```

```
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all" & tab$AGE=="25-64" , "N_EXAM1"] <- as.integer(nexam)
            tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-64", "N_QUEST1"] <- as.integer(nquest)
            tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-64", "N_EXAMQUEST"] <-
as.integer(nexamquest )

            pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
            pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
            pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
            copr_examquest <-  round( 100*(nexamquest / (nelig)),
digits=1 )
            contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )

           if (nexamquest != nelig) {
               tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="25-64", "PR_EXAM"] <- pr_exam
               tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-64", "PR_QUEST"] <- pr_quest
               tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-64", "PR_EXAMQUEST"] <- pr_examquest
               tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-64", "COOPERATE"] <- copr_examquest
               tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-64", "CONTACT_RATE"] <- contact
           }
           else {
               exclude = c(exclude,paste(country, survey, sep=""))
           }

           age1 <-  selected[selected$SEX %in% c("1", "2")   &
selected$AGE >= 25 & selected$AGE < 35 , ]
           age2 <-  selected[selected$SEX %in% c("1", "2")   &
selected$AGE >= 35 & selected$AGE < 45 , ]
           age3 <-  selected[selected$SEX %in% c("1", "2")   &
selected$AGE >= 45 & selected$AGE < 55 , ]
           age4 <-  selected[selected$SEX %in% c("1", "2")   &
selected$AGE >= 55 & selected$AGE < 65 , ]

           #### 25-34 years
           nall <- nrow(age1)
           nelig <-  nrow(age1[ age1$ELIG == "1"  ,])
           nnonelig <-  nrow(age1[ age1$ELIG == "2"  ,])
           nunres <-  nrow(age1[ age1$ELIG == "3"  ,])
           nexam <-  nrow(age1[ age1$EXAM1 == "1"  ,])
           nquest <-  nrow(age1[ age1$QUEST1 == "1"  ,])
           nexamquest <-  nrow(age1[ age1$EXAMQUEST == "1"  ,])
           pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
           pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
           pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
           copr_examquest <-  round( 100*(nexamquest / (nelig)),
```

```
digits=1 )
           contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )
           if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all" & tab$AGE=="25-34",]) != 0 ) {
              tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="25-34", "N_NONMISSING"] <- nall
            }
            else {
                tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, "all", "25-34")
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="25-34", "N_NONMISSING"] <- nall
            }
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="25-34", "N_ELIGIBLE"] <- as.integer(nelig)
            tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="25-34", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
            tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="25-34", "N_UNRESOLVED"] <-
as.integer(nunres)
             tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="25-34", "N_EXAM1"] <- as.integer(nexam)
            tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-34", "N_QUEST1"] <- as.integer(nquest)
            tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-34", "N_EXAMQUEST"] <-
as.integer(nexamquest )
             if (nexamquest != nelig) {
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="25-34", "PR_EXAM"] <- pr_exam
               tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-34", "PR_QUEST"] <- pr_quest
               tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-34", "PR_EXAMQUEST"] <- pr_examquest
               tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-34", "COOPERATE"] <- copr_examquest
               tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="25-34", "CONTACT_RATE"] <- contact
           }
           #### 35-44 years
               nall <- nrow(age2)
               nelig <-  nrow(age2[ age2$ELIG == "1"  ,])
               nnonelig <-  nrow(age2[ age2$ELIG == "2"  ,])
               nunres <-  nrow(age2[ age2$ELIG == "3"  ,])
               nexam <-  nrow(age2[  age2$EXAM1 == "1"  ,])
               nquest <-  nrow(age2[  age2$QUEST1 == "1"  ,])
               nexamquest <-  nrow(age2[ age2$EXAMQUEST == "1"  ,])
               pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1
)
               pr_quest <- round( 100*(nquest / (nelig+nunres)),
digits=1 )
               pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
               copr_examquest <-  round( 100*(nexamquest / (nelig)),
```

```
digits=1 )
               contact <-  round( 100*(nelig / (nelig+nunres)), digits=1
)
               if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all" & tab$AGE=="35-44",]) != 0 ) {
                  tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="35-44", "N_NONMISSING"] <- nall
                }
                else {
                    tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, "all", "35-44")
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="35-44", "N_NONMISSING"] <- nall
                }
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="35-44", "N_ELIGIBLE"] <- as.integer(nelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="35-44", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="35-44", "N_UNRESOLVED"] <-
as.integer(nunres)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="35-44", "N_EXAM1"] <- as.integer(nexam)
                tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="35-44", "N_QUEST1"] <- as.integer(nquest)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all" & tab$AGE=="35-44", "N_EXAMQUEST"] <-
as.integer(nexamquest )
                if (nexamquest != nelig) {
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="35-44", "PR_EXAM"] <- pr_exam
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="35-44", "PR_QUEST"] <- pr_quest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="35-44", "PR_EXAMQUEST"] <- pr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="35-44", "COOPERATE"] <- copr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="35-44", "CONTACT_RATE"] <- contact
               }

                #### 45-54 years
               nall <- nrow(age3)
               nelig <-  nrow(age3[ age3$ELIG == "1"  ,])
               nnonelig <-  nrow(age3[ age3$ELIG == "2"  ,])
               nunres <-  nrow(age3[ age3$ELIG == "3"  ,])
               nexam <-  nrow(age3[  age3$EXAM1 == "1"  ,])
               nquest <-  nrow(age3[age3$QUEST1 == "1"  ,])
               nexamquest <-  nrow(age3[ age3$EXAMQUEST == "1"  ,])
               pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1
)
               pr_quest <- round( 100*(nquest / (nelig+nunres)),
digits=1 )
               pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
```

```
digits=1 )
               copr_examquest <-  round( 100*(nexamquest / (nelig)),
digits=1 )
               contact <-  round( 100*(nelig / (nelig+nunres)), digits=1
)
               if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all" & tab$AGE=="45-54",]) != 0 ) {
                  tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="45-54", "N_NONMISSING"] <- nall
                }
                else {
                    tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, "all", "45-54")
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="45-54", "N_NONMISSING"] <- nall
                }
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="45-54", "N_ELIGIBLE"] <- as.integer(nelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="45-54", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="45-54", "N_UNRESOLVED"] <-
as.integer(nunres)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="45-54", "N_EXAM1"] <- as.integer(nexam)
                tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="45-54", "N_QUEST1"] <- as.integer(nquest)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all" & tab$AGE=="45-54", "N_EXAMQUEST"] <-
as.integer(nexamquest )
                if (nexamquest != nelig) {
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="45-54", "PR_EXAM"] <- pr_exam
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="45-54", "PR_QUEST"] <- pr_quest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="45-54", "PR_EXAMQUEST"] <- pr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="45-54", "COOPERATE"] <- copr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="45-54", "CONTACT_RATE"] <- contact
               }

                 #### 55-64 years
               nall <- nrow(age4)
               nelig <-  nrow(age4[ age4$ELIG == "1"  ,])
               nnonelig <-  nrow(age4[ age4$ELIG == "2"  ,])
               nunres <-  nrow(age4[ age4$ELIG == "3"  ,])
               nexam <-  nrow(age4[  age4$EXAM1 == "1"  ,])
               nquest <-  nrow(age4[  age4$QUEST1 == "1"  ,])
               nexamquest <-  nrow(age4[ age4$EXAMQUEST == "1"  ,])
               pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1
)
               pr_quest <- round( 100*(nquest / (nelig+nunres)),
```

```
digits=1 )
               pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
               copr_examquest <-  round( 100*(nexamquest / (nelig)),
digits=1 )
               contact <-  round( 100*(nelig / (nelig+nunres)), digits=1
)
               if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all" & tab$AGE=="55-64",]) != 0 ) {
                  tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="55-64", "N_NONMISSING"] <- nall
                }
                else {
                    tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, "all", "55-64")
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="55-64", "N_NONMISSING"] <- nall
                }
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="55-64", "N_ELIGIBLE"] <- as.integer(nelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="55-64", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all"  & tab$AGE=="55-64", "N_UNRESOLVED"] <-
as.integer(nunres)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="55-64", "N_EXAM1"] <- as.integer(nexam)
                tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="55-64", "N_QUEST1"] <- as.integer(nquest)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX=="all" & tab$AGE=="55-64", "N_EXAMQUEST"] <-
as.integer(nexamquest )
                if (nexamquest != nelig) {
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="55-64", "PR_EXAM"] <- pr_exam
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="55-64", "PR_QUEST"] <- pr_quest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="55-64", "PR_EXAMQUEST"] <- pr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="55-64", "COOPERATE"] <- copr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX=="all" & tab$AGE=="55-64", "CONTACT_RATE"] <- contact
               }
               age1=NULL
               age2=NULL
               age3=NULL
               age4=NULL

           for (sex in c("1", "2")) {
               selected1 <-    selected[selected$SEX == sex ,]
               if (sex == "1") {
                  sex2 <- "men"
               }
```

```
                else {sex2 <- "women"}
                 nall <- nrow(selected1)
                 if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2 & tab$AGE=="25-64",]) != 0 ) {
                   tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-64", "N_NONMISSING"] <- nall
                 }
                 else {
                     tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, sex2, "25-64")
                     tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-64", "N_NONMISSING"] <- nall
                 }
                 nelig <-  nrow(selected1[ selected1$ELIG == "1"  ,])
                 nnonelig <-  nrow(selected1[ selected1$ELIG == "2"  ,])
                 nunres <-  nrow(selected1[ selected1$ELIG == "3"  ,])

                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-64", "N_ELIGIBLE"] <- as.integer(nelig)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2  & tab$AGE=="25-64", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-64", "N_UNRESOLVED"] <- as.integer(nunres)

                 nexam <-  nrow(selected1[ selected1$EXAM1 == "1"  ,])
                 nquest <-  nrow(selected1[ selected1$QUEST1 == "1"  ,])
                 nexamquest <-  nrow(selected1[selected1$EXAMQUEST == "1"
,])

                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-64", "N_EXAM1"] <- as.integer(nexam)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="25-64", "N_QUEST1"] <- as.integer(nquest)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2 & tab$AGE=="25-64", "N_EXAMQUEST"] <-
as.integer(nexamquest )

                 pr_exam <-  round( 100*(nexam / (nelig+nunres)),
digits=1 )
                 pr_quest <- round( 100*(nquest / (nelig+nunres)),
digits=1 )
                 pr_examquest <- round( 100*(nexamquest /
(nelig+nunres)), digits=1 )
                 copr_examquest <-  round( 100*(nexamquest / (nelig)),
digits=1 )
                 contact <-  round( 100*(nelig / (nelig+nunres)),
digits=1 )

                 if (nexamquest != nelig) {
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-64", "PR_EXAM"] <- pr_exam
                    tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="25-64", "PR_QUEST"] <- pr_quest
                    tab[tab$COUNTRY==country & tab$SURVEY==survey  &
```

```
tab$SEX==sex2 & tab$AGE=="25-64", "PR_EXAMQUEST"] <- pr_examquest
                    tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="25-64", "COOPERATE"] <- copr_examquest
                    tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="25-64", "CONTACT_RATE"] <- contact
               }

               age1 <-  selected1[selected1$AGE >= 25 & selected1$AGE <
35 , ]
               age2 <-  selected1[selected1$AGE >= 35 & selected1$AGE <
45 , ]
               age3 <-  selected1[selected1$AGE >= 45 & selected1$AGE <
55 , ]
               age4 <-  selected1[selected1$AGE >= 55 & selected1$AGE <
65 , ]

               #### 25-34 years
               nall <- nrow(age1)
               nelig <-  nrow(age1[ age1$ELIG == "1"  ,])
               nnonelig <-  nrow(age1[ age1$ELIG == "2"  ,])
               nunres <-  nrow(age1[ age1$ELIG == "3"  ,])
               nexam <-  nrow(age1[ age1$EXAM1 == "1"  ,])
               nquest <-  nrow(age1[ age1$QUEST1 == "1"  ,])
               nexamquest <-  nrow(age1[ age1$EXAMQUEST == "1" ,])
               pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1
)
               pr_quest <- round( 100*(nquest / (nelig+nunres)),
digits=1 )
               pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
               copr_examquest <-  round( 100*(nexamquest / (nelig)),
digits=1 )
               contact <-  round( 100*(nelig / (nelig+nunres)), digits=1
)
               if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2 & tab$AGE=="25-34",]) != 0 ) {
                   tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-34", "N_NONMISSING"] <- nall
                }
                else {
                    tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, sex2, "25-34")
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-34", "N_NONMISSING"] <- nall
                }
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-34", "N_ELIGIBLE"] <- as.integer(nelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2  & tab$AGE=="25-34", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-34", "N_UNRESOLVED"] <- as.integer(nunres)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-34", "N_EXAM1"] <- as.integer(nexam)
                tab[tab$COUNTRY==country & tab$SURVEY==survey  &
```

```
tab$SEX==sex2 & tab$AGE=="25-34", "N_QUEST1"] <- as.integer(nquest)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2 & tab$AGE=="25-34", "N_EXAMQUEST"] <-
as.integer(nexamquest )
                if (nexamquest != nelig) {
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-34", "PR_EXAM"] <- pr_exam
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="25-34", "PR_QUEST"] <- pr_quest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="25-34", "PR_EXAMQUEST"] <- pr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="25-34", "COOPERATE"] <- copr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="25-34", "CONTACT_RATE"] <- contact
               }

               #### 35-44 years
               nall <- nrow(age2)
               nelig <-  nrow(age2[ age2$ELIG == "1"  ,])
               nnonelig <-  nrow(age2[ age2$ELIG == "2"  ,])
               nunres <-  nrow(age2[ age2$ELIG == "3"  ,])
               nexam <-  nrow(age2[  age2$EXAM1 == "1"  ,])
               nquest <-  nrow(age2[ age2$QUEST1 == "1"  ,])
               nexamquest <-  nrow(age2[ age2$EXAMQUEST == "1"  ,])
               pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1
)
               pr_quest <- round( 100*(nquest / (nelig+nunres)),
digits=1 )
               pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
               copr_examquest <-  round( 100*(nexamquest / (nelig)),
digits=1 )
               contact <-  round( 100*(nelig / (nelig+nunres)), digits=1
)
               if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2 & tab$AGE=="35-44",]) != 0 ) {
                  tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="35-44", "N_NONMISSING"] <- nall
                }
                else {
                    tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, sex2, "35-44")
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="35-44", "N_NONMISSING"] <- nall
                }
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="35-44", "N_ELIGIBLE"] <- as.integer(nelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2  & tab$AGE=="35-44", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="35-44", "N_UNRESOLVED"] <- as.integer(nunres)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="35-44", "N_EXAM1"] <- as.integer(nexam)
```

```
                tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="35-44", "N_QUEST1"] <- as.integer(nquest)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2 & tab$AGE=="35-44", "N_EXAMQUEST"] <-
as.integer(nexamquest )
                if (nexamquest != nelig) {
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="35-44", "PR_EXAM"] <- pr_exam
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="35-44", "PR_QUEST"] <- pr_quest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="35-44", "PR_EXAMQUEST"] <- pr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="35-44", "COOPERATE"] <- copr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="35-44", "CONTACT_RATE"] <- contact
               }

                #### 45-54 years
               nall <- nrow(age3)
               nelig <-  nrow(age3[ age3$ELIG == "1"  ,])
               nnonelig <-  nrow(age3[ age3$ELIG == "2"  ,])
               nunres <-  nrow(age3[ age3$ELIG == "3"  ,])
               nexam <-  nrow(age3[ age3$EXAM1 == "1"  ,])
               nquest <-  nrow(age3[ age3$QUEST1 == "1"  ,])
               nexamquest <-  nrow(age3[  age3$EXAMQUEST == "1"  ,])
               pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1
)
               pr_quest <- round( 100*(nquest / (nelig+nunres)),
digits=1 )
               pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
               copr_examquest <-  round( 100*(nexamquest / (nelig)),
digits=1 )
               contact <-  round( 100*(nelig / (nelig+nunres)), digits=1
)
               if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2 & tab$AGE=="45-54",]) != 0 ) {
                  tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="45-54", "N_NONMISSING"] <- nall
                }
                else {
                    tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, sex2, "45-54")
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="45-54", "N_NONMISSING"] <- nall
                }
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="45-54", "N_ELIGIBLE"] <- as.integer(nelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2  & tab$AGE=="45-54", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="45-54", "N_UNRESOLVED"] <- as.integer(nunres)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
```

```
tab$SEX==sex2  & tab$AGE=="45-54", "N_EXAM1"] <- as.integer(nexam)
                tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="45-54", "N_QUEST1"] <- as.integer(nquest)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2 & tab$AGE=="45-54", "N_EXAMQUEST"] <-
as.integer(nexamquest )
                if (nexamquest != nelig) {
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="45-54", "PR_EXAM"] <- pr_exam
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="45-54", "PR_QUEST"] <- pr_quest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="45-54", "PR_EXAMQUEST"] <- pr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="45-54", "COOPERATE"] <- copr_examquest
                   tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="45-54", "CONTACT_RATE"] <- contact
               }

               #### 55-64 years
               nall <- nrow(age4)
               nelig <-  nrow(age4[ age4$ELIG == "1"  ,])
               nnonelig <-  nrow(age4[ age4$ELIG == "2"  ,])
               nunres <-  nrow(age4[ age4$ELIG == "3"  ,])
               nexam <-  nrow(age4[  age4$EXAM1 == "1"  ,])
               nquest <-  nrow(age4[ age4$QUEST1 == "1"  ,])
               nexamquest <-  nrow(age4[ age4$EXAMQUEST == "1" ,])
               pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1
)
               pr_quest <- round( 100*(nquest / (nelig+nunres)),
digits=1 )
               pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
               copr_examquest <-  round( 100*(nexamquest / (nelig)),
digits=1 )
               contact <-  round( 100*(nelig / (nelig+nunres)), digits=1
)
               if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2 & tab$AGE=="55-64",]) != 0 ) {
                  tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="55-64", "N_NONMISSING"] <- nall
                }
                else {
                    tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX",
"AGE")] <- c(country, survey, sex2, "55-64")
                    tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="55-64", "N_NONMISSING"] <- nall
                }
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="55-64", "N_ELIGIBLE"] <- as.integer(nelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2  & tab$AGE=="55-64", "N_NONELIGIBLE"] <-
as.integer(nnonelig)
                tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="55-64", "N_UNRESOLVED"] <- as.integer(nunres)
```

```
                 tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="55-64", "N_EXAM1"] <- as.integer(nexam)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="55-64", "N_QUEST1"] <- as.integer(nquest)
                 tab[tab$COUNTRY==country & tab$SURVEY==survey
& tab$SEX==sex2 & tab$AGE=="55-64", "N_EXAMQUEST"] <-
as.integer(nexamquest )
                 if (nexamquest != nelig) {
                     tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="55-64", "PR_EXAM"] <- pr_exam
                    tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="55-64", "PR_QUEST"] <- pr_quest
                    tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="55-64", "PR_EXAMQUEST"] <- pr_examquest
                    tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="55-64", "COOPERATE"] <- copr_examquest
                    tab[tab$COUNTRY==country & tab$SURVEY==survey  &
tab$SEX==sex2 & tab$AGE=="55-64", "CONTACT_RATE"] <- contact
                 }
                selected1=NULL
                age1=NULL
                age2=NULL
                age3=NULL
                age4=NULL
            }
         }
     }

     ##### TOTAL LINE
     country="ALL"
     survey="ALL"
     #### both sex , not surveys with incomplete eligibility dasta
     selected <- data[!(paste(data$COUNTRY, data$SURVEY, sep="") %in%
exclude) & data$AGE >= 25 & data$AGE < 65 & data$SEX %in% c("1", "2"),]
     #### 25-64 years
     nall <- nrow(selected)
     if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="25-64",]) != 0 ) {
         tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all" &
tab$AGE=="25-64", "N_NONMISSING"] <- nall
     }
     else {
         tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX", "AGE")] <-
c(country, survey, "all", "25-64")
         tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="25-64", "N_NONMISSING"] <- nall
     }
     nelig <-  nrow(selected[ selected$ELIG == "1"  ,])
     nnonelig <-  nrow(selected[ selected$ELIG == "2"  ,])
     nunres <-  nrow(selected[ selected$ELIG == "3"  ,])

     tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-64", "N_ELIGIBLE"] <- as.integer(nelig)
     tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-64", "N_NONELIGIBLE"] <- as.integer(nnonelig)
```

```
    tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-64", "N_UNRESOLVED"] <- as.integer(nunres)

    nexam <-  nrow(selected[ selected$EXAM1 == "1"  ,])
    nquest <-  nrow(selected[  selected$QUEST1 == "1"  ,])
    nexamquest <-  nrow(selected[  selected$EXAMQUEST == "1"  ,])

    tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all" &
tab$AGE=="25-64" , "N_EXAM1"] <- as.integer(nexam)
    tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-64", "N_QUEST1"] <- as.integer(nquest)
    tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-64", "N_EXAMQUEST"] <- as.integer(nexamquest )

    pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
    pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
    pr_examquest <- round( 100*(nexamquest / (nelig+nunres)), digits=1 )
    copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
    contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )

   tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-64", "PR_EXAM"] <- pr_exam
   tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-64", "PR_QUEST"] <- pr_quest
   tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-64", "PR_EXAMQUEST"] <- pr_examquest
   tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-64", "COOPERATE"] <- copr_examquest
   tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-64", "CONTACT_RATE"] <- contact

   age1 <-  selected[selected$SEX %in% c("1", "2")  & selected$AGE >= 25
& selected$AGE < 35 , ]
   age2 <-  selected[selected$SEX %in% c("1", "2")  & selected$AGE >= 35
& selected$AGE < 45 , ]
   age3 <-  selected[selected$SEX %in% c("1", "2")  & selected$AGE >= 45
& selected$AGE < 55 , ]
   age4 <-  selected[selected$SEX %in% c("1", "2")  & selected$AGE >= 55
& selected$AGE < 65 , ]

   #### 25-34 years
   nall <- nrow(age1)
   nelig <-  nrow(age1[ age1$ELIG == "1"  ,])
   nnonelig <-  nrow(age1[ age1$ELIG == "2"  ,])
   nunres <-  nrow(age1[ age1$ELIG == "3"  ,])
   nexam <-  nrow(age1[ age1$EXAM1 == "1"  ,])
   nquest <-  nrow(age1[ age1$QUEST1 == "1"  ,])
   nexamquest <-  nrow(age1[ age1$EXAMQUEST == "1"  ,])
   pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
   pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
   pr_examquest <- round( 100*(nexamquest / (nelig+nunres)), digits=1 )
   copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
   contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )
   if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all" & tab$AGE=="25-34",]) != 0 ) {
```

```
      tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-34", "N_NONMISSING"] <- nall
    }
    else {
        tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX", "AGE")] <-
c(country, survey, "all", "25-34")
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="25-34", "N_NONMISSING"] <- nall
    }
    tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-34", "N_ELIGIBLE"] <- as.integer(nelig)
    tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-34", "N_NONELIGIBLE"] <- as.integer(nnonelig)
    tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-34", "N_UNRESOLVED"] <- as.integer(nunres)
     tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-34", "N_EXAM1"] <- as.integer(nexam)
    tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-34", "N_QUEST1"] <- as.integer(nquest)
    tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-34", "N_EXAMQUEST"] <- as.integer(nexamquest )
    tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"  &
tab$AGE=="25-34", "PR_EXAM"] <- pr_exam
   tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-34", "PR_QUEST"] <- pr_quest
   tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-34", "PR_EXAMQUEST"] <- pr_examquest
   tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-34", "COOPERATE"] <- copr_examquest
   tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="25-34", "CONTACT_RATE"] <- contact
   #### 35-44 years
       nall <- nrow(age2)
       nelig <-  nrow(age2[ age2$ELIG == "1"  ,])
       nnonelig <-  nrow(age2[ age2$ELIG == "2"  ,])
       nunres <-  nrow(age2[ age2$ELIG == "3"  ,])
       nexam <-  nrow(age2[  age2$EXAM1 == "1"  ,])
       nquest <-  nrow(age2[  age2$QUEST1 == "1"  ,])
       nexamquest <-  nrow(age2[ age2$EXAMQUEST == "1"  ,])
       pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
       pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
       pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
       copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
       contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )
       if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all" & tab$AGE=="35-44",]) != 0 ) {
          tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="35-44", "N_NONMISSING"] <- nall
        }
        else {
            tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX", "AGE")] <-
c(country, survey, "all", "35-44")
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="35-44", "N_NONMISSING"] <- nall
```

```
        }
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==”all”
& tab$AGE==”35-44”, ”N_ELIGIBLE”] <- as.integer(nelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==”all”
& tab$AGE==”35-44”, ”N_NONELIGIBLE”] <- as.integer(nnonelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==”all”
& tab$AGE==”35-44”, ”N_UNRESOLVED”] <- as.integer(nunres)
         tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==”all”
& tab$AGE==”35-44”, ”N_EXAM1”] <- as.integer(nexam)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==”all”
& tab$AGE==”35-44”, ”N_QUEST1”] <- as.integer(nquest)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==”all”
& tab$AGE==”35-44”, ”N_EXAMQUEST”] <- as.integer(nexamquest )
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==”all”
& tab$AGE==”35-44”, ”PR_EXAM”] <- pr_exam
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==”all” &
tab$AGE==”35-44”, ”PR_QUEST”] <- pr_quest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==”all” &
tab$AGE==”35-44”, ”PR_EXAMQUEST”] <- pr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==”all” &
tab$AGE==”35-44”, ”COOPERATE”] <- copr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==”all” &
tab$AGE==”35-44”, ”CONTACT_RATE”] <- contact

        #### 45-54 years
       nall <- nrow(age3)
       nelig <-  nrow(age3[ age3$ELIG == ”1”  ,])
       nnonelig <-  nrow(age3[ age3$ELIG == ”2”  ,])
       nunres <-  nrow(age3[ age3$ELIG == ”3”  ,])
       nexam <-  nrow(age3[  age3$EXAM1 == ”1”  ,])
       nquest <-  nrow(age3[age3$QUEST1 == ”1”  ,])
       nexamquest <-  nrow(age3[ age3$EXAMQUEST == ”1”  ,])
       pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
       pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
       pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
       copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
       contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )
       if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==”all” & tab$AGE==”45-54”,]) != 0 ) {
          tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==”all”
& tab$AGE==”45-54”, ”N_NONMISSING”] <- nall
        }
        else {
            tab[dim(tab)[1]+1 , c(”COUNTRY”, ”SURVEY”, ”SEX”, ”AGE”)] <-
c(country, survey, ”all”, ”45-54”)
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==”all”  & tab$AGE==”45-54”, ”N_NONMISSING”] <- nall
        }
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==”all”
& tab$AGE==”45-54”, ”N_ELIGIBLE”] <- as.integer(nelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==”all”
& tab$AGE==”45-54”, ”N_NONELIGIBLE”] <- as.integer(nnonelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==”all”
& tab$AGE==”45-54”, ”N_UNRESOLVED”] <- as.integer(nunres)
```

```
         tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="45-54", "N_EXAM1"] <- as.integer(nexam)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all"
& tab$AGE=="45-54", "N_QUEST1"] <- as.integer(nquest)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all"
& tab$AGE=="45-54", "N_EXAMQUEST"] <- as.integer(nexamquest )
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="45-54", "PR_EXAM"] <- pr_exam
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="45-54", "PR_QUEST"] <- pr_quest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="45-54", "PR_EXAMQUEST"] <- pr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="45-54", "COOPERATE"] <- copr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="45-54", "CONTACT_RATE"] <- contact

         #### 55-64 years
       nall <- nrow(age4)
       nelig <-  nrow(age4[ age4$ELIG == "1"  ,])
       nnonelig <-  nrow(age4[ age4$ELIG == "2"  ,])
       nunres <-  nrow(age4[ age4$ELIG == "3"  ,])
       nexam <-  nrow(age4[  age4$EXAM1 == "1"  ,])
       nquest <-  nrow(age4[  age4$QUEST1 == "1"  ,])
       nexamquest <-  nrow(age4[ age4$EXAMQUEST == "1"  ,])
       pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
       pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
       pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
       copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
       contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )
       if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all" & tab$AGE=="55-64",]) != 0 ) {
          tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="55-64", "N_NONMISSING"] <- nall
        }
        else {
            tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX", "AGE")] <-
c(country, survey, "all", "55-64")
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX=="all"  & tab$AGE=="55-64", "N_NONMISSING"] <- nall
        }
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="55-64", "N_ELIGIBLE"] <- as.integer(nelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="55-64", "N_NONELIGIBLE"] <- as.integer(nnonelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="55-64", "N_UNRESOLVED"] <- as.integer(nunres)
         tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
& tab$AGE=="55-64", "N_EXAM1"] <- as.integer(nexam)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all"
& tab$AGE=="55-64", "N_QUEST1"] <- as.integer(nquest)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all"
& tab$AGE=="55-64", "N_EXAMQUEST"] <- as.integer(nexamquest )
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX=="all"
```

```
& tab$AGE=="55-64", "PR_EXAM"] <- pr_exam
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="55-64", "PR_QUEST"] <- pr_quest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="55-64", "PR_EXAMQUEST"] <- pr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="55-64", "COOPERATE"] <- copr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX=="all" &
tab$AGE=="55-64", "CONTACT_RATE"] <- contact
       age1=NULL
       age2=NULL
       age3=NULL
       age4=NULL

   for (sex in c("1", "2")) {
       selected1 <-    selected[selected$SEX == sex ,]
       if (sex == "1") {
          sex2 <- "men"
       }
       else {sex2 <- "women"}
        nall <- nrow(selected1)
        if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2 & tab$AGE=="25-64",]) != 0 ) {
          tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2
& tab$AGE=="25-64", "N_NONMISSING"] <- nall
        }
        else {
            tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX", "AGE")] <-
c(country, survey, sex2, "25-64")
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-64", "N_NONMISSING"] <- nall
        }
        nelig <-  nrow(selected1[ selected1$ELIG == "1"  ,])
        nnonelig <-  nrow(selected1[ selected1$ELIG == "2"  ,])
        nunres <-  nrow(selected1[ selected1$ELIG == "3"  ,])

        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="25-64", "N_ELIGIBLE"] <- as.integer(nelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="25-64", "N_NONELIGIBLE"] <- as.integer(nnonelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="25-64", "N_UNRESOLVED"] <- as.integer(nunres)

        nexam <-  nrow(selected1[ selected1$EXAM1 == "1"  ,])
        nquest <-  nrow(selected1[ selected1$QUEST1 == "1"  ,])
        nexamquest <-  nrow(selected1[selected1$EXAMQUEST == "1"  ,])

        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="25-64", "N_EXAM1"] <- as.integer(nexam)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-64", "N_QUEST1"] <- as.integer(nquest)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-64", "N_EXAMQUEST"] <- as.integer(nexamquest )

        pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
```

```
        pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
        pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
        copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
        contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )

       tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="25-64", "PR_EXAM"] <- pr_exam
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-64", "PR_QUEST"] <- pr_quest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-64", "PR_EXAMQUEST"] <- pr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-64", "COOPERATE"] <- copr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-64", "CONTACT_RATE"] <- contact

       age1 <-  selected1[selected1$AGE >= 25 & selected1$AGE < 35 , ]
       age2 <-  selected1[selected1$AGE >= 35 & selected1$AGE < 45 , ]
       age3 <-  selected1[selected1$AGE >= 45 & selected1$AGE < 55 , ]
       age4 <-  selected1[selected1$AGE >= 55 & selected1$AGE < 65 , ]

       #### 25-34 years
       nall <- nrow(age1)
       nelig <-  nrow(age1[ age1$ELIG == "1"  ,])
       nnonelig <-  nrow(age1[ age1$ELIG == "2"  ,])
       nunres <-  nrow(age1[ age1$ELIG == "3"  ,])
       nexam <-  nrow(age1[ age1$EXAM1 == "1"  ,])
       nquest <-  nrow(age1[ age1$QUEST1 == "1"  ,])
       nexamquest <-  nrow(age1[ age1$EXAMQUEST == "1" ,])
       pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
       pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
       pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
       copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
       contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )
       if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2 & tab$AGE=="25-34",]) != 0 ) {
          tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2
& tab$AGE=="25-34", "N_NONMISSING"] <- nall
        }
        else {
            tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX", "AGE")] <-
c(country, survey, sex2, "25-34")
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="25-34", "N_NONMISSING"] <- nall
        }
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="25-34", "N_ELIGIBLE"] <- as.integer(nelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="25-34", "N_NONELIGIBLE"] <- as.integer(nnonelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="25-34", "N_UNRESOLVED"] <- as.integer(nunres)
         tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2
& tab$AGE=="25-34", "N_EXAM1"] <- as.integer(nexam)
```

```
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-34", "N_QUEST1"] <- as.integer(nquest)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-34", "N_EXAMQUEST"] <- as.integer(nexamquest )
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="25-34", "PR_EXAM"] <- pr_exam
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-34", "PR_QUEST"] <- pr_quest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-34", "PR_EXAMQUEST"] <- pr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-34", "COOPERATE"] <- copr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="25-34", "CONTACT_RATE"] <- contact

       #### 35-44 years
       nall <- nrow(age2)
       nelig <-  nrow(age2[ age2$ELIG == "1"  ,])
       nnonelig <-  nrow(age2[ age2$ELIG == "2"  ,])
       nunres <-  nrow(age2[ age2$ELIG == "3"  ,])
       nexam <-  nrow(age2[  age2$EXAM1 == "1"  ,])
       nquest <-  nrow(age2[ age2$QUEST1 == "1"  ,])
       nexamquest <-  nrow(age2[ age2$EXAMQUEST == "1"  ,])
       pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
       pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
       pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
       copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
       contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )
       if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2 & tab$AGE=="35-44",]) != 0 ) {
          tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2
& tab$AGE=="35-44", "N_NONMISSING"] <- nall
        }
        else {
            tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX", "AGE")] <-
c(country, survey, sex2, "35-44")
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="35-44", "N_NONMISSING"] <- nall
        }
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="35-44", "N_ELIGIBLE"] <- as.integer(nelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="35-44", "N_NONELIGIBLE"] <- as.integer(nnonelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="35-44", "N_UNRESOLVED"] <- as.integer(nunres)
         tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2
& tab$AGE=="35-44", "N_EXAM1"] <- as.integer(nexam)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="35-44", "N_QUEST1"] <- as.integer(nquest)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="35-44", "N_EXAMQUEST"] <- as.integer(nexamquest )
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="35-44", "PR_EXAM"] <- pr_exam
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
```

```
tab$AGE=="35-44", "PR_QUEST"] <- pr_quest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="35-44", "PR_EXAMQUEST"] <- pr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="35-44", "COOPERATE"] <- copr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="35-44", "CONTACT_RATE"] <- contact

        #### 45-54 years
       nall <- nrow(age3)
       nelig <-  nrow(age3[ age3$ELIG == "1"  ,])
       nnonelig <-  nrow(age3[ age3$ELIG == "2"  ,])
       nunres <-  nrow(age3[ age3$ELIG == "3"  ,])
       nexam <-  nrow(age3[ age3$EXAM1 == "1"  ,])
       nquest <-  nrow(age3[ age3$QUEST1 == "1"  ,])
       nexamquest <-  nrow(age3[  age3$EXAMQUEST == "1"  ,])
       pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
       pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
       pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
       copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
       contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )
       if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2 & tab$AGE=="45-54",]) != 0 ) {
          tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2
& tab$AGE=="45-54", "N_NONMISSING"] <- nall
        }
        else {
            tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX", "AGE")] <-
c(country, survey, sex2, "45-54")
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="45-54", "N_NONMISSING"] <- nall
        }
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="45-54", "N_ELIGIBLE"] <- as.integer(nelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="45-54", "N_NONELIGIBLE"] <- as.integer(nnonelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="45-54", "N_UNRESOLVED"] <- as.integer(nunres)
         tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2
& tab$AGE=="45-54", "N_EXAM1"] <- as.integer(nexam)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="45-54", "N_QUEST1"] <- as.integer(nquest)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="45-54", "N_EXAMQUEST"] <- as.integer(nexamquest )
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="45-54", "PR_EXAM"] <- pr_exam
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="45-54", "PR_QUEST"] <- pr_quest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="45-54", "PR_EXAMQUEST"] <- pr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="45-54", "COOPERATE"] <- copr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="45-54", "CONTACT_RATE"] <- contact
```

```
        #### 55-64 years
       nall <- nrow(age4)
       nelig <-  nrow(age4[ age4$ELIG == "1"  ,])
       nnonelig <-  nrow(age4[ age4$ELIG == "2"  ,])
       nunres <-  nrow(age4[ age4$ELIG == "3"  ,])
       nexam <-  nrow(age4[  age4$EXAM1 == "1"  ,])
       nquest <-  nrow(age4[ age4$QUEST1 == "1"  ,])
       nexamquest <-  nrow(age4[ age4$EXAMQUEST == "1" ,])
       pr_exam <-  round( 100*(nexam / (nelig+nunres)), digits=1 )
       pr_quest <- round( 100*(nquest / (nelig+nunres)), digits=1 )
       pr_examquest <- round( 100*(nexamquest / (nelig+nunres)),
digits=1 )
       copr_examquest <-  round( 100*(nexamquest / (nelig)), digits=1 )
       contact <-  round( 100*(nelig / (nelig+nunres)), digits=1 )
       if ( nrow(tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2 & tab$AGE=="55-64",]) != 0 ) {
          tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2
& tab$AGE=="55-64", "N_NONMISSING"] <- nall
        }
        else {
            tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "SEX", "AGE")] <-
c(country, survey, sex2, "55-64")
            tab[tab$COUNTRY==country & tab$SURVEY==survey &
tab$SEX==sex2  & tab$AGE=="55-64", "N_NONMISSING"] <- nall
        }
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="55-64", "N_ELIGIBLE"] <- as.integer(nelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="55-64", "N_NONELIGIBLE"] <- as.integer(nnonelig)
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="55-64", "N_UNRESOLVED"] <- as.integer(nunres)
         tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2
& tab$AGE=="55-64", "N_EXAM1"] <- as.integer(nexam)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="55-64", "N_QUEST1"] <- as.integer(nquest)
        tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="55-64", "N_EXAMQUEST"] <- as.integer(nexamquest )
        tab[tab$COUNTRY==country & tab$SURVEY==survey & tab$SEX==sex2  &
tab$AGE=="55-64", "PR_EXAM"] <- pr_exam
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="55-64", "PR_QUEST"] <- pr_quest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="55-64", "PR_EXAMQUEST"] <- pr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="55-64", "COOPERATE"] <- copr_examquest
       tab[tab$COUNTRY==country & tab$SURVEY==survey  & tab$SEX==sex2 &
tab$AGE=="55-64", "CONTACT_RATE"] <- contact
       selected1=NULL
       age1=NULL
       age2=NULL
       age3=NULL
       age4=NULL
   }
    return(tab)
```

```
}
```

drawtable_sexage - to calculate sex distribution of participants and non-participants by 10-year age-group

```
drawtable_sexage <- function(data)  {
    #### data = dataset
    # tab, output table
    tab <- NULL
    countries <- sort(unique(data$COUNTRY))
    for (country in countries) {
        surveys <- sort(unique(data[data$COUNTRY == country,]$SURVEY))
        for (survey in surveys) {
            selected <- data[data$COUNTRY == country & data$SURVEY ==
survey & data$AGE >= 25 & data$AGE < 65 & data$SEX %in% c("1", "2"),]
            #### 25-64 years
           for (participate in c("yes", "no")) {
               #### 25-64 years participants
               if (participate == "yes") {
                  selected1 <-    selected[selected$EXAM1 == "1" |
selected$QUEST1 == "1" ,]
               }
               else {
                   selected1 <-    selected[selected$EXAM1 == "2" &
selected$QUEST1 == "2" ,]
               }
                nmen <-  nrow(selected1[selected1$SEX == "1"  ,])
                nwomen <-  nrow(selected1[selected1$SEX == "2"  ,])
              if (is.null(tab)) {
                  tab <- as.data.frame(cbind(country, survey,
participate, "25-64", "men"), stringsAsFactors  = F )
                  tab[,dim(tab)[2]+1] <- format(round(100 * (nmen/
(nmen+nwomen)), 1), nsmall=1)
                  tab[,dim(tab)[2]+1] <- nmen
                  names(tab) <- c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")
                  tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country,
survey,  participate, "25-64", "women", format(round(100 * (nwomen/
(nmen+nwomen)), 1), nsmall=1), nwomen)
              }
              else {
                  tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country, survey,
participate,  "25-64", "men",  format(round(100 * (nmen/ (nmen+nwomen)),
1), nsmall=1), nmen)
                  tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country,
survey, participate,  "25-64", "women",  format(round(100 * (nwomen/
(nmen+nwomen)), 1), nsmall=1), nwomen)
              }

               age1 <-  selected1[selected1$SEX %in% c("1", "2")  &
selected1$AGE >= 25 & selected1$AGE < 35 , ]
               age2 <-  selected1[selected1$SEX %in% c("1", "2")  &
```

```
selected1$AGE >= 35 & selected1$AGE < 45 , ]
               age3 <-  selected1[selected1$SEX %in% c("1", "2")  &
selected1$AGE >= 45 & selected1$AGE < 55 , ]
               age4 <-  selected1[selected1$SEX %in% c("1", "2")  &
selected1$AGE >= 55 & selected1$AGE < 65 , ]

               #### 25-34 years
                nmen <-  nrow(age1[age1$SEX == "1"  ,])
                nwomen <-  nrow(age1[age1$SEX == "2"  ,])
               tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country, survey,
participate, "25-34", "men",  format(round(100 * (nmen/ (nmen+nwomen)),
1), nsmall=1), nmen)
               tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country,
survey, participate, "25-34", "women",  format(round(100 * (nwomen/
(nmen+nwomen)), 1), nsmall=1), nwomen)

               #### 35-44 years
               nmen <-  nrow(age2[age2$SEX == "1"  ,])
               nwomen <-  nrow(age2[age2$SEX == "2"  ,])

               tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country, survey,
participate, "35-44", "men",  format(round(100 * (nmen/ (nmen+nwomen)),
1), nsmall=1), nmen)
               tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country,
survey, participate, "35-44", "women",  format(round(100 * (nwomen/
(nmen+nwomen)), 1), nsmall=1), nwomen)

               #### 45-54 years
               nmen <-  nrow(age3[age3$SEX == "1"  ,])
               nwomen <-  nrow(age3[age3$SEX == "2"  ,])

               tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country, survey,
participate, "45-54", "men",  format(round(100 * (nmen/ (nmen+nwomen)),
1), nsmall=1), nmen)
               tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country,
survey, participate, "45-54", "women",  format(round(100 * (nwomen/
(nmen+nwomen)), 1), nsmall=1), nwomen)

               #### 55-64 years
                nmen <-  nrow(age4[age4$SEX == "1"  ,])
               nwomen <-  nrow(age4[age4$SEX == "2"  ,])

               tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country, survey,
participate, "55-64", "men",  format(round(100 * (nmen/ (nmen+nwomen)),
1), nsmall=1), nmen)
               tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country,
survey, participate, "55-64", "women",  format(round(100 * (nwomen/
```

```
(nmen+nwomen)), 1), nsmall=1), nwomen)
           }
        }
    }

    ##### TOTAL LINE
    country="ALL"
    survey="ALL"
    selected <- data[data$AGE >= 25 & data$AGE < 65 & data$SEX %in%
c("1", "2"),]
    #### 25-64 years
   for (participate in c("yes", "no")) {
       #### 25-64 years participants
       if (participate == "yes") {
          selected1 <-    selected[selected$EXAM1 == "1" |
selected$QUEST1 == "1" ,]
       }
       else {
           selected1 <-    selected[selected$EXAM1 == "2" &
selected$QUEST1 == "2" ,]
       }
        nmen <-  nrow(selected1[selected1$SEX == "1"  ,])
        nwomen <-  nrow(selected1[selected1$SEX == "2"  ,])

      if (is.null(tab)) {
          tab <- as.data.frame(cbind(country, survey, participate, "25-
64", "men"), stringsAsFactors  = F )
          tab[,dim(tab)[2]+1] <- format(round(100 * (nmen/
(nmen+nwomen)), 1), nsmall=1)
          tab[,dim(tab)[2]+1] <- nmen
          names(tab) <- c("COUNTRY", "SURVEY", "PARTICIPATED","AGE",
"SEX", "PROPORTION", "N")
          tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country,
survey,  participate, "25-64", "women", format(round(100 * (nwomen/
(nmen+nwomen)), 1), nsmall=1), nwomen)
      }
      else {
          tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country, survey,
participate,  "25-64", "men",  format(round(100 * (nmen/ (nmen+nwomen)),
1), nsmall=1), nmen)
          tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY",
"PARTICIPATED","AGE", "SEX", "PROPORTION", "N")] <- c(country,
survey, participate,  "25-64", "women",  format(round(100 * (nwomen/
(nmen+nwomen)), 1), nsmall=1), nwomen)
      }
       age1 <-  selected1[selected1$SEX %in% c("1", "2")  &
selected1$AGE >= 25 & selected1$AGE < 35 , ]
       age2 <-  selected1[selected1$SEX %in% c("1", "2")  &
selected1$AGE >= 35 & selected1$AGE < 45 , ]
       age3 <-  selected1[selected1$SEX %in% c("1", "2")  &
selected1$AGE >= 45 & selected1$AGE < 55 , ]
       age4 <-  selected1[selected1$SEX %in% c("1", "2")  &
selected1$AGE >= 55 & selected1$AGE < 65 , ]
```

```
        #### 25-34 years
         nmen <-  nrow(age1[age1$SEX == "1"  ,])
         nwomen <-  nrow(age1[age1$SEX == "2"  ,])
        tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "PARTICIPATED","AGE",
"SEX", "PROPORTION", "N")] <- c(country, survey, participate, "25-34",
"men",  format(round(100 * (nmen/ (nmen+nwomen)), 1), nsmall=1), nmen)
        tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "PARTICIPATED","AGE",
"SEX", "PROPORTION", "N")] <- c(country, survey, participate, "25-34",
"women",  format(round(100 * (nwomen/ (nmen+nwomen)), 1), nsmall=1),
nwomen)

        #### 35-44 years
        nmen <-  nrow(age2[age2$SEX == "1"  ,])
        nwomen <-  nrow(age2[age2$SEX == "2"  ,])
        tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "PARTICIPATED","AGE",
"SEX", "PROPORTION", "N")] <- c(country, survey, participate, "35-44",
"men",  format(round(100 * (nmen/ (nmen+nwomen)), 1), nsmall=1), nmen)
        tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "PARTICIPATED","AGE",
"SEX", "PROPORTION", "N")] <- c(country, survey, participate, "35-44",
"women",  format(round(100 * (nwomen/ (nmen+nwomen)), 1), nsmall=1),
nwomen)

        #### 45-54 years
        nmen <-  nrow(age3[age3$SEX == "1"  ,])
        nwomen <-  nrow(age3[age3$SEX == "2"  ,])
        tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "PARTICIPATED","AGE",
"SEX", "PROPORTION", "N")] <- c(country, survey, participate, "45-54",
"men",  format(round(100 * (nmen/ (nmen+nwomen)), 1), nsmall=1), nmen)
        tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "PARTICIPATED","AGE",
"SEX", "PROPORTION", "N")] <- c(country, survey, participate, "45-54",
"women",  format(round(100 * (nwomen/ (nmen+nwomen)), 1), nsmall=1),
nwomen)

        #### 55-64 years
         nmen <-  nrow(age4[age4$SEX == "1"  ,])
        nwomen <-  nrow(age4[age4$SEX == "2"  ,])
        tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "PARTICIPATED","AGE",
"SEX", "PROPORTION", "N")] <- c(country, survey, participate, "55-64",
"men",  format(round(100 * (nmen/ (nmen+nwomen)), 1), nsmall=1), nmen)
        tab[dim(tab)[1]+1 , c("COUNTRY", "SURVEY", "PARTICIPATED","AGE",
"SEX", "PROPORTION", "N")] <- c(country, survey, participate, "55-64",
"women",  format(round(100 * (nwomen/ (nmen+nwomen)), 1), nsmall=1),
nwomen)
    }

    return(tab)
}
```

Appendix 5e. Derivation of equivalised disposable income classes

Kasimir Kaliva[1] and Kari Kuulasmaa[1]

[1]National Institute for Health and Welfare (THL), Helsinki, Finland

Household's equivalised disposable income is a measure of household income that takes account of the differences in a household's size and composition. EHES uses a definition of EUROSTAT: "Equivalised disposable income is the total income of a household, after tax and other deductions, that is available for spending or saving, divided by the number of household members converted into equalised adults; household members are equalised or made equivalent by weighting each according to their age, using the so-called modified OECD equivalence scale." (Eurostat).

The modified OECD equivalence scale (Hagernaas et al. 1994) gives the following weigths for the members of a household: a) 1.0 to the first adult, b) 0.5 to the second adult and each subsequent person aged 14 and over and c) 0.3 to each child aged under 14. Information about children under 14 is not available in the questionnaire used in the EHES Pilot Project, but it has been added to the final EHES questionnaire. The analysis of the EHES Pilot Project data uses a measure for equivalised disposable income where the total income is divided by the square root of the household's size (OECD 2008).

Imputation of household income

The EHES questionnaire asks the household income as a classified variable. Thus, we have only information about the interval in which a household's income belongs. In order to calculate equivalised disposable income we have to impute values of the household income. We will use a proxy for the median of income class in the imputation. A simple method would be to use the middle point of the income class. A problem of this method is what value to use for the lowest and the highest class. A second problem of this method is that the middle point is not a good proxy for median of a class when the income distribution is skewed within class.

Instead of using the middlepoints of income classes, we use a rough approximation of the household income distribution. For the lowest and the highest classes we apply a log-normal approximation. For the intermediate classes we apply a log-linear approximation. Logarithmic transformation of household's income takes into account the fact that the distribution is usually positively skewed, and it eliminates the possibility of a negative income in the lowest class.

For the lowest and the highest income classes we estimate the lognormal distribution, $\ln(TOTAL_{INCOME}) \sim N(\mu, \sigma^2)$, where *TOTAL_INCOME* denotes the household's income. The parameters of this distribution are determined such that the cumulative distribution function equals the empirical distribution function in the limit points of the lowest and highest classes. When the lowest and the highest classes are not empty this condition determines the log-normal distribution uniquely. More precisely, we denote the proportion of observations in the lowest income class by p_l, and the proportion of observations in the highest income class by p_h . The logarithms of the limit points of these classes are denoted by x_l and x_h . Then parameters μ and σ^2 are determined by conditions

$$\frac{x_i - \mu}{\sigma} = \Phi^{-1}(p_i)$$

and

$$\frac{x_h - \mu}{\sigma} = \Phi^{-1}(1 - p_h)$$

where Φ is the standard normal cumulative distribution function. We get

$$\sigma = \frac{x_h - x_t}{\Phi^{-1}(1 - p_h) - \Phi^{-1}(p_1)}$$

and

$$\mu = x_i - \Phi^{-1}(p_t) \times \sigma.$$

For the lowest and the highest class, the imputed household income are medians m_1 and highest class m_h determined by the equations

$$m_i = exp\left[\mu + \sigma \times \Phi^{-1}\left(\frac{P_t}{2}\right)\right]$$

and

$$m_h = exp\left[\mu + \sigma \times \Phi^{-1}\left(1 - \frac{p_h}{2}\right)\right].$$

For the intermediate classes, the density function of the logarithmic total income is approximated by a linear function: $a_i + b_i \ln(TOTAL\_INCOME)$, where coefficients a_i and b_i are chosen such that the integral over the income class *i* equals the proportion of observations p_i in this class. Coefficient b_i is determined by the frequencies of the neighbor classes. Firstly, we choose two points: l_i from the income class *i*-1 and u_i from the income class *i*+1. If the class is lowest or highest we use the limit point and otherwise the middle point of the class (i.e. average of logarithmic limit points). Then we calculate density functions in these points. In the case of the lowest and the highest classes the densities are calculate using the above log-normal approximations. Otherwise, we use the average density: frequency of the class divided by the difference between the logarithmic limit point of the class. After that we determine b_i as

$$b_i = \frac{dens(u_i) - dens(l_i)}{u_i - l_i},$$

where dens(x) is the density function at point x.

The median m_i of the intermediate class i is $m_i = exp(y_i)$, where y_i is the solution of the following equation

$$\int_{s_i}^{y_i} a_i + b_i x dx = 0.5 p_i$$

where p_i is proportion of observations in class i and s_i is the logarithmic lower limit of class i. After easy calculus we get

$$m_i = exp\left\{\frac{-a_i + \sqrt{a_i^2 + 2b_i c_i}}{b_i}\right\},$$

where $c_i = 0.5 b_i s_i^2 + a_i s_i + 0.5 p_i$.

Definition of five categories of the equivalised income

For the comparison of health indicators between income categories, the equivalised disposable income will be classified into five distinct categories. Observations which have the same value should belong to the same category. A conventional method is to divide households according to quintiles (5-quantiles). By the definition, the value x_k is the k:th quintile if and only if

$Pr(X < x_k) \leq \frac{k}{5}$ and $\Pr(X \leq x_k) \geq \frac{k}{5}$

where the random variable X follows the empirical distribution function.

A problem now is the classification of the observations which equal the k:th quintile x_k. For such observations we will apply a multistage classification method. In the first stage we classify the observations which do not equal any quintile. The observation which are smaller than the first quintile x_1 are classified to the class 1 and the observations which are larger than x_1 but smaller than x_2 are classifying to class 2 etc.

In the second stage of the classification, the observations which equal the *k*:th quintile x_k are classified to class *k* if

$$\left|\Pr(X < x_k) - \frac{k}{5}\right| < \left|\Pr(X \leq x_k) - \frac{k}{5}\right|$$

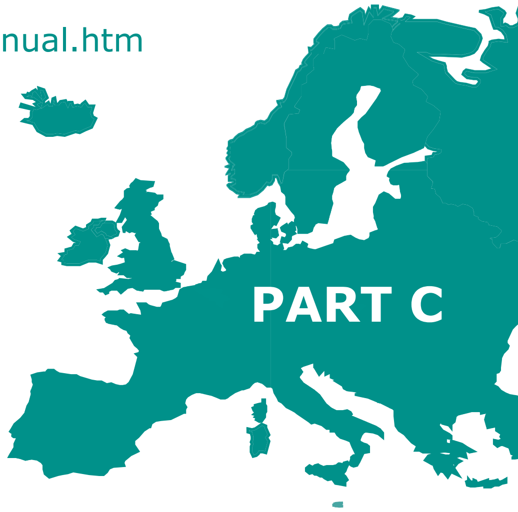
and to the class *k+1* if

$$\left|\Pr(X < x_k) - \frac{k}{5}\right| > \left|\Pr(X \leq x_k) - \frac{k}{5}\right|.$$

In the third stage, we classify the remaining observations (i.e. those where the above differences are equal). We put the limit observations to the class which is smaller before the classification of these observations. In this stage the observations which equal the first quatile x_1 are classified first, then the observations which equal the second quintile x_2. If classes have equal size we put limit observations to the class which is further away from the median class.

References

- Eurostat modified OECD scale, available at http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Glossary:Equivalised_disposable_income
- Hagenaars, A., K. de Vos & M.A.Zaidi (1994): *Poverty statistics in the late 1980s: research based on micro-data,* Office for official publications of the European communities. Luxemburg.
- OECD (2008): *Growing unequal? Income distribution and poverty in OECD countries,* Paris (http://www.oecd-ilibrary.org/social-issues-migration-health/growing-unequal_9789264044197-en)



6. Organization and European level coordination of EHES

Kari Kuulasmaa[1]

[1]National Institute for Health and Welfare (THL), Finland

The Feasibility of the European Health Examination Survey (FEHES) study made recommendations for Organizing EHES (Tolonen et al. 2008). The key players of the organization, which has been strongly supported by the experience from the Pilot study, are countries, EHES Reference Centre, research community and European Union.

6.1 Countries

The responsibility of the countries is to:

- *Carry out the national surveys.* The responsibility of planning and conducting the national HESs is at the national level. This increases local motivation for obtaining high quality results, possibly increases the participation rate and is important for the selection of those measurements that are most important nationally. The national infrastructure and other national aspects such as habits, public and professional attitudes and health information needs can be taken into account. This approach also facilitates the capacity building in the countries and training of national experts for the proper analysis and interpretation of the survey results.
- *Fund the national survey.* A partial EU contribution to the costs would lower the threshold for national funding in the countries. Furthermore, it would help to ensure that each survey meets the minimum size and measurement requirements, and that the individual-level data can be transferred for joint analysis.
- *Analyze the national data and disseminate results.* The purpose of the survey is to provide processed information widely for health policy planners and decision makers, health professionals, the general public, and researchers. The countries have a key role in this in the national level.

6.2 EHES Reference Centre

The responsibility of the EHES Reference Centre is to organize the necessary European level collaboration. This involves:

- *Coordinating the EHES network* of national survey organizers, and European level and international stake holders.
- *Creating, maintaining and disseminating the EHES Manual* on the EHES guidelines and measurement standards.
- *Providing advice to the national survey organizers* in planning and preparation of the national surveys, such as preparation of national manuals, sampling, ethical issues and data management.
- *Organizing training seminars and preparation of training material.*
- *External quality assessment (EQA)*, including evaluation of national survey plans, site visits during the pilot and full-size surveys and laboratory quality assessment.
- *Collecting data from the national HESs* for survey evaluation and joint analysis and reporting. This facilitates rapid assessment of the quality of the data from each country and European level reporting and dissemination.
- *Evaluation of each national HES and EHES as a whole.* The experience has to be documented for future HESs. The quality and country specific characteristics of each HES have to be documented so that those who analyze the data can take these into account in their analysis and interpretation of the results.
- Undertaking *rapid basic reporting* and interpretation of the results for use at the European level; and
- Maintaining the EHES database and sharing analysis data sets with research groups. There is a desire to encourage the use of the data widely for public health benefit while maintaining the legitimate interests of the countries which collected the data. Research collaboration should also help to increase the capacity of all countries to analyse the data on the different health aspects covered by EHES.

(During the EHES Pilot Project in 2009-2011, the EHES Reference Centre was established jointly by the national institutes of Finland and Italy and Statistics Norway with financial support from the European Union. This funding ended in April 2012, and new arrangements will be needed for the continuation of the EHES Reference Centre Activities.)

6.3 Research community

The research community, nationally and internationally, has the responsibility to convert the EHES data into useful interpretable information for the benefit of public health. Another task of the research community is to evaluate and develop survey methods to even better meet the future needs.

6.4 Commission of the European Union

The European Commission is not directly involved in the activities of EHES. However, being a major European level stakeholder, it has shown interest in:

- Facilitating and overseeing EHES, to ensure that it meets its European level and national objectives, and operates in harmony with WHO, OECD and other international agencies;
- Funding the European level activities, such as the EHES Reference Centre, and meetings for countries to share their experiences and plan for the future.
- Providing partial funding for the countries carrying out surveys. The EHES Pilot Project showed that this makes the national fund raising much easier. It also helps to ensure that each survey meets the minimum size and measurement requirements, and that the individual-level data can be transferred for joint assessment and reporting.

References

- Tolonen H, Koponen P, Aromaa A, Conti S, Graff-Iversen S, Grøtvedt L, Heldal J, Kanieff M, Mindell J, Natunen S, Primatesta P, Verschuren M, Viet L, Kuulasmaa K for the Feasibility of a European Health Examination Survey (FEHES) Project. Recommendations for Organizing a Standardized European Health Examination Survey. B22/2008. Helsinki, Publications of the National Public Health Institute, 2008. Available at http://urn.fi/URN:ISBN:978-951-740-840-0.

ISBN (pdf) 978-952-245-846-9

ISSN (pdf) 2323-4172

http://urn.fi/URN:ISBN:978-952-245-846-9

http://www.ehes.info

Reference:

Tolonen H (Ed.) EHES Manual. Part C. European level collaboratoin.
National Institute for Health and Welfare. 2013. Directions 2013_003.
URN:ISBN:978-952-245-846-9,
URL: http://urn.fi/URN:ISBN:978-952-245-846-9