European Doctoral School of Demography (EDSD)
Computer Programming E140

Özer Bakar
Calderon Bernal Liliana Patricia
Gonzalo Daniel Garcia
Ainhoa Elena Leger
Özge Elif Özer

10 October 2020

Exercise 1

The first step to work with the practice data set of the German Socio Economic Panel (SOEP) is to save it locally. After telling R where we want the data to be located, we download and unzip it in the specified path.

```
# Directory to be set by the user
setwd("C:/Users/Ainhoa/Desktop/EDSD-PHD/1 - EDSD/2 - Preparatory courses/4 - C Dudel - Computer Programm

# Download and unzip the file
soep_url <- "https://www.diw.de/documents/dokumentenarchiv/17/diw_01.c.412698.de/soep_lebensz_en.zip"
destfile <- "soep_lebensz_en.zip"
download.file(soep_url, destfile)
unzip(zipfile = "soep_lebensz_en.zip")
```

1a) Load the data set into R

We load the `foreign` and `tidyverse` packages which will be useful to solve the exercises. The `read.dta()` function allows us to load the Stata's database into R.

```
# Loading necessary libraries to solve the assignment
#install.packages("foreign")
#install.packages("tidyverse")

library(foreign)
library(tidyverse)

# Importing the Stata data into R framework by using foreign library
soep <- read.dta("soep_lebensz_en.dta", convert.factors = TRUE)
```

Let's have a look whether the data set has been correctly imported.

```
glimpse(soep)
```

```
## Rows: 12,922
## Columns: 9
## $ id         <int> 312, 399, 399, 457, 457, 457, 748, 761, 761, 1044, 1044,...
## $ year       <int> 2004, 2000, 2001, 2000, 2002, 2004, 2000, 2000, 2001, 20...
```

```
## $ sex        <fct> female, male, male, male, male, male, female, female, fe...
## $ education  <dbl> NA, 12.0, 12.0, 18.0, 18.0, 18.0, 14.0, 16.0, 16.0, 14.0...
## $ no_kids    <dbl> 1, 1, 1, 0, 0, 0, 0, 2, 2, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,...
## $ health_org <fct> good, good, good, satisfactory, satisfactory, poor, sati...
## $ satisf_org <fct> 7 Satisfied: On Scale 0-Low to 10-High, 8 Satisfied: On ...
## $ health_std <dbl> 0.5670103, 0.5670103, 0.5670103, -0.4639175, -0.4639175,...
## $ satisf_std <dbl> -0.09090909, 0.47727272, 1.04545450, 0.47727272, -0.0909...
```

The practice data set consists of a total of 9 variables and 12922 observations. Two variables identify the individuals and the year of the survey, while the other collect information about sex, education measured in years, number of kids, perceived subjective health, satisfaction in life and their standardized versions.

1b) How many unique individuals are included in the practice data set?

We just need to find out how many identification number ID are in the data set, as they are unique for each individual. The function `unique()` returns the vector with duplicate elements removed and the function `length()` returns the length of that vector.

```
soep$id %>% unique %>% length
```

```
## [1] 3550
```

The same result can be obtained with base R.

```
length(unique(soep$id))
```

There are 3550 distinct individuals in the data set.

1c) Tabulate the number of observations per year

We group the observations by year and count the observation for each year thanks to the function `tally()`.

```
obs_per_year <- soep %>%
                  group_by(year) %>%
                  tally()
```

There are 5 years of measurement and more observations in the most distant years. An alternative in base R is to use the function `table()`. To have an idea of the relative frequencies, we apply the function `prop.table()`.

```
prop.table(table(soep$year))
```

```
##
##      2000      2001      2002      2003      2004
## 0.2474849 0.2081721 0.1923077 0.1779136 0.1741217
```

The observations in 2000 constitute the 25% of the data set and the ones in 2004 the 17% of the data set.

1d) Restrict the data to the most recent year

Both the functions `filter()` and `subset()` return a subset of the data and can be used to retain all rows for which the year is the last available.

```
last_soep <- soep %>%
               filter(year==max(year))

dim(last_soep) # ok
```

```
## [1] 2250    9
```

As expected, the new data set contains the same 9 variables but 2250 observations from 2004.

What is the proportion of females in this subset of the data?

The function `prop.table()` used together with the function `table()` gives us the relative frequencies of females and males.

```
last_soep$sex %>% table %>% prop.table
```

```
## .
##      male    female
## 0.4577778 0.5422222
```

About 54.22% of the surveyed individuals are females.

Is the average subjective health higher for men or for women?

To obtain the average subjective health for men or for women we first need to create a numerical variable from the categorical variable health_org. We look at the levels of health_org and create our numerical variable.

```
# Levels of the variable health_org
levels(last_soep$health_org)
```

```
## [1] "not valid"        "does not concern" "no answer"        "bad"
## [5] "poor"             "satisfactory"     "good"             "very good"
```

```
# Creation of the corresponding numerical variable
last_soep$health_num[last_soep$health_org == c("not valid",
                                               "does not concern",
                                               "no answer")] <- 0
last_soep$health_num[last_soep$health_org =="bad"]          <- 1
last_soep$health_num[last_soep$health_org =="poor"]         <- 2
last_soep$health_num[last_soep$health_org =="satisfactory"] <- 3
last_soep$health_num[last_soep$health_org =="good"]         <- 4
last_soep$health_num[last_soep$health_org =="very good"]    <- 5
```

Let's double check whether the old variable and the new variable coincides.

```
table(last_soep$health_org, last_soep$health_num)
```

```
##
##                     1   2   3   4   5
##   not valid         0   0   0   0   0
##   does not concern  0   0   0   0   0
##   no answer         0   0   0   0   0
##   bad              73   0   0   0   0
##   poor              0 306   0   0   0
##   satisfactory      0   0 696   0   0
##   good              0   0   0 944   0
##   very good         0   0   0   0 231
```

The creation of the new variable has worked. We can now compute the means for the females and the males. The function `tapply()` allows us to apply the mean to health_num by group.

```
tapply(last_soep$health_num, last_soep$sex, mean)
```

```
##     male   female
## 3.464078 3.390164
```

The subjective health is perceived higher in men (3.46 on average) compared to women (3.39 on average).

The code below does the same operations and returns the same results.

```r
last_soep <- last_soep %>%
              mutate(health_org_numeric = case_when(
                health_org %in% c("not valid", "does not concern", "no answer") ~ 0,
                health_org == "bad" ~ 1,
                health_org == "poor" ~ 2,
                health_org == "satisfactory" ~ 3,
                health_org == "good" ~ 4,
                health_org == "very good" ~ 5,
                TRUE ~ NA_real_
                ))

last_soep %>%
  group_by(sex) %>%
  filter(health_org != 0) %>%
  summarise(mean_subjective_health = mean(health_org_numeric, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   sex    mean_subjective_health
##   <fct>                   <dbl>
## 1 male                     3.46
## 2 female                   3.39
```

Exercise 2

2a) Load the data

We choose to analyze life expectancies at birth in Italy and we used the `readHMDweb` from the package `HMDHFDplus` to read the data online from the The Human Mortality Database.

```r
#install.packages("HMDHFDplus")
library(HMDHFDplus)

# The user has to provide its own credentials (username and password)
italy_e0 <- readHMDweb("ITA","E0per", "ainoleger@gmail.com", "Allodola93")
```

The data loaded in R contain the life expectancies by gender and for the total population in Italy from 1872 to 2017. Below are the first and last rows of the data set.

```r
head(italy_e0)
```

```
##   Year Female  Male Total
## 1 1872  30.26 29.28 29.76
## 2 1873  31.84 31.49 31.66
## 3 1874  32.02 31.62 31.81
## 4 1875  31.66 31.13 31.39
## 5 1876  34.00 33.36 33.67
## 6 1877  35.20 34.75 34.96
```

```r
tail(italy_e0)
```

```
##     Year Female  Male Total
## 141 2012  84.53 79.83 82.28
## 142 2013  84.93 80.25 82.69
## 143 2014  85.15 80.54 82.95
```

```
## 144 2015  84.69 80.27 82.57
## 145 2016  85.23 80.79 83.10
## 146 2017  84.91 80.51 82.79
```

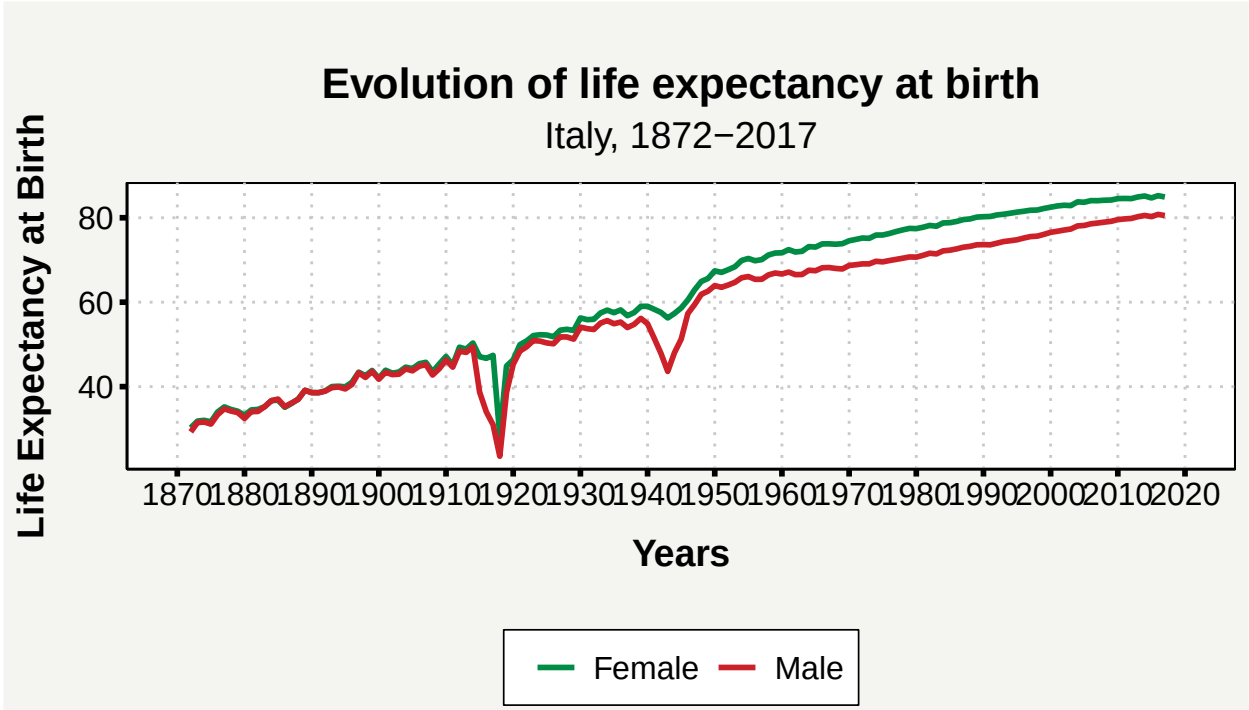
2b) Visualize the trend in life expectancy at birth

We personalize the background theme and plot the trends in life expectancy at birth by gender.

```
#install.packages("ggplot2")
library(ggplot2)

# Background theme
theme_graphs <- function (base_size = 16, base_family = "sans") {
  theme(plot.title = element_text(size = 16, face = "bold", hjust=0.5, margin = margin(20, 0, 5, 0)),
        plot.subtitle = element_text(colour = "#000000", size = 14,hjust=0.5, margin = margin(0, 0, 10,
        plot.caption = element_text(colour = "#000000", size = 12, hjust=1, margin = margin(10, 0, 20, (
        plot.background = element_rect(fill = "#F4F5F0"),
        panel.background = element_rect(fill = "white", colour = "#000000", linetype = "solid"),
        panel.grid.major.x = element_line(colour = "gray79", linetype = "dotted"),
        panel.grid.major.y = element_line(colour = "gray79", linetype = "dotted"),
        panel.grid.minor = element_blank(),
        axis.title.x = element_text(size = 14, colour = "#000000", hjust=0.5, face = "bold", margin = ma
        axis.title.y = element_text(size = 14, colour = "#000000", face = "bold", margin = margin(0, 10
        axis.text = element_text(size = 12, colour = "#000000"),
        axis.line.y = element_line(colour = "#000000"),
        axis.line.x = element_line(colour = "#000000"),
        axis.ticks = element_line(colour = "#000000", size = 1),
        legend.text = element_text(size = 12, colour = "#000000"),
        legend.background = element_rect(fill = "white", colour = "#000000", size = 0.3, linetype = "sol
        legend.key = element_rect(fill = NA),
        legend.position = "bottom",
        legend.direction = "horizontal")
}

# Trends in life expectancy
italy_e0 %>%
  select(Year, Female, Male) %>%
  pivot_longer(., cols = c(Male, Female), names_to = "Sex", values_to = "Life_Exp") %>%
  ggplot(aes(x = Year, y = Life_Exp, color = Sex)) +
  theme_graphs() +
  geom_line(size=1) +
  scale_colour_manual(limits=c("Female", "Male"), values=c("#008c45", "#cd212a")) +
  labs(title = "Evolution of life expectancy at birth",
       subtitle = "Italy, 1872-2017",
       x = "Years", y = "Life Expectancy at Birth",
       caption = "Own elaboration\nSource: Human Mortality Database. University of California, Berkeley
    scale_x_continuous(breaks=seq(from=1870,to=2020,by=10),limits=c(1870,2020))
```



The life expectancies at birth are increasing both for men and women. The sharp decreases in 1910-1920 and 1940-1950 are attribuable to war mortality.

2c) Visualize the evolution of the gender gap in life expectancy at birth over time

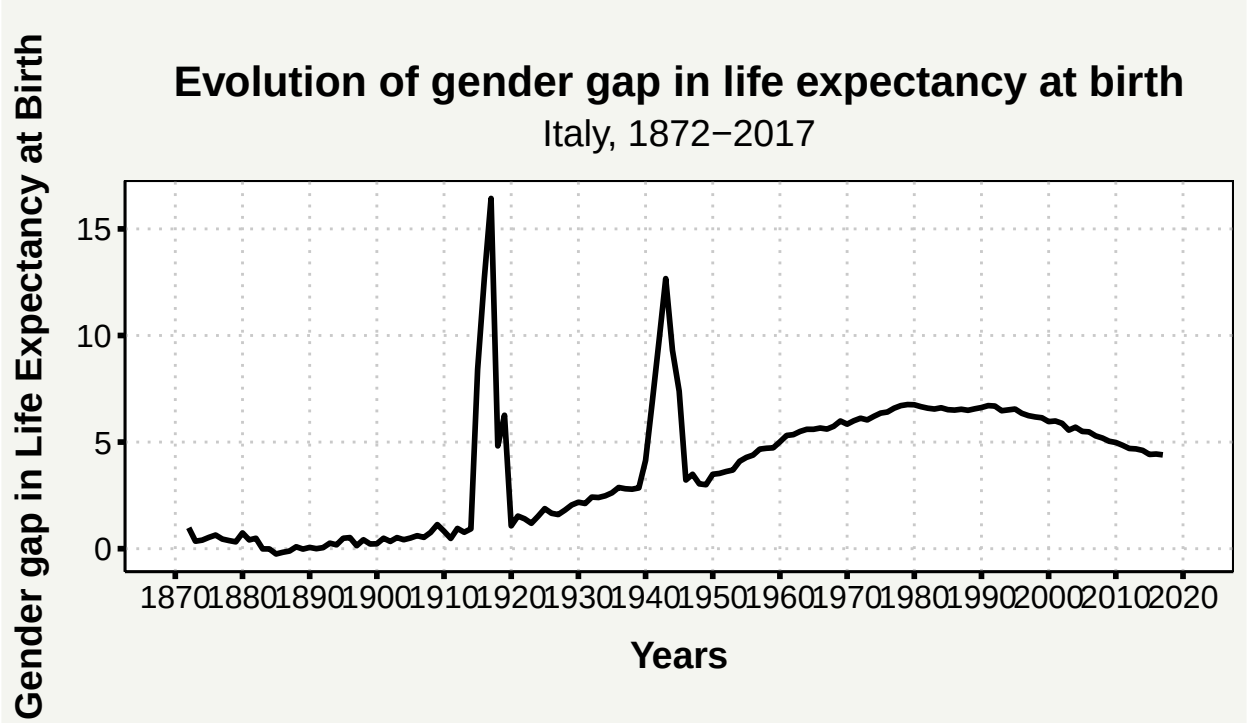
We define the gender gap in life expectancy at birth for Italy as the life expectancy of females minus the life expectancy of males.

```
italy_e0$gender_gap <- italy_e0$Female - italy_e0$Male
summary(italy_e0$gender_gap)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.2500  0.7625  4.2100  3.7829  6.0300 16.4300
```

Let's look how it has developed over time.

```
italy_e0 %>%
  ggplot(aes(x = Year, y = gender_gap)) +
  theme_graphs() +
  geom_line(size=1) +
  scale_colour_manual(limits=c("Female", "Male"), values=c("#008c45", "#cd212a")) +
  labs(title = "Evolution of gender gap in life expectancy at birth",
       subtitle = "Italy, 1872-2017",
       x = "Years", y = "Gender gap in Life Expectancy at Birth",
       caption = "Own elaboration\nSource: Human Mortality Database. University of California, Berkeley
  scale_x_continuous(breaks=seq(from=1870,to=2020,by=10),limits=c(1870,2020))
```



The gender gap has been increasing until 1980, after which it started to decrease.

References

Goebel, Jan, Markus M Grabka, Stefan Liebig, Martin Kroh, David Richter, Carsten Schröder, and Jürgen Schupp. 2019. "The German Socio-Economic Panel (Soep)." *Jahrbücher Für Nationalökonomie Und Statistik* 239 (2): 345–60.

HMD. n.d. "Human Mortality Database, University of California, Berkeley (Usa), and Max Planck Institute for Demographic Research (Germany)." https://www.mortality.org.

Riffe, Tim. 2015. "Reading Human Fertility Database and Human Mortality Database Data into R." TR-2015-004. MPIDR. http://www.demogr.mpg.de/en/projects_publications/publications_1904/mpidr_technical_reports/reading_human_fertility_database_and_human_mortality_database_data_into_r_5438.htm.

Team, R Core, and others. 2013. "R: A Language and Environment for Statistical Computing." Vienna, Austria.