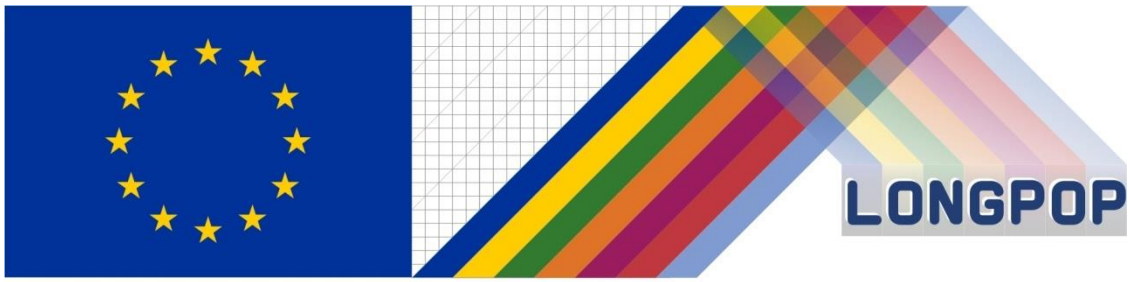

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 676060

LONGPOP

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

Report on the coordination of the building of extraction software

Deliverable 3.2.

Contents

Acronyms used in this document

ESR: Early-Stage Research

COR-IDS: Intermediate Data Structure based on the Antwerp COR-database

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).

KU Leuven: Catholic University of Leuven

TELNET: Redes Inteligentes (inTELligent NETworks)

WP3: Work Package 3

1. Introduction

In this report we are summarizing those projects in which IISG-KNAW (the International Institute of Social History – the Royal Dutch Academy of Arts and Sciences) participated through ESR 11 or ESR 2 in the coordination and further development of software devoted to the extraction, managing and reconstruction of data belonging to large historical data sets.

Four different projects can be highlighted in collaboration with TELNET, Radboud University of Nijmegen, the Catholic University of Leuven (KU Leuven) and the Institute of Statistics and Cartography of Andalusia (IECA).

The first involves the process of creating the infrastructure necessary to build elastic search in order for it to be accessible from different databases and from different countries. The tasks were planned mostly at the IISG, and carried out by his developer ESR 8, Vasilis Giagloglou.

The second one mostly involves the reconstruction of family units and households from COR database. These tasks were undertaken together with ESR 7, Sam Jenkinson, at the University of Leuven. The coordination aimed at the construction of the INDIV_INDIV and INDIV_CONTEXT tables of the future COR IDS database, once it was possible to reorganize the data from the birth, marriage and population registers.

The third carries out the restructuring of extracted data from the original historical population records with the goal of retrieving all possible family connections between records in the data sets. The exercise was run in collaboration with the Radboud University.

Finally, the last one outlines the steps undertaken in order to produce the R version of the IDS Transposer tool, and it was developed at IECA.

Except for the work in IECA-Sevilla, done by ESR 11, and for the one in the Radboud University, accomplished by ESR 2, these projects were also discussed in deliverable 3.1 of WP3. The main differences between the two reports is that the one presented for deliverable 3.1 gives an overview of the work done by ESRs alone in their projects, whereas the present report (deliverable 3.2) shows the achievements of conjoint work by IISG-KNAW and other institutions on the development of software for extraction and/or management of historical data tables.

2. Institutions with which IISG collaborated

TELNET, Spain

IDS database format makes easy demographic, historic and sociologic research based on individual-level data to be compared at national and international scales. However, issues of anonymity add difficulty for sharing the information by these databases and thus limiting the possibility for a full-scale research.

To overcome this, a conjoint work led by ESR 8 is being carried out. Using Elasticsearch we can provide a safe distributed solution for doing analysis to all the IDS databases and extract data for analysis, either with Kibana (the native web interface of elasticsearch), or with R (the programming language for analysis). Now all the data can be shared to trusted parties in a safe network of IDS users, to be compared, extracted, analysed in international scales. All IDS databases can be available, in an easy to use web client, from all the interested scientists.

Elasticsearch is a powerful search engine with the possibility to visualize and select valuable data from IDS databases without the necessary programming skills.

The work of coordination involves the making of a preliminary format adjustment to Elasticsearch in order to be used by every stakeholder interested in IDS.

Set up and configuration. Steps of the project.

The IDS database we use as a sample is the HSN_IDS (Historical sample of Netherlands). The HSN is a representative sample of about 85,000 people born in the Netherlands in the period of 1812 – 1922 with individual life courses. This database is a unique tool for research in Dutch history and demography. Additionally, the HSN is involved in different projects where comparison of data and collaborative research is important (www.iisg.nl/~hsn/projects), where compatibility of access is important.

In order to provide HSN_IDS database with the correct format and for further use, PowerBI and DAX studio were used. Th purpose of them is to do a preliminary cleaning of the tables (removing empty columns and dumping the information into csv format). CSV is a format suitable for elasticsearch. These tools were used for extracting samples of the database into csv.

The reasons for the use of samples and not the whole of the database are mainly double folded:

1. In order to use the full functionality of elasticsearch, the server needs 32 gb of RAM, whereas our laptop has only 8 gb of RAM.
2. The second reason is security. Elasticsearch does not come with any security backed into it. Anyone with access to the server url can wipe the whole setup clean via curl commands.

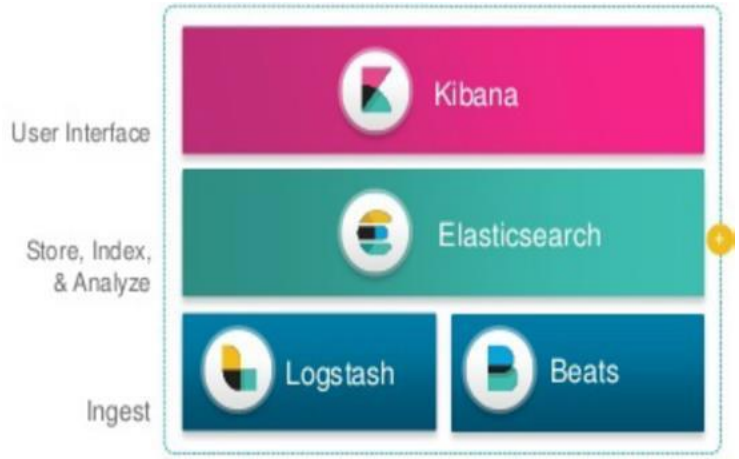

Figure 1. The complete ELK stack

For installing data into elasticsearch we decided to usElasticsearch, Logstash, Kibana and Filebeat, where the latter is the data shipper into the ELK Stack.

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

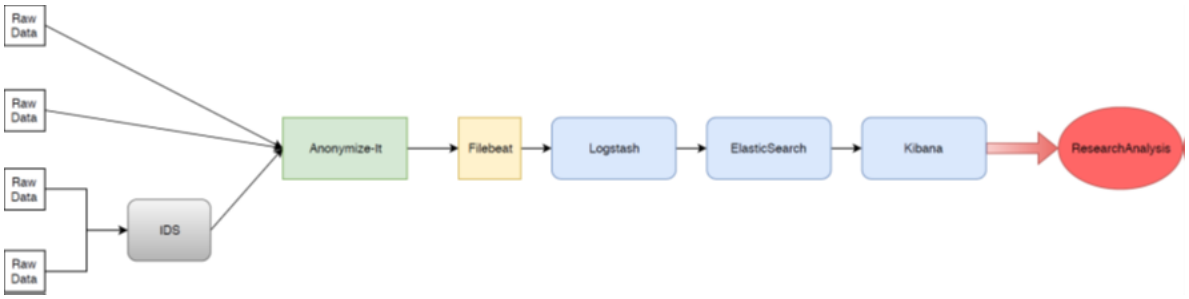

Figure 2. Preliminary diagram of the LONGPOP project

Our goal with the project is to present the potentiality of elasticsearch and Kibana for the use in terms of visualisation and feature extraction of an IDS format databases. A preliminary work has been made for the implementation of the tools as described in the section.

KU Leuven, Belgium

Introduction

The following paragraphs outline the main concepts and steps taken in the collaborative project of the development of software routines for the extraction, management and reconstruction of family units within the Antwerp COR database, and aiming at the development of the Intermediate Data Structure (IDS) version of this historical database.

For the process of the reconstruction of households we counted on four different sources or data sets. The Residence file (56,368 records) provided information of each person´s address, neighbourhood and a reference number for each household compound, as well as their personal information, like names, age, birth dates and place of birth, given also a unique individual identifier or IDNR. The civil status and occupations were provided, as well as the family roles or relations towards other potential household members. Nevertheless, most of the relations provided by the sources where unknown -almost 20,000 out of more than 56,000 observations. From the whole set, another 14,000 registers where stated as heads of their households. One extra important feature is also the period of reference of the population registers, that delimits the time individuals lived in a particular place.

In the Events file (126,399 records), along with the personal information of each IDNR, like names and address, we can find a collection of events that the dwellers of those addresses may experiment in time. They refer to eight possible situations, as the following: migration in or out the municipality; moving in or out a particular address; marriage; widowhood; death; and 'others', like divorce or separation.

For this previous two sources, a time period of reference was also available, helping to delimit the structure of potential households in time.

In addition, the Births file (7,222 records), among other things, collects information of birth and death of individuals, as well as their residence, and finally, the Marriages file (2,118 records) documents information on the IDNR's marriages, including witnesses' names and relation to the groom and the bride, the couple´s parents' names from both sides, as well as their children's names.

It also gathers information from other marriages and divorces the members of a couple may have experienced.

Methodology

On the basis of these sources, the methodology applied for this research project is constituted by a block of common methodology and a second part with two differentiated approaches.

Common methods

The goal in the common section was to merge the large amount of information contained in the Events file along with that of the Residence file. This was done by means of four common features in both data sets: the unique identifier of individuals, the residence address, the identification number of the house and the period of reference. In doing so we expected to connect individuals from both datasets under a certain address, sharing the same space defined by the unique number and for a certain period in time.

Prior to that purpose, a process of cleaning and harmonization of the data was conducted, neglecting many irrelevant variables, making up impossible dates and unifying formats. Also important was to compensate the lack of data in some sources with the others. For instance, around 7,800 observations from the Residence file didn´t have a birth year, but by means of the Births file, almost 5,800 birth years could be granted to them.

Two Approaches

Once carried out the initial merge, the first approach focussed on the addition of the events that individuals underwent in their places of residence, as well as the changes they experimented during their stay or when they moved out.

On the one hand we made use of the information provided by four of the types of events involving the movement of proto-family units. The relevant events for our purposes where those granting individuals movement from or to a new address, whether in or outside the same municipality. In several cases we detected there was a joint movement of several dwellers of the same address on the same moment in time from and to the same location and addresses. Based on this characteristic of the sources we can shape the composition of some households to the extent of the available information in them, and also to follow some evolution in time.

We initially discarded those events that genuinely involved one single person occurrences, like death or widowhood, as they were not so constructive towards the understanding of the configuration of each multiple member entity. Even though these events can also configure the composition of a household from our perspective, once it is formed, for the process of its reconstruction, a one-individual event doesn´t help to identify more family members and their roles. Nevertheless, they do after the units are outlined.

On the basis of this first approach, we also made use of the information that gives evidence of the relation, at a specific moment in time and address, of the dwellers with respect to the person stated as the head –which was given by one of the variables in the Residence file[1]-, spreading some light over the roles of the components of each unit.

On the other hand, the second approach neglected the use of events, focussing on the development of a kind of algorithm based on some assumptions and information from the sources. In this sense

[1] The variable *relhhh of the Residence file.*

the composition of households is based on the sharing of three common characteristics: address, identification number of the house and the period of reference. Once this is achieved, all the potential members undergo two basic assumptions. First, the elder member of each group is assigned the value of *head person*. Second, members sharing the same family name and a wider age difference with respect to the head of more than or equal to fifteen years, are assumed to be the children. This assumption seemed to work reasonably fine as we had 34,582 observations of head persons in the sources, whereas 12,396 were derived by means of assumptions (36% of the total), with 6,374 matches between both.

We can also say that have 13,138 observations of sons or daughters from the sources, while through our assumptions we achieve 7,654 (58% of the total amount). In 4,472 times they both matched, which also accounts for 58% of the derived feature.

Finally, for the reconstitution of the members of the household we made use of some extra information in the civil registers. In them, espouses are identified by the same unique number as in the population registers. Then the goal is to identify under which household they lived together, which can easily be done if for a particular address, identification number and period, the two members are found to be sharing these combinations.

From the marriage certificates, the same procedure was carried out in order to potentially include the information provided for witnesses and parents, as sometimes witnesses may be part of the households as well as parents, but no matches where achieved this way. One of the reasons is that the number of observation in the Marriages file accounted only for 2,118 couples compared to the 29,602 unique individuals in the Residence file.

Radboud University Nijmegen, The Netherlands

Introduction

The project we explain in this section focused on the so-called *thombos*, a sort of sources created by the Dutch colonial government for a detailed recording of the indigenous population of Sri Lanka, including the kin relationships of extended family members co-residing on ancestral land.

In this project the IISG-KNAW participated providing with approaches to transform the stated kin relations to the household head into dyadic relations among all family members, thus allowing us to simulate co-residence of kin across the life course in 18th century Sri Lanka.

The process of redesigning the input data set with the original kinship relations from the sources has the following steps. The goal of them was to reconstruct all the empty relations between the members in the original input table, giving a rectangular format for an output table where each person from the sample could be documented in a row together with all the possible relations/roles this person has with the rest of the members of the sample.

The idea was thus, to pivot our source table converting it into one with the basic information of each person (ID -or identification number-, name and sex), together with as many new variables as all the kinship roles this ID has with the rest of IDs in the initial data set.

Methodology

Input table and the format one-to-one

After loading the input table, we neglect those irrelevant variables for our purpose of formatting it. The input file gives a bi-univocal perspective of the roles in the sample, providing paired relations for each two individuals through their family ties. This is done by presenting the roles of both individuals from the perspective of each other. For instance, if the person of reference is the head, then initial data set will provide his/her unique identification number (ID), his/her name (first and last), sex and the role this person has towards the second. From the other side, the second person will also have his/her unique identification number (ID), his/her name (first and last), sex and the role this second person has towards the head.

Any other variable is not significant.

Enhancement of the blanks in the input table by creating the new roles

Given this structure, we begin by creating as many roles as we can think between all the members of the sample family in the one-to-one format described in the previous paragraph. If in the sources, we had for example two siblings stated as *brother 1* and b*rother 2*, and *brother 1* had also another sibling (*sister 3*) who was not documented as sister of *brother 2*, then, our software should be capable of filling this gap in and of creating the third link of the triangle. Thus, it should create a new entry for the relation between *brother* 2 and *sister 3*, stating that are also siblings: *brother 2* is brother of *sister 3*, and *sister 3* is sister of *brother 2*.

This simple concept was translated into the proper algorithm using SQL language by creating two big case blocks. In each block we had the perspective of the family relation from one of the two persons, and in the second case block we had the perspective of the second individual. The key for them to be linked is the existence of a third party who is the person that connected to both and by whom the roles are given.

In our example, this third party is represented by *brother 1*, and we can analyse the relation either from the perspective of *brother 2* (our algorithm's first case block), or from the perspective of *sister 3* (our algorithm's second case block). Then, from the perspective of *brother 1* the logic applied is: "if *brother 1* is the brother of *brother 2*, and *brother 1* is the brother of *sister 3*, then *brother 2* is brother of *sister 3*"

From the reverse perspective, we would just say that *sister 3* is the sister of *brother 2*. Something like this:

```
CASE
        WHEN person1.Role = brother' AND person2.Role = 'sister'
        THEN 'brother'
CASE
        WHEN person1.Role = 'sister' AND person2.Role = 'brother'
        THEN 'sister'
```

Once this part with all the potential family roles in a one-to-one structure is finished, then we start reformatting it with the aim of the new rectangular shape of the table. For it, two extra steps are needed.

Loop to account for the new roles and the maximum number of them, per ID

In the first, we account for the number of different roles in the sample file, as well as the maximum number that each role has in it. In other words, we run the whole sample counting for each ID that the family roles this ID has, and how many. For instance, *ID 1* may have three brothers and five

uncles. Then, if the rest of IDs in the data set don't have as many brothers and as many uncles as *ID 1*, the final output table will show the new columns for the roles: three columns for brothers (*brother 1, brother 2, brother 3*) and five more for the uncles (*uncle 1, uncle 2, uncle 3, uncle 4, uncle 5*).

The algorithm to do so requires a loop that runs a) the IDs and b) the maximum number of roles, creating a new variable that collects all the potential new roles. In our example: *brother 1, brother 2, brother 3, uncle 1, uncle 2, uncle 3, uncle 4* and *uncle 5*.

Building of the first columns of the output table with the features of the ID

We start to build the output table with two columns: the first providing the ID and name of each individual and the second, his or her sex.

Last loop for the building of the rectangular table by filling in all the roles

Once at this stage, all we need to do is to run the last loop accounting for a) all the IDs and b) all the rows of the input files, and we will be adding as many new columns to the table as maximum number of roles are there in the sample, also filling in the values of the correspondent cell with the ID and the name of the person.

IECA, Spain

Among other general tasks, IISG-KNAW was involved through ESR 11 in the development of a tool that transforms large population register databases into the IDS format, so that extraction software can be applied to work with records.

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.
This software was applied to the data base of the IECA, and it's underway to produce its IDS version, and it will be applied during the month of August to the COR database of Antwerp to transform it also in its IDS version.

The process of creation of the software tool involved the following milestones (functions build per stage):

1. Creation of the function *Episod2ymd* which transforms two dates (defining an episode) into a data.table with 6 columns. It has as arguments a vector with the start dates of episodes (start.date), and a vector with ending dates of the episode (end.date). The outcome is a "data.table" object with the columns: Start_year Start_month Start_day End_year End_month End_day.

2. Creation of the function *date2ymd* which transforms a date vector into a list with year, month, day. It has as arguments a date and a vector with dates. The outcome is a "data.table" object with the columns Year, Month and Day.

3. Creation of the function *ddate2date.3d* which transforms date in decimal year format to range of days. It has as arguments a date and a numeric value in years with decimals. The outcome is a "data.table" object with three dates: start.date, end.date, central.date.

4. Creation of the function *granule.data2ymd* which transforms a table of 3 columns in format dates to a table with 9 columns of integers (converting each date into 3 integers: year, month, day). It has as arguments a date of a data.table with columns: start.date, end.date and central.date. The

outcome is a “data.table” object with the columns Year, Month, Day, Start_year, Start_month, Start_day, End_year, End_month, End_day.

5. Creation of the function ids.skeleton contains an empty structure list of the tables of the Intermediate Data Structure: INDIVIDUAL, CONTEXT, INDIV_INDIV, INDIV_CONTEX, CONTEX_CONTEX.

6. Creation of the function *transposer* which is the main function to transfer a set of data.table(s) present in global environment (.GlobalEnv) to an IDS structure. It has as arguments a

- file.definition: xlsx-files aiming at performing the transfer of the data contained in the data.table objects present in .GlobalEnv to the tables required by the IDS format
- sheet: Name of the pages within the book that contain the instructions. Only 'Entity' and/or 'Relationship' is allowed.
- Name.DataBase: Name of the database that identifies the data within the IDS network.
- output.csv: path where IDS files are stored. If output.csv = NA csv files are not produced

The outcome is a list with the tables INDIVIDUAL, CONTEXT, INDIV_INDIV, INDIV_CONTEXT, CONTEXT_CONTEXT with the results of the data transfer

3. References

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

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