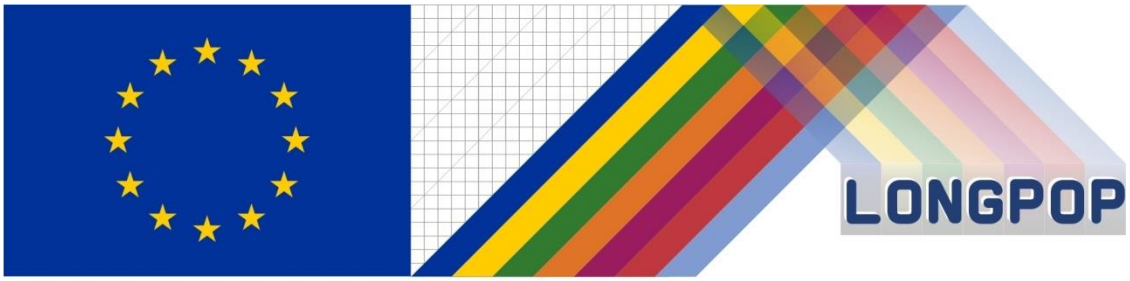

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 676060

LONGPOP

Methodologies and Data mining techniques for the analysis of Big Data based on Longitudinal Population and Epidemiological Registers

Report on the use of the IDS and Extraction Software

Deliverable 3.1.

Contents

Acronyms used in this document

CEDAR: Centre for Demographic and Ageing Research (University of Umeå)

ESR: Early-Stage Research

GIS: Geographic Information System

IDS: Intermediate Data Structure

IECA: Institute of Statistics and Cartography of Andalusia

IISG-KNAW: International Institute of Social History of Amsterdam (IISG) as part of the Royal Dutch Academy of Sciences (KNAW).

IRP: Individual Research Project

KU Leuven: Catholic University of Leuven

SQL: Structure Query Language

STATA: Statistics software packages by StataCorp.

TELNET: Redes Inteligentes (inTELligent NETworks)

WP3: Work Package 3

1. Introduction

Over the past decades the field of historical demography has greatly expanded partly due to the availability of digitized historical longitudinal micro-level databases. These types of databases allowed researchers to improve their understanding on fields like mortality, fertility, social stratification and mobility, the long-term impacts of early life conditions, and other demographic matters.

The use of longitudinal micro-level historical demographic data presents many challenges connected to their multilevel relations. In order to make good use of the possibilities of analysis of this type of data, they have to be set up in a specific format: the so called rectangular episodes files. But the problem is that the creation of such files requires advanced data management skills, and the lack of programs to do so restricts the usability of such data limiting the scope of historical demographic research.

The Intermediate Data Structure (IDS) was developed as a strategy aimed at simplifying the collecting, storing and sharing of historical demographic data (Alter & Mandemakers 2014; Alter, Mandemakers & Gutmann 2009). It is a common format that allows to manage and compare data from different databases, regardless of their original structure. IDS structures longitudinal databases and overcomes the problems of comparability which are inherent on large historical datasets on populations by providing a common dissemination format. In order to analyse the data contained in this intermediate structure, extraction software is developed to select the information from the IDS tables and to convert it into datasets ready for analysis. In this extraction process data additional variables are constructed from the IDS files and converted into a rectangular episodes table. All algorithms and programming code to achieve this is the so-called extraction software.

The solutions and programs created this way can be applied to historical demographic databases created from population registers or family reconstitutions.

This report gives an overview of the use of these tools (IDS and extraction software) by the ESRs of the LONGPOP project who had to apply these common methodologies to their research studies and deliverables. The extraction software is and will become available by way of the following webpages: LONGPOP (http://longpop-itn.eu/) and EHPS-network (https://ehps-net.eu/).

This report also collects the implementation of the algorithms involved in the construction of extraction software or the IDS format database. In addition, data mining techniques involved in these processes, as well as data linkage, are taken into account.

Data linkage is a method of bringing information from different sources together, mostly about the same person, but also about the same entity, to create a new, richer dataset. Linking information from different sources contributes to the construction of episodes -longitudinal sequences of events of those persons or entities- that may help to reconstruct, for example, life courses.

In addition, data mining or data exploration is a field of statistics and computer science referring to the process that attempts to discover patterns in large volumes of data sets, using the methods of artificial intelligence, machine learning, statistics and systems of data bases.

This report is organized by research institution, starting from the Radboud University of Nijmegen, Telnet, The Catholic University of Louvain (KU Leuven) and the International Institute of Social History of Amsterdam (IISG) as part of the Royal Dutch Academy of Sciences (KNAW).

2. ESRs' Developments

Radboud University Nijmegen, The Netherlands

ESR 4.- Dolores Sesma.

The substantive topic in this IRP is to find ways to integrate individual residential histories in the study of vulnerability and well-being; to extract migration and household trajectories from micro-data sets; to develop definitions and typologies to analyze migration trajectories and sequences of households over the life course; and to convert data from datasets on addresses and households to files ready for statistical processing.

The main points concerning WP3 are:

1. Concepts. Definitions and typologies of residential mobility over the life course that can be used comparatively.
2. Algorithms. Algorithms to extract mobility data from IDS converted databases
3. Statistical tool. A tool to explore this mobility statistically, e.g. based on sequence analysis.
4. GIS mobility tool. A tool to visualize trajectories and extended kin networks.
5. Analytical tool. A tool to explore the links between migration trajectories and later-life outcomes in terms of social status and well-being.

Owing to this, creating analysis files requires data management and programming skills that not all researchers possess. This is especially true for the study of migration since that information is mostly not readily available / explicitly stored in a longitudinal data set.

ESR 4, in collaboration with Thijs Hermsen have been working in a tool concerning migration. The *Migration Extraction Tool* (*MET*) is developed to help researchers extract data on migration stored in historical longitudinal data sets and create a migration episode file. The *MET* is specifically developed to use data from the Intermediate Data Structure.

A crucial part in the process of creating the Migration Episode File are the techniques in the isdr-package that we applied for the *MET*. This package was developed by Göran Broström (Broström 2017). The isdr-package is a modified variant for R of the Episode File Creator in STATA (Quaranta 2016 and 2015) that creates an episode file for statistical analysis with the help of two tables, the Chronicle data frame with variant data and the Personal data frame with invariant data.

The output of the *MET* is an episode table for each Research Person records, ranging from a few to many per individual. Each record is a spell, that starts with *enter* and stops with *exit* in which variant data is constant. When the event (in our case a migration) takes place, this happens at the end of a spell and it is marked in the episode table. More on this in paragraph 6.

Overview of the Migration Extraction Tool

The first version of the *MET* focusses on the Research Persons in a IDS dataset and filters out the movements of person within a municipality. Migrations because of municipal changes are not filtered out.

The *MET* is split up into 5 parts, each with one or more scripts that go through one or more steps when executed.

Each part and each script build on the outcome(s) of the previous one, and the user has only a few choices to make in the first. Of course, it is also possible to use the lines of codes without the functions. In most steps, a new data frame is created. This way, users can trace back what the results were, for a particular action.

The parts are:

0. Preparation: Four steps to do manually before the first script can be executed.
1. Extract migration data: in this step a script will create a data frame with the migrations from one municipality to another for all Research Persons in the IDS set.
2. Build Personal data frame: using the data frames created in step 1 and the INDIVIDUAL data the Personal data frame is build.
3. Build Chronicle data frame: using the data frames created in step 1 and the INDIVIDUAL data the Chronicle data frame is build.
4. Create Migration Episode File: with the help of the techniques in Broström's isdr-package the Migration Episode File (MEF) is created by first transforming both data frames and then joining them together.

TELNET, Spain

ESR 8.- Vasiliis Giagloglou.

One of the aims of this ESR is the participation on the design and development of an application that makes use of historical and present data stored in non-homogeneous, non-structured databases using technics of Big Data processing and statistical analysis to offer researchers appropriate tools to analyze longitudinal data. Therefore, some of the tasks that will be included in this IRP are: to compile the requirements regarding the information to show in the application and the databases to get the information from, with the definition of the queries, relationships between them, intermediate and common data structures needed and data conversion algorithms. The design and development of algorithms for extracting information from the different databases selected (if viable) and the translation to an intermediate data structures for data analysis. Also, the specific points within WP3 are the following expected results:

a. Catalogue with the final information to show in the application and the databases used to obtain information, the definition of the queries, the relationships between them, necessary intermediate and common data structures and data conversion algorithms needed to implement.
b. Report on the conversion techniques needed for each database used in the project.

Even though ESR 8 enrolled the project in the middle of the whole-time period, he is working on both points, and on the concept of *Elasticsearch*, an example of data mining software as a powerful search engine, with the possibility to visualise and select valuable data from IDS databases without the necessary programming skills. His work is a demonstration of the *Elasticsearch* utility in the domain of the common format IDS.

The HSN is a representative sample of about 85,000 people born in the Netherlands in the period of 1812 - 1922. The data of the HSN-database contain individual life courses and it constitutes a unique tool for research in Dutch history and demography.

Making use of the HSN_IDS (the IDS version of the Historical Sample of Netherlands), a preliminary work has been made for the implementation of the tools as described below. Future work will include extraction software based on R programming and visualisation of features based on R.

As repeatedly stated, IDS makes easy demographic, historic and sociologic research based on individual-level data. However, issues of anonymity add difficulty for sharing the information by these databases and thus limiting the possibility for a full-scale research. *Elasticsearch* provides a safe distributed solution for analysis to all the IDS databases and the capacity of extracting data, either with *Kibana* -the native web interface of *Elasticsearch*- or even with the programming language R. With it, all the data can be shared to trusted parties in a safe network of IDS interested scientists, to be compared, extracted, analysed in international scales. All the IDS databases can be available, in an easy to use web client, from all the interested scientists.

Materials and Methods.

Initially a preliminary format adjustment has been made to the IDS version of the HSN, HSN_IDS. To acquire the proper format for further use, specific software was used. *PowerBI* and *DAX* Studio were employed for doing a preliminary cleaning of the dataset, like removing empty columns and extracting the information into a proper csv format suitable for *Elasticsearch*. In addition, they were used for extracting samples of the database into csv.

For installing data into *Elasticsearch* it was decided to use the whole *ELK* stack, which are the acronyms of *Elasticsearch*, *Logstash*, *Kibana* and another open source project *Filebeat*.

- *Filebeat* is the data shipper into the *ELK* stack.
- *Logstash* is a server-side data processing pipeline that ingests data from multiple sources simultaneously, transforms them, and then sends them to a "stash" like *Elasticsearch*.
- *Elasticsearch* is the search and analytics engine.
- *Kibana* lets users visualize data with charts and graphs in *Elasticsearch*.

Results

After finishing the developments, ESR 8's work will present the potentiality of *Elasticsearch* and its counterpart *Kibana* for the use in terms of visualisation and feature extraction of longitudinal IDS format databases. A preliminary work has been made for the implementation of the tools as described above. The following steps to come include extraction software and visualisation of features, all based on R programming.

Although *Elasticsearch* is a useful tool, it doesn't provide cross-index queries without the same mapping, although it may be in progress for future releases of the tool.

IDS is the best basis for creating common mappings inside *Elasticsearch*. And combining with the transformation of more databases in IDS format, we can inject all the data inside *Elasticsearch* for an international scale research solution.

KU Leuven, Belgium

ESR 7.- Sam Jenkinson.

The field and main goal of this IRP is to link large numbers of individual time segments from large samples or populations to those of relevant networks, capturing the situation for particular periods and change over time means accounting for intra- and intergenerational connections.

ESR 7 is engaged with Family and Population Studies (FaPOS) department in KU Leuven, focusing on family research from longitudinal, intra-intergenerational and comparative perspective. Also, a highly valued database on 19th century Antwerp (COR-database) has been created, consisting of

intra- and intergenerational demographics, sociological information recorded at micro, meso and macro levels. This database allows one to link three generations of longitudinal life course trajectories on key demographic events. Moreover, additional cross-national IDS datasets from the European Historical Population Samples Network (EHPSN) can be incorporated.

ESR 7's main goals with respect to WP3 are:

1. Concepts and techniques for individual record linkage. Algorithms and relevant syntaxes to construct intergenerational life course trajectories from original or IDS-converted multi-level and multi-source databases.
2. Methodologies and techniques for the longitudinal analysis: harmonization and transformation of historical data sets
3. Methodologies and techniques for the longitudinal analysis: harmonization and transformation of historical data sets

ESR 7 Report.

The *COR technical note for the construction of intermediate data structure (IDS)* was developed by Sam Jenkinson, Hideko Matsuo & Koen Matthijs (KU Leuven), and it concerns the work deliverable of 1. regarding concepts and techniques for individual record linkage providing algorithms and relevant syntaxes to construct intergenerational life course trajectories from original or IDS-converted multi-level and multi-source data bases. The ultimate goal of this exercise was to produce new COR-IDS, preparing inputs for the construction of intermediate data structure (IDS), serving for the research output of 1. The construction of COR-IDS takes full account of the objective of the LONGPOP project aiming, to create long-term, longitudinal, multi-source and intergenerational data sets. The original input of the COR multi-level, multi-source data file authors used here is based on the 2010 version of the COR sample (Matthijs and Moreels 2010). They envisage at least three generations of families (i.e. child, parents and grandparents) will be present in this COR-IDS data and will be available for the wider public use of COR-IDS.

Concerning the data, the Antwerp COR*(2010) database is a historical demographic database, constructed using a letter sample and with a total sample size of +/-33,000 individuals. It spans nearly seven decades (1846 to 1920) and consists of information drawn from the population registers and the vital registration records (i.e. birth, marriage, and death) of the whole district of Antwerp (Flanders, Belgium). Every person whose family name starts with the letter combination COR* is selected in the database. The database covers three linked generations and contains micro-data on the individual level demographic events, partnership and family formation, household composition including biological and non-biological relationships, migration and death.

This note, as well as the construction of the IDS database format, the construction of algorithms and the process of record linkage are shown in detail in the Appendix A, and it consists of the following information based on the process documentation since March 2017:

1. Standardization of core variables for the input of individual and context attribute tables;
2. Standardization and evaluation of individual linkages COR-2010;
3. Specification of standardized variables in meta data format;
4. Individual attributes on core variables from output 1 (indiv data file); and
5. State of art on the population register and how to proceed with indiv-indiv data.

KNAW – IISG, The Netherlands

ESR 2.- Francisco Anguita

This ESR has been working on both, the testing and replicating of extraction software and developing new code. On the one hand, two examples of extraction software are being tested prior to their upload to the EHPS-network. On the other, four small pieces of extraction software are being developed to encapsulate certain features of the IDS records. Finally, seven STATA programs developed by Luciana Quaranta for the use of the IDS (Quaranta, 2016) are being replicated into R. These are the R versions of Quaranta's extraction software.

Extraction Software of Occupations

The first one is linked to the Extraction Software for Occupations from the IDS. This software was developed originally to work in a DB2 database environment using SQL programming language in the University of Umeå, Sweden. In it, occupation at a given date is chosen according to this priority:

a) Occupation is chosen from the record where the date for the event is within the period of the occupational record.

b) Occupation is chosen from the record with the least number of years between event and record.

c) When two occupational records have the same number of years between an event and the record, then the record that is closest before the event is chosen.

The data selected concerns the father's occupation at the birth of the research person, the father's occupation at the research person's marriage, the research person's first and last occupations as well as the highest status position achieved.

The ESR2 replicated this software into R so that it could be generally and free available. This replication tries to overcome the several problems BD2 sql language creates when it is directly translated into R by means of the package "*sqldf*". This package is a way designed by R developers to embed complete queries into R. By calling the *sqldf* function and embedding the query into it, R can potentially run it as in a SQL environment.

Several obstacles were confronted to do so for the Software of Occupations: some commands are not supported by the package. For instance, some join instructions like the *Full Join* wouldn't work, as well as the ranking command (*Rank Over Partition*) gets error messages that prevent the program from continuation.

In order to work around these pitfalls, along with the succession of nested joins, the original code was broken down into several queries encapsulating each *Select* command. This forced the reorganization of the data sets involved and the creation of numerous new variables, and the *Full Join* command was converted into the cartesian product of the correspondent two tables. In addition, the ranking was avoided by means of the *transform* function of R and employing the *average* command in it (*ave()*).

```
last_rankRanking <- transform (
                    last_rank,
                    Rank = ave (c(Start_year, End_year, Value, Value3, Id, Id3, Value4, Id4),
          Ident,
                    FUN = function(x) rank (-x, ties.method ="min")) )
```

The code could finally be completely replicated, but we're waiting for a data set from the University of Umeå, so that a systematic comparison can be carried out between our outcome and that from the original extraction software.

Idsr Package

The idsr is an R (R Core Team 2017) package, meaning that it is not a stand-alone program, but needs to be running in an R environment. It is the R version of the Episodes File Creator by Luciana Quaranta (Quaranta 2015) in Stata. This version (0.1.1) of idsr is in an early alpha state and under development. Its purpose is to bridge the gap between an IDS database and software for survival analysis, typically the R packages eha (Broström 2017; Broström 2012) and survival (Therneau 2015), and it's been developed at the CEDAR, Umeå University. It transforms the IDS data to a suitable form for survival and event history analysis using R as a dialect of S, and it goes through two main steps: from the IDS database to the creation of the Chronicle and Personal files, and from this to the final data set for the analysis represented by the Episodes file.

The last step is straightforward, but the problem lies on the step from the IDS to the *Chronicle*. A great deal must be done in preparing the *Chronicle* and the *Personal* files as the first step has a slightly different approach with respect to the original development: the *Chronicle* is supposed to contain records of events. These events define in turn changes in levels of variables, that must be defined. This approach also forces us to strip off time-fixed covariates, such as sex, birth place, birth order, etc, from the *Chronicle*. These variables are characterized by the absence of a time-stamp and they are stored in the *Personal* frame. For instance, *NA*s in the column of civil_status should probably be changed to 'single' (not married), but, we don't know from the sources. The function must also be told the names of the start and stop events for the aimed-at analysis.

Also, regarding idsr, one problem is the dependence on the R packages 'dplyr' and 'tidyr'. While these packages are easy to use interactively, they are trickier in packages, as they use 'nonstandard evaluation' (NSE). It should be possible to rewrite everything using 'standard R', and this should be one of the things to address.

Nonetheless, we still work on the new approaches as well as the testing of this version of the *Episodes Files Creator*.

New Modules based upon the householdSize Program

On the light of the extraction software developed by Luciana Quaranta (Quaranta 2015), some other software is being developed. The goal was to encapsulate the original code and to compartmentalize the extraction software based upon some specific features the IDS presents. In this sense, four different algorithms have been developed. On the one hand, extraction software providing information of constant features of the individuals of the IDS. On the other hand, extraction software showing migration, religion and occupation characteristics.

Explanation into details is presented in the Appendix B.

1. Extraction Software for the Constant Features of IDS

The objective of this software is to keep track of only the constant features of the table INDIVIDUAL. They are time invariant and we can find aspects like birth location and date, or those linked to the individual's baptism, death, funeral (again with date and location). We also include other aspects like the name. Concerning the last point, we assume that in some databases (as in the Historical Sample

of the Netherlands) this component need not change, as women kept their last names after marrying (although this would not be applicable to other databases)

2. Extraction Software for migration features

The objective for this software is to collect migration features within the records present in the IDS. As this is not reported (by way of the creation of specific variables) in the IDS tables, the goal is to collect the data that keep track of the changes individuals go through concerning the locations they are registered at. The idea behind was to document the possible changes appearing in the Value_Id_C variable, in the table INDIVIDUAL. This variable gives the code of a certain context that can be traced back using the CONTEXT_CONTEXT and CONTEXT tables.

3. Extraction Software for religion features

The process is as follows:

a) We collect the data linked to the Type variable of the INDIVIDUAL table that may report about religion. Variables like "BAPTISM_LOCATION", "RELIGION", "RELIGION_STANDARD" may be interesting for that purpose.

b) After that, we subset the table Individual for only the records that have some or all of these features.

4. Extraction Software for occupation features.

This software follows the same steps and structure a for the occupations.

Replication into R of the seven STATA Programs for using of the IDS

Seven STATA program are being converted into their R version. These programs have been developed to be used directly by researchers, and they are easy to run. They are developed modularly and, to a large extent, can be used independently of the others.

The open-access program *Extended IDS table maker* can be used to create the EIDS tables INDIVIDUAL_EXT, CONTEXT_EXT as well as a Chronicle file and a Variable Setup file. It creates empty tables, which can be filled in with variables constructed locally or by other extraction programs.

Household size is an example of a program that can be used to construct extended variables at the contextual level. In this case, it computes the number of members of a household along with time.

Using the *program Import data*, variables created by extraction programs or by locally written functions can be inserted into the INDIVIDUAL_EXT and CONTEXT_EXT tables; and information relating to such variables can be added to the METADATA table.

Variables stored in the INDIVIDUAL, INDIVIDUAL_EXT, CONTEXT or CONTEXT_EXT tables can be selected by using the *program Select Type*. The program produces an Excel file which contains the columns Type, Select and Duration and which lists each unique Type stored in the tables.

Selected individual variables can be automatically added to the Chronicle file using the *program Append individual variables*. This program obtains the variables selected by the user from the INDIVIDUAL or the INDIVIDUAL_EXT tables and appends such information to the Chronicle file. It also appends information relating to these variables to the Variable setup file.

The *program Append contextual variables* can be used to transform contextual extended variables selected for analysis into individual extended variables and to append these transformed variables to the Chronicle file.

Finally, the *Episodes file creator* is the program that produces rectangular episodes files. Using input from the Chronicle file and Variable Setup file, this program combines the variables included in such extraction, transforming the extraction into a rectangular table and formatting this file to be ready for statistical analysis.

ESR 11.- Diogo Paiva

The main goal of this ESR is to tackle the problem of vague addresses for households in historical nominal lists, census records or population registers. The way to do so is by means of geo-referencing address to a point on the map in a detailed way in combination with a time stamp. But, along with these tasks, ESR11, has also participated in the implementation of an IDS format for the databases of IECA in Seville, Andalusia.

This involved the development of an R-package following the IDS Transposer strategy presented by Alter et al. (2017) for the restructuring of databases in the IDS format and following an Excel file for the workflow. Not everything has been automated from this Excel file, and for some things it was still necessary to construct (intermediate) input tables, though they're a little more efficient.

Another task in this project was the testing of the abovementioned package along with the Transposer examples to verify that the same result is obtained. Also, it was necessary to test the package with other datasets to check gaps and extend the scope of it.

Finally, they achieved the conversion of two (public) sample datasets of mortality and fertility from IECA in the IDS format, by way of the package.

3. References

Alter, G. & K. Mandemakers (2014). **The Intermediate Data Structure (IDS) for Longitudinal Historical Microdata, version 4.** Historical Life Course Studies 1 (2014), 1-26, published on line 26th of May 2014. PI: http://hdl.handle.net/10622/23526343-2014-0001?locatt=view:master

Bröstrom, G. (2012) **Event history analysis with R**. Boca Ratón, United States: CRC press.

Caranci N, Biggeri A, Grisotto L, Pacelli B, Spadea T, Costa G. **The Italian deprivation index at census block level: definition, description and association with general mortality**. Epidemiol Prev. 2010; 34: 167–176.

Klancher Merchant, E. & Alter, G. (2017). **IDS Transposer: A Users Guide**. Historical Life Course Studies, 4, 59-96. http://hdl.handle.net/10622/23526343-2017-0004?locatt=view:master

Quaranta, L. (2015**). Using the Intermediate Data Structure (IDS) to Construct Files for Statistical Analysis**. Historical Life Course Studies, 2, 86-107. http://hdl.handle.net/10622/23526343-2015-0007?locatt=view:master

Quaranta, L. (2016**). STATA Programs for Using the Intermediate Data Structure (IDS) to Construct Files for Statistical Analysis**. Historical Life Course Studies, 3, 1-19. http://hdl.handle.net/10622/23526343-2016-0001?locatt=view:master

Schumacher, R., Matthijs, K., & Moreels, S. (2013**). Migration and reproduction in an urbanizing context. Family life courses in 19th century Antwerp and Geneva**. Revue Quetelet/Quetelet Journal, 1, 51–72.

Therneau, Terry M. 2015. **A Package for Survival Analysis in S**. https://CRAN.R-project.org/package=survival.

4. Appendix

Appendix A.

This appendix describes the process outputs for the bullet points explained in the section of ESR 7 project in KU-Leuven.

1. *Standardization of core variables for the input of individual and context attribute table*

In order to ensure the accuracy of the COR-2010 inputs, the authors have performed an inconsistency check across different sources using the unique identifier of the COR-2010 sample. They acknowledge that this will be subject to the evaluation of the linkages across different sources, taking into account the newly applied individual identifier (see on 2. standardization and evaluation of individual linkage). They also consider that identifying inconsistencies contributes to the evaluation of linkage too.

Our initial control on core variables (names, gender, event dates, date of birth/death and occupation) based on COR-2010 show the following intra-inconsistencies (see note on 10/3/2017, section 5.1).

When clear differences of information are found, following Mandemakers and Dillon (2010), they give priorities on the sources depending on the types of event and dates they are interested in. For instance, when multiple sources exist for birth dates, preference will be given to birth certificate rather than any other sources such as marriage certificate or death certificate.

2.*Standardization and evaluation of individual linkages (COR-2010)*

The purpose of this exercise is to examine the quality of the linkage (IDNR) of individual records and attributes across different sources in the COR database. Currently the database is stored as one fully merged database, and not as separate source tables, in line with the best practice database methods, as described by Mandemakers and Dillon (2004). Here they outline how the best practice in historical database management is to retain separate original source tables, in a format as close as possible to a digitised version of the original source. The authors seek to bring COR closer into line with these guidelines, as best as possible in light of being a pre-existing database with no original source tables or digitised records of the original source, and to examine the accuracy of previous linkages, as well as replicating and improving the original linkage process, in order to prepare for implementing COR IDS.

In order to assess the linkage quality in COR the authors began by splitting the database into observations from the original historical sources: birth, death, marriage certificate and event (i.e. population register) databases. Then they compared the accuracy of vital information of gender, dates and locations of birth, death and names across IDNR (unique identifiers from previous linkage). This will give them information concerning the size of any errors in linkage currently within the database.

The second step was to relink the database from the original source tables, ignoring the previous identification numbers given during the prior record linkage process. This allowed the authors to evaluate the process of the original linkage and provide the linkage algorithm for others to use in line with deliverable (1). Here initially they use the methods as in line with the initial linkage 2010 (Van Balen). The authors use a stochastic record linkage method provided by Sariyar and Borg (Sariyar & Borg, 2010) as part of their R package “record linkage” (2016).

The record linkage process provided by the package uses the Fellegi-Sunter Model (Sariyar & Borg, 2010), as seen below.

$$\omega_{\tilde{\gamma}} = \log(\frac{P(\gamma = \tilde{\gamma} \mid Z = 1)}{P(\gamma = \tilde{\gamma} \mid Z = 0)})$$

Where the linkage relies on the assumption of conditional probabilities regarding comparison patterns. This works on the probability that a random vector γ=(γ1,…,γn) , having the value $\tilde{\gamma}$=($\tilde{\gamma}$1,…,$\tilde{\gamma}n$) , is conditional on the matching status of Z. Where Z=0 stands for a non-match and Z=1 equals match. In the full Fellegi-Sunter model these are used to compute weights which are used in order to discern matches and non-matches. The weights within the package are computed using an expectation maximum (EM) algorithm in line with Haber (1984) and Contiero et al (2005).

Then common variables are used across observations sources to calculate the likelihood of a record being a match using a string comparison tool. The selected variables here include given and family names, birth location, birth day, birth month and year separately. These are then used to calculate similarities across different records and to create pairs.

The string comparison used here is the Jarrow-Winkler distance string comparison tool (Winkler, W.E 1990). This function works by measuring the edit distance between two strings and calculates the minimum numbers of single character transpositions to transform one word into another.

The original Jaro distance calculation can be seen below;

$$Jaro\ dist = \frac{1}{3}\ (\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m})$$

Where m= the number of common characters which are within half of the length of the longer string and t the number of transpositions. The Jaro-Winkler distance is the same, but with the following changes

$$JaroWinkler\ dist = Jari\ dist + (1 - Jarodist)$$

Where ι represents the common prefix at the start of a string up to a maximum of 4 characters, a standard approach for these purposes.

In addition to this the process using the procedure of blocking (Sariyar & Borg, 2010) is begun to ensure the strictest criteria for record matches, before sequentially relaxing the criteria in order to report the frequency of matches given at a particular matching stringency. Blocking limits the number of matches by ensuring that one particular column or variable is a 100% match. This uses a phonetic tool contained within the package before moving on to assess the individual data contained in other columns using the string comparison tool. This is useful to provide perfect matches but also used to reduce computation time. The process begins with one column which is a unique column based on all data contained in the other columns used for linkage. This column contains a string made of all names, birth dates and birth locations. This will then give us the number of matches which are 100% identical across all comparison fields for birth and death certificates. The authors then slowly remove the number of variables, over three rounds initially here, included in the initial blocking identifier column and report the number of matches and therefore their quality.

The steps are implemented as follows. The first name was split into 5 possible first name variables owing to the likelihood for some to have multiple given names. Initially all 5 of these first names, as well as family name, Birth location in NIS code, birthday, birth month and birth year are included in the blocking criteria. In the second round only the first three given names, surnames, birth places

and birth dates are included and in the final round the blocking criteria included only the full birth dates (day, month, year) and two first names and family names.

During this linkage procedure, a cleaning function is created in R. This involves standardising all variables, just for the purposes of linkage, to lower case and removing any white space before or after characters which may have been mistakenly entered by initial data entry. This caused multiple problems on initial linkages as many observations had stray white space before, after and within names making comparisons difficult. In addition erroneous capitalisation was frequently also a problem.

The exercise was carried out first with both birth and death certificates, representing the easiest to match with only one entry in each record per person, in comparison to marriages and events where each person may have multiple observations. The first step was to remove duplicate rows containing identical information. This process of sub-setting the tables to retain only unique information removed 171 rows from the birth table and 183 from death[1].

Following this the cleaning function was carried out on both tables. This removed white spaces from before or after each string. It also removed blank information and stray punctuation inputted during data entry with no meaning and replaced with "NA" which is standard for the R package "record linkage".

After the first three rounds there were 1,514 pairs created. Of those 79 were new pairs not existing in the prior database, representing 5.2% of all new IDNR. These new pairs have been extracted for analysis to investigate why these were missed in the initial linkage. Many appear identical, indicating that it is the script cleaning stage which has led to new linkages and in others it is where information is missing and replaced with NA. However, 1431 cases that has the perfect match are used to create the first step of individual table.

3. Specification of standardized variables for meta data

Following Alter and Mandemakers (2014) and metadata file version 4, several variables provided from COR-2010 were selected, which it is planned to be release in our first COR IDS version. Most designated variables are present in COR-2010. Special attention was given for the time stamped information, concerning exact dates, periods and types of missing information (ibid, 21).

(Individual level)

- Name: last; first
- Birth: date; location
- Stillbirth: date; location
- Multiple birth: number of births
- Legitimacy: yes/no (clarify as they are many items on this)
- Recognition: date; location
- Nationality:
- Title:
- Marriage: date: sequence; location; proclamation date;
- Signature: birth: marriage
- Divorce: date: location
- Occupation: standardized; hisco status; hisco relation

[1] Within the birth and death tables there were some rows for people which were identical within the table rather than between.

- Death: date: location;
- Event status: alive
- Gender:
- Civil status:
- Age: years; months; weeks; daysObservation: start; end
- Arrival: from; to

(individual plus)

- Relationship: multiple types of familial and non-familial types.
- Name of source: population register, birth register, marriage register, funeral register.
- Location: Street; house_number; house_name; institution; latitude, longitude; inhabitants; page, volume; sequence number; period.
- Relationship to the municipality/locality: birth certificate; marriage certificate; death certificate; baptismal register; marriage register; funeral register; population register

4. Individual attributes on core variables from output 1

This output is attached to this note in a data format. Individual attributes on core variables from output 1 is included in the individual table. The authors provide individual attributes of core variables only and indicate consistent linkage (i.e. variable name link) between COR-2010 and COR-test-2017. The test version is attached to this note.

5. State of art on production of indiv-indiv data

The indiv-indiv relationships data serves as a valuable resource in conducting data analysis including social mobility and inequality, social networks and residential mobility etc. However such information requires detailed records of family and non-family networks over both spatial and longitudinal dimensions (Alter & Mandemakers 2017).

IDS in this context requires potentially numerous types of individual-individual information, specifying the exact observation time for each relationship.

Table 1 *Records in the table INDIV_INDIV (excluding timestamp variable)*

Id	Id_D	Id_I_1	Id_I_2	Source	Relation
1	HSN_release_2010.02	1	2	Birth certificate	Wife
2	HSN_release_2010.02	2	1	Population register	Husband
3	HSN_release_2010.02	1	22	Birth certificate	Mother
4	HSN_release_2010.02	22	1	Birth certificate	Child
5	HSN_release_2010.02	2	22	Population register	Father
6	HSN_release_2010.02	22	2	Marriage certificate	Child
7	HSN_release_2010.02	2	23	Population register	Householder
8	HSN_release_2010.02	23	2	Population register	Maid
9	HSN_release_2010.02	2	8493	Population register	Master
10	HSN_release_2010.02	8493	2	Population register	Servant
11	HSN_release_2010.02	823	824	Population register	Sibling
12	HSN_release_2010.02	824	823	Population register	Sibling

Source: Alter and Mandemakers (2014)

Creating indiv-indiv data in COR-IDS requires this information also. However, after consulting with Professor Kees Mandemakers, some shortcomings concerning the relationship level information

present in the COR-2010 population register were encountered. An illustration of this problem is first outlined and then possible solutions concerning how to convert the original information into the IDS required format are provided.

The first important issue concerns the type of information present in the COR-2010 database and its accuracy. COR-IDS requires a copy of raw materials where the first input file mirrors this exactly. This concerns the recording of the type of event, timing of event and name. COR-2010 not only documents the raw information, but also collects and interprets new information not present in the original material. For instance, all relationship variables regarding the association of individual to the head of household, and also to other household members are added, which are interpreted by the data collector (see below). The authos note that this additional information is a valuable source of information, but introduces potential errors to the data adding serious bias in the data set. In other words, the mixture of raw and obtained new information present in COR-2010 requires additional evaluation to review the quality of the information collected and interpreted by the data collector. This information is also not always complete resulting into high proportion missing.

There are currently at least the following variables present in the COR2010 population register that can be important to the indiv-indiv table.

- (kin) relationship with head of household (relhhh)
- Identifier of head of household (indrhh)
- Identifier of the father (indnrva)
- Identifier of the mother (indnrmo)
- Relationship with other member of the household (rellid1-10): this information is furthermore coded by 60+ types of relationship distinguished by family and non-family relationship.
- Identifier of the family member (indr1-10)

The schema of the relation code is the following (Dutch):

	00: referentiepersoon
0: 0de generatie	01: vader of moeder 02: schoonvader of schoonmoeder 03: stiefvader of stiefmoeder 04: oom of tante 05: voogd 06: grootvader/grootmoeder 07: stiefschoonvader/moeder 08: stiefgrootvader/moeder 09: overgrootvader/moeder

1: 1ste generatie	11: gehuwde partner 12: ongehuwde partner (concubin of concubine) 13: broer of zus
	14: schoonbroer of schoonzus 15: stiefbroer of stiefzus 16: halfbroer of halfzus 17: neef of nicht (kind van oom of tante, FR: cousin/e) 18: halfschoonbroer/halfschoonzus 19: halfneef/nicht (kind van oom of tante)
2: 2de generatie	21: zoon of dochter 22: zoon of dochter uit een vorig huwelijk 23: onwettige zoon of dochter 24: schoonzoon of schoondochter 25 stiefzoon of stiefdochter 27: bevoogd kind 28: neef of nicht (oom/tantezegger, FR: neveu of nièce) 29: halfneef/nicht (oom/tantezegger)
3: 3de generatie	31: kleinzoon of kleindochter 32: stiefkleinzoon of stiefkleindochter 33: schoonkleinzoon of schoonkleindochter 34: achterkleinkind
4: andere familierelatie	40: onbekende familierelatie 41: andere familierelatie
5: geen familierelatie	51: collectief huishouden 52: andere relatie 53: buur (bij aktes) 54: ambtenaar burgerlijke stand (bij aktes) 55: bekende (bij aktes) 56: personeel burgerlijk gasthuis (bij aktes, bv. dienstbode, directeur)
6: onbekende relatie	60: onbekende relatie

Source Van Balen (2007) pp.25-26.

The nature of the records in COR_2010 is Illustrated here.

How to get ID mother: First, sort all individual units by identification house. This allows to identify individuals to select to those who have resided one moment in time in the same location. All individual identifier is indicated in IDNR. Number in the household is given where 1 is the head of the household. Identifying the id number of the mother (=2625) can be done by matching the number in the column number in household, Idnrmo and then to IDNR. The assigned idnr of the mother is created at idmoeder. As it can already be seen, this process of identifying the id of the mother is complex.

How to identify siblings: rellid equals to 13 for household number 22, 23 and 24. Since 13 is brother or sisters, the authors assume that these 3 individuals are sisters/brothers and the mother is id 2625. As this information is interpreted, they assume that they are biologically related but cannot verify. The remaining rellidX variables and corresponding indrX are great mystery as the authors are uncertain which relationship they refer to given the fact that in this location, only number of the household 1, 21-24 and 29 exists and all others in between are not recorded. The authors find the guessing and reinterpreting of the relationship between individuals to be particularly problematic to be used for the creation of indiv-indiv table.

In addition to the above, there are spatial information present in the location table which potentially allows to create the relationship variable when the location of residence is shared. This is the potential information that may be used to create indiv-indiv attribute by replacing the source information of the items noted just above.

- Reference house information concerning
  - Municipality (refgem)
  - Population register (refBR)
  - Quarter number (refwk)
  - Quarter house number (refwkhsnr)
  - Street (refstr)
  - House number (refhsnr)

The authors consider “workable” solutions to the aforementioned problem given the circumstances: what methods they can apply in obtaining the indiv-indiv table. At the conceptualization stage, they first acknowledge that there exist familial and non-familial relationship to each individual.

The first familial ones are namely determined by intergenerational linkages mostly referring to mothers, fathers and children, and if identifiable aunts, uncles, cousins. They consider that these information can be identified and matched using relevant sources, primarily through certificates.

The second ones refer to the location where non-familial relationship is established when they share the same address for instance. This can be achieved by making use of location table, population register (if possible) and then also with the relevant certificates. For the latter, the obtained location identifier is present (a1, a10, …) (population register, huissamen2) where these sources allow to match with the relevant individuals in the population register. However, if the same is applied theyto only work with raw inputs and not with the acquired variable present source, they will need to match location with individuals through addresses. This means another linkage based on location variable can be performed by identifying location (addresses) present in the certificates and

population register. This step of new direction in their implementation, will however mean that they will first establish a separate table on context, then link between individual and context and lastly determine individual-individual (i.e. individual – context do overlap with this). At this stage, the authors are rather more confident to identify familial relationship than non-familial ones. Note should be also made that within non-familial relationship there are at least two types: co-residence and non-co residence but based on social network (e.g. witnesses of events).

Appendix B.

This appendix describes the outputs for the construction of extraction software belonging to the section of ESR 2 project at the KNAW, Amsterdam. This software is still under construction and more modules and R version of already developed software will be added.

1. Extraction Software for the Constant Features of IDS

The objective of this software is to keep track of only the constant features of the table INDIVIDUAL. They are time invariant and we can find aspects like birth location and date, or those linked to the individual's baptism, death, funeral (again with date and location). We also include other aspects like the name. Concerning the latter, we assume that in some databases (as in the Historical Sample of the Netherlands) this component need not change, as women kept their last names after marrying (although this would not be applicable to other databases)

The program has the following milestones:

Firstly, the tables are read and uploaded to RStudio. We need to export the INDIVIDUAL table in Access to a csv format (changing the comas to dots for numbers). Through this all the records in this table belonging to HSN were loaded. The software needs to check if all the files to be used are available. In this case, the INDIVIDUAL table should not be empty.

```
if (nrow(individual) == 0 ) {
print("File INDIVIDUAL.xlsx empty")
} else {print("File INDIVIDUAL.xlsx is not empty")}
```

In the second part, a vector with all the constant features of the registers (those that don't change on time), is created. The idea is to set a list of the variable Type that collects these characteristics. For instance, the birth date or the birth location, as they won't change across the individual's lifetime.

The vector with the static information is the one that follows:

```
a <- c ("BIRTH_DATE",
        "BIRTH_LOCATION",
        "BAPTISM_DATE",
        "BAPTISM_LOCATION",
        "DEATH_DATE",
        "DEATH_LOCATION",
        "FUNERAL_DATE",
        "FUNERAL_LOCATION",
        "STILLBIRTH_DATE",
        "STILLBIRTH_LOCATION",
        "LAST_NAME",
        "NAME",
```

```
"PREFIX_LAST_NAME",
"FIRST_NAME",
"LEGITIMACY",
"RECOGNITION_DATE",
"RECOGNITION_LOCATION",
"SEX",
"CHILDBIRTH_ASSISTANT")
```

After the isolation of the static characteristics, the next step is to go over the dataset (table INDIVIDUAL), and if we identify any of the constant features of the vector, we select the correspondent register (line of the table INDIVIDUAL). This line is then added to the subset of the table we are developing. This is performed through the command line:

```
individual <- individual [which (individual$Type %in% a), ]
```

After it, we get the desired result of static information of the INDIVDUAL table.

2. Extraction Software for migration features

The objective for this software is to collect migration features within the records present in the IDS. As this is not reported (by way of the creation of specific variables) in the IDS tables, the goal is to collect the data that keep track of the changes individuals go through concerning the locations they are registered at. The idea behind was to document the possible changes appearing in the Value_Id_C variable, in the table INDIVIDUAL. This variable gives the code of a certain context that can be traced back using the CONTEXT_CONTEXT and CONTEXT tables.

For that purpose, the program developed the following steps:

Firstly, the variable “date” is created because we aim to organize the records with respect to their dates (and also by their identification). This way we start by initially grouping all the registers by their personal ID (Id_I) in the database they were collected from. Once grouped, the records within each group are also ordered time wise (by the date each record has).

```
individual$date <- as.Date(paste(individual$Year, individual$Month, individual$Day, sep = "-"))
individual <- individual[order(individual$Id_I, individual$date), ]
individual <- individual[!duplicated(individual), ]
```

After this, the code runs the reorganized dataset checking for changes on the variable Value_Id_c, showing potential changes in the context for the particular time considered. This way, a change in the location could be evaluated as a potential case of migration.

The steps are:

a) Creation of an auxiliary table (individual2):

```
individual2 <- individual
```

b) Running of the table and checking for changes in the context

```
for (n in 1:nrow(individual2)){
```

```
 if(individual2$Value_Id_C[n] != individual2$Value_Id_C[n+1]){
  individual2$flag <- 1
  }
}
```

c) Sub-setting of only the records that show changes in their context.

```
individual2 <- subset(individual2[which(individual2$flag=="1")])
```

Finally, we can collect the records linked to some variables of the INDIVIDUAL table that may report about migration. Variables like “NATIONALITY", "NATIONALITY_STANDARD", "NATIONALITY_ENG", "ARRIVAL_FROM", "DEPARTURE_TO" may be interesting for that purpose. In order to gather this information, the code is as follows:

First, the code asks the user of the extraction software to fill in the nationality of the database, and we store this information in the variable database.nationality:

```
database.nationality <- readline (prompt="Enter database origin: ")
```

Then, we select the extra variables that provide some information liable to be linked to where the records are originally from, and those of which information could be different to the database ambit or scope. In other words, if the user is extracting software from the HSN and some of the records within the Individual table have a different nationality rather than Dutch, then, we can assume this record may be included as immigrant.

```
ind <- c ("NATIONALITY", "NATIONALITY_STANDARD", "NATIONALITY_ENG")
```

In the next steps, we check if the records’ reported nationality is the same as the one of the database
If so, we select them and add them to the new subset of the dataset (Individual)

```
If (database.nationality %in% ind){
  individual3 <- individual[which(individual$Type %in% ind),]
 }
```

We can also include information about register's arrivals from or departure to by way of the same procedure: selection of variables:

```
ind2 <- c ("ARRIVAL_FROM", "DEPARTURE_TO")
```

And sub-setting of the Individual table to those records that have information of the latter:

```
Individual4 <- individual [which (individual$Type %in% ind2),]
```

After all these last steps, we bind the three tables (Individual2, 3 and 4) together:

```
Individual_Final <- rbind(individual2, individual3, individual4)
```

For the remaining two modules, the philosophy is very similar: first we select the variables from the variable Type of Individual, and then we subset those records of the table Individual that have these Type variables documented.

3. Extraction Software for religion features

The process is as follows:

a) We collect the data linked to the Type variable of the INDIVIDUAL table that may report about religion. Variables like "BAPTISM_LOCATION", "RELIGION", "RELIGION_STANDARD" may be interesting for that purpose.

```
ind <- c ("BAPTISM_LOCATION", "RELIGION", "RELIGION_STANDARD")
```

b) After that, we need to subset the table Individual for only the records that have some or all of these features:

```
individual<- individual [which(individual$Type %in% ind),]
```

4. Extraction Software for occupation features

The same steps go for the occupations:

a) Discrimination of the variables with occupation features

```
ind <- c ( "OCCUPATION", "OCCUPATION_STANDARD", "OCCUPATION_ENG",
           "OCCUPATION_HISCO", "OCCUPATION_HISCO_STATUS",
           "OCCUPATION_HISCO_RELATION", "OCCUPATION_HISCO_PRODUCT")
```

b) After that, we perform the subsetting of the table Individual for only the records that have some or all of these features:

```
individual<- individual [which(individual$Type %in% ind),]
```