Data Cleaning (Mattia Guarnerio - 14350920)

Mattia Guarnerio

2023-05-16

## Data Cleaning for Final Paper

Before executing the analysis on the ESS Waves 4, 6, and 9 data, I must appropriately clean it, only keeping the necessary countries and variables, and add the Level-2 and Level-3 (country) variables taken from the World Bank’s open data repository and Transparency International. For these tasks, I employ R and not Stata since the former is swifter when handling .csv files, such as the ones that can be downloaded from the World Bank website.

## ESS Wave 4 (2008) data set

I must take care of the following sub-tasks:

1. Drop all the Level-1 variables that do not concern my analysis. This means that I will be keeping *idno* - i.e., the respondent’s (Level-1) unique identifier - *cntry* - i.e., the country (Level-2) unique identifier - *euftf* - i.e., the dependent variable, the respondent’s degree of belief in European unification - *gndr* - i.e., the respondent’s gender - *agea* - i.e., the respondent’s age in years - *eduyrs* - i.e., the number of years of full-time education the respondent has completed - and *lrscale* - i.e., the main independent variable, the respondent’s placement on the left-to-right scale.
2. Exclude the countries that do not have sufficient time-variant data (Level-2) to be included - i.e., Greece, and Romania - or that are not European Union members. Remember that in this wave there is missing data at Level-2 for Italy.
3. Grand-mean centre *lrscale* and generate a new aggregate variable that indicates each country’s average placement on the left-to-right scale in the given wave - i.e., *cntry\_lrscale\_wave*.
4. Recode the *gndr* variable to a dummy *female* variable, which takes the value of 1 when the respondent is female, and grand-mean centre the rest of the Level-1 confounders - i.e., *agea*, and *eduyrs*.
5. Generate the Level-2 (country \* wave) unique identifier *cntry\_wave*, a numeric value that derives from the wave number (4, 6, or 9) - i.e., *wave* - multiplied by 100, added to the numeric transformation of the country identifier. Also, create a character version of this identifier to foster its visualization.
6. Repeat the previous six steps with the separate data set concerning Lithuania.
7. Attach the Lithuanian data to the main ESS Wave 4 dataset.

# Setting the working directory  
setwd("C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper")  
  
# Checking if the working directory was appropriately set  
getwd()

## [1] "C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper"

# Loading my data sets of interest: a. (ESS Round 4, 2008, N = 56752) b. (ESS  
# Round 4 (Lithuania), 2008, N = 2002)  
  
ess4 <- haven::read\_stata("ESS4e04\_5.dta")  
ess4lt <- haven::read\_stata("ESS4LT.dta")  
  
# 1) + 6)  
  
# Specifying the variables I will be keeping for my analysis  
vars <- c("idno", "cntry", "euftf", "gndr", "agea", "eduyrs", "lrscale")  
  
# Selecting the variables of interest by conditioning on the specified column  
# labels  
d4 <- ess4[, vars]  
d4lt <- ess4lt[, vars]  
  
# 2)  
  
unique(d4$cntry) # Checking all the unique country codes to see the identifiers that I must keep in the dataset to exclude all non-EU members and Greece

## [1] "BE" "BG" "CH" "CY" "CZ" "DE" "DK" "EE" "ES" "FI" "FR" "GB" "GR" "HR" "HU"  
## [16] "IE" "IL" "LV" "NL" "NO" "PL" "PT" "RO" "RU" "SE" "SI" "SK" "TR" "UA"

# I must keep Belgium (BE), Bulgaria (BG), Croatia (HR), Cyprus (CY), Czech  
# Republic (CZ), Denmark (DK), Estonia (EE), Finland (FI), France (FR), Germany  
# (DE), Hungary (HU), Ireland (IE), Latvia (LV), The Netherlands (NL), Poland  
# (PL), Portugal (PT), Slovakia (SK), Slovenia (SI), Spain (ES), Sweden (SE),  
# and the United Kingdom (GB).  
  
d4 <- subset(d4, d4$cntry == "BE" | d4$cntry == "BG" | d4$cntry == "HR" | d4$cntry ==  
 "CY" | d4$cntry == "CZ" | d4$cntry == "DK" | d4$cntry == "EE" | d4$cntry == "FI" |  
 d4$cntry == "FR" | d4$cntry == "DE" | d4$cntry == "HU" | d4$cntry == "IE" | d4$cntry ==  
 "LV" | d4$cntry == "NL" | d4$cntry == "PL" | d4$cntry == "PT" | d4$cntry == "SK" |  
 d4$cntry == "SI" | d4$cntry == "ES" | d4$cntry == "SE" | d4$cntry == "GB") # I am left with N = 39903 respondents  
  
unique(d4$cntry) # Checking all the unique country codes to assess whether this cleaning step worked smoothly

## [1] "BE" "BG" "CY" "CZ" "DE" "DK" "EE" "ES" "FI" "FR" "GB" "HR" "HU" "IE" "LV"  
## [16] "NL" "PL" "PT" "SE" "SI" "SK"

# 7)  
  
# I now attach the Lithuanian respondents to the rest of the observations,  
# since I need to execute operations of grand mean-centering.  
  
d4 <- rbind(d4, d4lt) # I am left with N = 41905 respondents  
  
unique(d4$cntry) # Checking all the unique country codes to assess whether Lithuanian respondents were added to the main ESS Wave 4 data set

## [1] "BE" "BG" "CY" "CZ" "DE" "DK" "EE" "ES" "FI" "FR" "GB" "HR" "HU" "IE" "LV"  
## [16] "NL" "PL" "PT" "SE" "SI" "SK" "LT"

rm(d4lt, ess4, ess4lt) # I remove the now unnecessary data sets to keep the environment clean  
  
# 3)  
  
# I mean-centre the 'lrscale' variable by subtracting its overall mean,  
# specifying that missing values must be excluded before computation.  
  
d4$lrscale\_m <- d4$lrscale - mean(d4$lrscale, na.rm = TRUE)  
  
# I group the observations by country, and then calculate the Level-2 (country,  
# time-variant) mean of 'lrscale\_m', specifying that missing values must be  
# removed before running the operation. I save the computed values in the  
# 'cntry\_lrscale\_wave' column, in a new auxiliary dataset  
  
aux <- d4 %>%  
 group\_by(cntry) %>%  
 summarise(cntry\_lrscale\_wave = mean(lrscale\_m, na.rm = TRUE))  
  
# I merge the 'cntry\_lrscale\_wave' values into the main data frame with a left  
# join, keeping all observations from the cleaned ESS Wave 4 data set  
  
d4 <- merge(x = d4, y = aux, by = "cntry", all.x = TRUE)  
  
unique(d4$cntry\_lrscale\_wave) # Checking all the unique country codes to assess whether 22 new values were saved into the 'cntry\_lrscale\_wave' column

## [1] -0.16788737 -0.18031888 -0.02265675 0.32312761 -0.55580933 0.21093072  
## [7] 0.09455244 -0.55312001 0.61955459 -0.30284089 -0.09146377 0.15820651  
## [13] 0.45911865 0.03379293 0.15857233 0.64862523 0.05538201 0.64849501  
## [19] -0.27157490 0.02225750 -0.47040588 -0.37226895

rm(aux) # I remove the auxiliary data set to keep the environment clean  
  
# 4)  
  
d4$female <- car::recode(d4$gndr, "1 = 0; 2 = 1") # I recode 1 (male) as 0, and 2 (female) as 1 to get a dummy variable for gender  
  
# I drop the 'gndr' variable since it now provides redundant information only.  
d4 <- subset(d4, select = -gndr)  
  
# I now subtract the means of the three remaining Level-1 variables that I want  
# to grand mean-centre, specifying that missing values must be removed before  
# running the operation.  
  
d4$agea\_m <- d4$agea - mean(d4$agea, na.rm = TRUE)  
d4$eduyrs\_m <- d4$eduyrs - mean(d4$eduyrs, na.rm = TRUE)  
  
# 5)  
  
# I create a vector with the codes in the desired country order  
cntry\_order <- c("BE", "BG", "HR", "CY", "CZ", "DK", "EE", "FI", "FR", "DE", "HU",  
 "IE", "IT", "LV", "LT", "NL", "PL", "PT", "SK", "SI", "ES", "SE", "GB")  
  
# I then generate a new column 'cntry\_num' and assign numeric identifiers based  
# on the order I specified  
  
d4$cntry\_num <- match(d4$cntry, cntry\_order)  
  
unique(d4$cntry\_num) # A non-ordered series from 1 to 23 should be printed, with the number 13 (Italy) missing.

## [1] 1 2 4 5 10 6 7 21 8 9 23 3 11 12 15 14 16 17 18 22 20 19

# I now generate the 'cntry\_wave' Level-2 identifier. Since I am working on the  
# 4th Wave of the ESS, this must be 400 + the 'cntry\_num' identifier.  
  
d4$cntry\_wave = 400 + d4$cntry\_num  
  
unique(d4$cntry\_wave) # A non-ordered series from 401 to 423 should be printed, with the number 413 (Italy) missing.

## [1] 401 402 404 405 410 406 407 421 408 409 423 403 411 412 415 414 416 417 418  
## [20] 422 420 419

# I now generate the 'cntry\_wave\_char' Level-2 identifier. Since I am working  
# on the 4th Wave of the ESS, this must be the country code identifier + '2008'  
  
d4$cntry\_wave\_char <- paste(d4$cntry, "2008", sep = " ")  
  
unique(d4$cntry\_wave\_char) # A series of 22 labels should be printed, with the label for Italy missing.

## [1] "BE 2008" "BG 2008" "CY 2008" "CZ 2008" "DE 2008" "DK 2008" "EE 2008"  
## [8] "ES 2008" "FI 2008" "FR 2008" "GB 2008" "HR 2008" "HU 2008" "IE 2008"  
## [15] "LT 2008" "LV 2008" "NL 2008" "PL 2008" "PT 2008" "SE 2008" "SI 2008"  
## [22] "SK 2008"

## ESS Wave 6 (2012) data set

I must take care of the following sub-tasks:

1. Drop all the Level-1 variables that do not concern my analysis. This means that I will be keeping *idno* - i.e., the respondent’s (Level-1) unique identifier - *cntry* - i.e., the country (Level-2) unique identifier - *euftf* - i.e., the dependent variable, the respondent’s degree of belief in European unification - *gndr* - i.e., the respondent’s gender - *agea* - i.e., the respondent’s age in years - *eduyrs* - i.e., the number of years of full-time education the respondent has completed - and *lrscale* - i.e., the main independent variable, the respondent’s placement on the left-to-right scale.
2. Exclude the countries that are not European Union members. Remember that in this wave there is missing data at Level-2 for Croatia, Latvia, and Romania.
3. Grand-mean centre *lrscale* and generate a new aggregate variable that indicates each country’s average placement on the left-to-right scale in the given wave - i.e., *cntry\_lrscale\_wave*.
4. Recode the *gndr* variable to a dummy *female* variable, which takes the value of 1 when the respondent is female, and grand-mean centre the rest of the Level-1 confounders - i.e., *agea*, and *eduyrs*.
5. Generate the Level-2 (country \* wave) unique identifier *cntry\_wave*, a numeric value that derives from the wave number (4, 6, or 9) - i.e., *wave* - multiplied by 100, added to the numeric transformation of the country identifier. Also, create a character version of this identifier to foster its visualization.

# Setting the working directory  
setwd("C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper")  
  
# Checking if the working directory was appropriately set  
getwd()

## [1] "C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper"

# Loading my data set of interest: ESS Round 6, 2012, N = 54673)  
ess6 <- haven::read\_stata("ESS6e02\_5.dta")  
  
# 1)  
  
# Selecting the variables of interest by conditioning on the specified column  
# labels, which I need not re-set since they are the same for all ESS Waves  
d6 <- ess6[, vars]  
  
# 2)  
  
unique(d6$cntry) # Checking all the unique country codes to see the identifiers that I must keep in the dataset to exclude all non-EU members and Greece

## [1] "AL" "BE" "BG" "CH" "CY" "CZ" "DE" "DK" "EE" "ES" "FI" "FR" "GB" "HU" "IE"  
## [16] "IL" "IS" "IT" "LT" "NL" "NO" "PL" "PT" "RU" "SE" "SI" "SK" "UA" "XK"

# I must keep Belgium (BE), Bulgaria (BG), Cyprus (CY), Czech Republic (CZ),  
# Denmark (DK), Estonia (EE), Finland (FI), France (FR), Germany (DE), Hungary  
# (HU), Ireland (IE), Italy (IT), Lithuania (LT), The Netherlands (NL), Poland  
# (PL), Portugal (PT), Slovakia (SK), Slovenia (SI), Spain (ES), Sweden (SE),  
# and the United Kingdom (GB).  
  
d6 <- subset(d6, d6$cntry == "BE" | d6$cntry == "BG" | d6$cntry == "CY" | d6$cntry ==  
 "CZ" | d6$cntry == "DK" | d6$cntry == "EE" | d6$cntry == "FI" | d6$cntry == "FR" |  
 d6$cntry == "DE" | d6$cntry == "HU" | d6$cntry == "IE" | d6$cntry == "IT" | d6$cntry ==  
 "LT" | d6$cntry == "NL" | d6$cntry == "PL" | d6$cntry == "PT" | d6$cntry == "SK" |  
 d6$cntry == "SI" | d6$cntry == "ES" | d6$cntry == "SE" | d6$cntry == "GB") # I am left with N = 41138 respondents  
  
unique(d6$cntry) # Checking all the unique country codes to assess whether this cleaning step worked smoothly

## [1] "BE" "BG" "CY" "CZ" "DE" "DK" "EE" "ES" "FI" "FR" "GB" "HU" "IE" "IT" "LT"  
## [16] "NL" "PL" "PT" "SE" "SI" "SK"

rm(ess6) # I remove the now unnecessary data set to keep the environment clean  
  
# 3)  
  
# I mean-centre the 'lrscale' variable by subtracting its overall mean,  
# specifying that missing values must be excluded before computation.  
  
d6$lrscale\_m <- d6$lrscale - mean(d6$lrscale, na.rm = TRUE)  
  
# I group the observations by country, and then calculate the Level-2 (country,  
# time-variant) mean of 'lrscale\_m', specifying that missing values must be  
# removed before running the operation. I save the computed values in the  
# 'cntry\_lrscale\_wave' column, in a new auxiliary dataset  
  
aux <- d6 %>%  
 group\_by(cntry) %>%  
 summarise(cntry\_lrscale\_wave = mean(lrscale\_m, na.rm = TRUE))  
  
# I merge the 'cntry\_lrscale\_wave' values into the main data frame with a left  
# join, keeping all observations from the cleaned ESS Wave 6 data set  
  
d6 <- merge(x = d6, y = aux, by = "cntry", all.x = TRUE)  
  
unique(d6$cntry\_lrscale\_wave) # Checking all the unique country codes to assess whether 21 new values were saved into the 'cntry\_lrscale\_wave' column

## [1] -0.065582810 -0.005353966 0.026041579 0.092585940 -0.595783527  
## [6] 0.229975472 0.337053059 -0.568402487 0.553064667 -0.120048323  
## [11] -0.024556610 0.300704704 0.058032038 -0.441111020 -0.179338492  
## [16] 0.203530041 0.565934479 -0.177853281 0.204625638 -0.394558511  
## [21] -0.198607496

rm(aux) # I remove the auxiliary data set to keep the environment clean  
  
# 4)  
  
d6$female <- car::recode(d6$gndr, "1 = 0; 2 = 1") # I recode 1 (male) as 0, and 2 (female) as 1 to get a dummy variable for gender  
  
# I drop the 'gndr' variable since it now provides redundant information only.  
d6 <- subset(d6, select = -gndr)  
  
# I now subtract the means of the three remaining Level-1 variables that I want  
# to grand mean-centre, specifying that missing values must be removed before  
# running the operation.  
  
d6$agea\_m <- d6$agea - mean(d6$agea, na.rm = TRUE)  
d6$eduyrs\_m <- d6$eduyrs - mean(d6$eduyrs, na.rm = TRUE)  
  
# 5)  
  
# I need not create a new country ordering, since I kept the object from the  
# last chunk. I now generate a new column 'cntry\_num' and assign numeric  
# identifiers based on the order I specified  
  
d6$cntry\_num <- match(d6$cntry, cntry\_order)  
  
unique(d6$cntry\_num) # A non-ordered series from 1 to 23 should be printed, with the numbers 3 (Croatia), and 14 (Latvia) missing.

## [1] 1 2 4 5 10 6 7 21 8 9 23 11 12 13 15 16 17 18 22 20 19

# I now generate the 'cntry\_wave' Level-2 identifier. Since I am working on the  
# 6th Wave of the ESS, this must be 600 + the 'cntry\_num' identifier.  
  
d6$cntry\_wave = 600 + d6$cntry\_num  
  
unique(d6$cntry\_wave) # A non-ordered series from 601 to 623 should be printed, with the numbers 603 (Croatia), and 614 (Latvia) missing.

## [1] 601 602 604 605 610 606 607 621 608 609 623 611 612 613 615 616 617 618 622  
## [20] 620 619

# I now generate the 'cntry\_wave\_char' Level-2 identifier. Since I am working  
# on the 6th Wave of the ESS, this must be the country code identifier + '2012'  
  
d6$cntry\_wave\_char <- paste(d6$cntry, "2012", sep = " ")  
  
unique(d6$cntry\_wave\_char) # A series of 21 labels should be printed, with the labels for Croatia, and Latvia missing.

## [1] "BE 2012" "BG 2012" "CY 2012" "CZ 2012" "DE 2012" "DK 2012" "EE 2012"  
## [8] "ES 2012" "FI 2012" "FR 2012" "GB 2012" "HU 2012" "IE 2012" "IT 2012"  
## [15] "LT 2012" "NL 2012" "PL 2012" "PT 2012" "SE 2012" "SI 2012" "SK 2012"

## ESS Wave 9 (2018) data set

I must take care of the following sub-tasks:

1. Drop all the Level-1 variables that do not concern my analysis. This means that I will be keeping *idno* - i.e., the respondent’s (Level-1) unique identifier - *cntry* - i.e., the country (Level-2) unique identifier - *euftf* - i.e., the dependent variable, the respondent’s degree of belief in European unification - *gndr* - i.e., the respondent’s gender - *agea* - i.e., the respondent’s age in years - *eduyrs* - i.e., the number of years of full-time education the respondent has completed - and *lrscale* - i.e., the main independent variable, the respondent’s placement on the left-to-right scale.
2. Exclude the countries that do not have sufficient time-variant data (Level-2) to be included - i.e., Austria - or are not European Union members.
3. Grand-mean centre *lrscale* and generate a new aggregate variable that indicates each country’s average placement on the left-to-right scale in the given wave - i.e., *cntry\_lrscale\_wave*.
4. Recode the *gndr* variable to a dummy *female* variable, which takes the value of 1 when the respondent is female, and grand-mean centre the rest of the Level-1 confounders - i.e., *agea*, and *eduyrs*.
5. Generate the Level-2 (country \* wave) unique identifier *cntry\_wave*, a numeric value that derives from the wave number (4, 6, or 9) - i.e., *wave* - multiplied by 100, added to the numeric transformation of the country identifier. Also, create a character version of this identifier to foster its visualization.

# Setting the working directory  
setwd("C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper")  
  
# Checking if the working directory was appropriately set  
getwd()

## [1] "C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper"

# Loading my data set of interest: ESS Round 9, 2018, N = 49519)  
ess9 <- haven::read\_stata("ESS9e03\_1.dta")  
  
# 1)  
  
# Selecting the variables of interest by conditioning on the specified column  
# labels, which I need not re-set since they are the same for all ESS Waves  
d9 <- ess9[, vars]  
  
# 2)  
  
# I must keep Belgium (BE), Bulgaria (BG), Croatia (HR), Cyprus (CY), Czech  
# Republic (CZ), Denmark (DK), Estonia (EE), Finland (FI), France (FR), Germany  
# (DE), Hungary (HU), Ireland (IE), Italy (IT), Latvia (LV), Lithuania (LT),  
# The Netherlands (NL), Poland (PL), Portugal (PT), Slovakia (SK), Slovenia  
# (SI), Spain (ES), Sweden (SE), and the United Kingdom (GB).  
  
d9 <- subset(d9, d9$cntry == "BE" | d9$cntry == "BG" | d9$cntry == "HR" | d9$cntry ==  
 "CY" | d9$cntry == "CZ" | d9$cntry == "DK" | d9$cntry == "EE" | d9$cntry == "FI" |  
 d9$cntry == "FR" | d9$cntry == "DE" | d9$cntry == "HU" | d9$cntry == "IE" | d9$cntry ==  
 "IT" | d9$cntry == "LV" | d9$cntry == "LT" | d9$cntry == "NL" | d9$cntry == "PL" |  
 d9$cntry == "PT" | d9$cntry == "SK" | d9$cntry == "SI" | d9$cntry == "ES" | d9$cntry ==  
 "SE" | d9$cntry == "GB") # I am left with N = 39968 respondents  
  
unique(d9$cntry) # Checking all the unique country codes to assess whether this cleaning step worked smoothly

## [1] "BE" "BG" "CY" "CZ" "DE" "DK" "EE" "ES" "FI" "FR" "GB" "HR" "HU" "IE" "IT"  
## [16] "LT" "LV" "NL" "PL" "PT" "SE" "SI" "SK"

rm(ess9) # I remove the now unnecessary data set to keep the environment clean  
  
# 3)  
  
# I mean-centre the 'lrscale' variable by subtracting its overall mean,  
# specifying that missing values must be excluded before computation.  
  
d9$lrscale\_m <- d9$lrscale - mean(d9$lrscale, na.rm = TRUE)  
  
# I group the observations by country, and then calculate the Level-2 (country,  
# time-variant) mean of 'lrscale\_m', specifying that missing values must be  
# removed before running the operation. I save the computed values in the  
# 'cntry\_lrscale\_wave' column, in a new auxiliary dataset  
  
aux <- d9 %>%  
 group\_by(cntry) %>%  
 summarise(cntry\_lrscale\_wave = mean(lrscale\_m, na.rm = TRUE))  
  
# I merge the 'cntry\_lrscale\_wave' values into the main data frame with a left  
# join, keeping all observations from the cleaned ESS Wave 9 data set  
  
d9 <- merge(x = d9, y = aux, by = "cntry", all.x = TRUE)  
  
unique(d9$cntry\_lrscale\_wave) # Checking all the unique country codes to assess whether 23 new values were saved into the 'cntry\_lrscale\_wave' column

## [1] -0.12436747 0.25361861 0.29983644 0.37342487 -0.69920789 -0.05262945  
## [7] 0.28403864 -0.65525545 0.55267325 -0.20232693 -0.14270058 -0.29764837  
## [13] 0.26380032 -0.05001418 0.17727846 -0.17135009 0.70041064 0.03750715  
## [19] 0.71145433 -0.27661815 0.08136702 -0.21360099 -0.22308118

rm(aux) # I remove the auxiliary data set to keep the environment clean  
  
# 4)  
  
d9$female <- car::recode(d9$gndr, "1 = 0; 2 = 1") # I recode 1 (male) as 0, and 2 (female) as 1 to get a dummy variable for gender  
  
# I drop the 'gndr' variable since it now provides redundant information only.  
d9 <- subset(d9, select = -gndr)  
  
# I now subtract the means of the three remaining Level-1 variables that I want  
# to grand mean-centre, specifying that missing values must be removed before  
# running the operation.  
  
d9$agea\_m <- d9$agea - mean(d9$agea, na.rm = TRUE)  
d9$eduyrs\_m <- d9$eduyrs - mean(d9$eduyrs, na.rm = TRUE)  
  
# 5)  
  
# I need not create a new country ordering, since I kept the object from the  
# last chunk. I now generate a new column 'cntry\_num' and assign numeric  
# identifiers based on the order I specified  
  
d9$cntry\_num <- match(d9$cntry, cntry\_order)  
  
unique(d9$cntry\_num) # A non-ordered series from 1 to 23 should be printed.

## [1] 1 2 4 5 10 6 7 21 8 9 23 3 11 12 13 15 14 16 17 18 22 20 19

# I now generate the 'cntry\_wave' Level-2 identifier. Since I am working on the  
# 9th Wave of the ESS, this must be 900 + the 'cntry\_num' identifier.  
  
d9$cntry\_wave = 900 + d9$cntry\_num  
  
unique(d9$cntry\_wave) # A non-ordered series from 601 to 623 should be printed.

## [1] 901 902 904 905 910 906 907 921 908 909 923 903 911 912 913 915 914 916 917  
## [20] 918 922 920 919

# I now generate the 'cntry\_wave\_char' Level-2 identifier. Since I am working  
# on the 9th Wave of the ESS, this must be the country code identifier + '2018'  
  
d9$cntry\_wave\_char <- paste(d9$cntry, "2018", sep = " ")  
  
unique(d9$cntry\_wave\_char) # A series of 23 labels should be printed.

## [1] "BE 2018" "BG 2018" "CY 2018" "CZ 2018" "DE 2018" "DK 2018" "EE 2018"  
## [8] "ES 2018" "FI 2018" "FR 2018" "GB 2018" "HR 2018" "HU 2018" "IE 2018"  
## [15] "IT 2018" "LT 2018" "LV 2018" "NL 2018" "PL 2018" "PT 2018" "SE 2018"  
## [22] "SI 2018" "SK 2018"

## GDP per capita (PPP adjusted, in $) data set

I must take care of the following sub-tasks:

1. Drop all the Level-2 variables that do not concern my analysis. This means that I will only keep *Country.Name* - i.e., the country’s name - *Country.Code* - i.e., the country identifier - and *X2008*, *X2012*, and *X2018* - i.e., the data for GDP per capita (PPP adjusted, in $) that corresponds to the years in which the respective ESS Waves were published.
2. Drop all the countries that do not concern my analysis. This means that I will only keep the 23 European countries of substantive interest: Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Hungary, Ireland, Italy, Latvia, Lithuania, Netherlands, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, and the United Kingdom.
3. Recode all the *Country.Code* values to match them with the country codes - i.e., *cntry* - in the main ESS data sets.
4. Run a left join, merging the GDP per capita (PPP adjusted, in $) data for the corresponding survey year (*X2008*, *X2012*, or *X2018*) into the main ESS data sets. This can be safely done by merging on the recoded *Country.Code* values, because the main data sets are still separate.
5. Generate the *gdp\_z\_wave* variable by standardizing the GDP per capita (PPP adjusted, in $) data in each main ESS data set.

# Setting the working directory  
setwd("C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper")  
  
# Checking if the working directory was appropriately set  
getwd()

## [1] "C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper"

# # Loading my data set of interest (GDP per capita, PPP adjusted, N = 266)  
gdp = read.csv("gdp\_ppp.csv", sep = ",")  
  
# 1)  
  
# Specifying the variables I will be keeping for my analysis  
vars <- c("Country.Name", "Country.Code", "X2008", "X2012", "X2018")  
  
# Selecting the variables of interest by conditioning on the specified column  
# labels  
gdp <- gdp[, vars]  
  
# 2) Drop all the countries that do not concern my analysis. This means that I  
# will only keep the 23 European countries of substantive interest: Belgium,  
# Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France,  
# Germany, Hungary, Ireland, Italy, Latvia, Lithuania, Netherlands, Poland,  
# Portugal, Slovakia, Slovenia, Spain, Sweden, and the United Kingdom.  
  
gdp <- subset(gdp, gdp$Country.Name == "Belgium" | gdp$Country.Name == "Bulgaria" |  
 gdp$Country.Name == "Croatia" | gdp$Country.Name == "Cyprus" | gdp$Country.Name ==  
 "Czechia" | gdp$Country.Name == "Denmark" | gdp$Country.Name == "Estonia" | gdp$Country.Name ==  
 "Finland" | gdp$Country.Name == "France" | gdp$Country.Name == "Germany" | gdp$Country.Name ==  
 "Hungary" | gdp$Country.Name == "Ireland" | gdp$Country.Name == "Italy" | gdp$Country.Name ==  
 "Latvia" | gdp$Country.Name == "Lithuania" | gdp$Country.Name == "Netherlands" |  
 gdp$Country.Name == "Poland" | gdp$Country.Name == "Portugal" | gdp$Country.Name ==  
 "Slovak Republic" | gdp$Country.Name == "Slovenia" | gdp$Country.Name == "Spain" |  
 gdp$Country.Name == "Sweden" | gdp$Country.Name == "United Kingdom")  
  
# 3)  
  
# I get all the unique country codes in the main ESS Wave 9 data set, to aid  
# the recoding task  
unique(d9$cntry)

## [1] "BE" "BG" "CY" "CZ" "DE" "DK" "EE" "ES" "FI" "FR" "GB" "HR" "HU" "IE" "IT"  
## [16] "LT" "LV" "NL" "PL" "PT" "SE" "SI" "SK"

# I get all the unique country codes GDP per capita (PPP adjusted, in $) data  
# set, to aid the recoding task  
unique(gdp$Country.Code)

## [1] "BEL" "BGR" "CYP" "CZE" "DEU" "DNK" "ESP" "EST" "FIN" "FRA" "GBR" "HRV"  
## [13] "HUN" "IRL" "ITA" "LTU" "LVA" "NLD" "POL" "PRT" "SVK" "SVN" "SWE"

# The country codes in the ESS Wave 9 data set versus the ones in the World  
# Bank data set are: Belgium | 'BE' versus 'BEL' Bulgaria | 'BG' versus 'BGR'  
# Croatia | 'HR' versus 'HRV' Cyprus | 'CY' versus 'CYP' Czech Republic | 'CZ'  
# versus 'CZE' Denmark | 'DK' versus 'DNK' Estonia | 'EE' versus 'EST' Finland  
# | 'FI' versus 'FIN' France | 'FR' versus 'FRA' Germany | 'DE' versus 'DEU'  
# Hungary | 'HU' versus 'HUN' Ireland | 'IE' versus 'IRL' Italy | 'IT' versus  
# 'ITA'. Latvia | 'LV' versus 'LVA' Lithuania | 'LT' versus 'LTU' The  
# Netherlands | 'NL' versus 'NLD' Poland | 'PL' versus 'POL' Portugal | 'PT'  
# versus 'PRT' Slovakia | 'SK' versus 'SVK' Slovenia | 'SI' versus 'SVN' Spain  
# | 'ES' versus 'ESP' Sweden | 'SE' versus 'SWE' United Kingdom | 'GB' versus  
# 'GBR'  
  
gdp$cntry <- car::recode(gdp$Country.Code, "'BEL' = 'BE'; 'BGR' = 'BG'; 'HRV' = 'HR'; 'CYP' = 'CY'; 'CZE' = 'CZ'; 'DNK' = 'DK'; 'EST' = 'EE'; 'FIN' = 'FI'; 'FRA' = 'FR'; 'DEU' = 'DE'; 'HUN' = 'HU'; 'IRL' = 'IE'; 'ITA' = 'IT'; 'LVA' = 'LV'; 'LTU' = 'LT'; 'NLD' = 'NL'; 'POL' = 'PL'; 'PRT' = 'PT'; 'SVK' = 'SK'; 'SVN' = 'SI'; 'ESP' = 'ES'; 'SWE' = 'SE'; 'GBR' = 'GB'")  
  
unique(gdp$cntry) # I check whether the country codes were correctly recoded

## [1] "BE" "BG" "CY" "CZ" "DE" "DK" "ES" "EE" "FI" "FR" "GB" "HR" "HU" "IE" "IT"  
## [16] "LT" "LV" "NL" "PL" "PT" "SK" "SI" "SE"

# I drop the 'Country.Name' and 'Country.Code' variables since they do not  
# contain useful information to add to the main ESS data sets  
  
# Specifying the variables I will be keeping for my analysis  
vars <- c("cntry", "X2008", "X2012", "X2018")  
  
# Selecting the variables of interest by conditioning on the specified column  
# labels  
gdp <- gdp[, vars]  
  
# I now must transform the data types of the 'cntry' columns in the main data  
# sets from labelled to character, or else the left\_join command from the dplyr  
# package will not work  
d4$cntry <- as.character(d4$cntry)  
d6$cntry <- as.character(d6$cntry)  
d9$cntry <- as.character(d9$cntry)  
  
# I merge the 'X2008' values into the corresponding main data frame with a left  
# join, keeping all observations from the cleaned ESS Wave 4 data set  
d4 <- left\_join(d4, gdp[, c("cntry", "X2008")], by = "cntry")  
  
unique(d4$X2008) # I check whether there are 22 unique values for GDP per capita (PPP, in $), since there are 22 countries in the cleaned ESS Wave 4 data set

## [1] 37883.33 14346.90 34830.27 27974.49 37802.39 41278.33 22802.34 33263.27  
## [9] 40083.70 35092.69 36863.58 20978.50 20727.89 44169.09 20721.01 19573.16  
## [17] 46420.20 18303.35 26665.84 42158.30 29604.29 23725.86

# I merge the 'X2012' values into the corresponding main data frame with a left  
# join, keeping all observations from the cleaned ESS Wave 6 data set  
d6 <- left\_join(d6, gdp[, c("cntry", "X2012")], by = "cntry")  
  
unique(d6$X2012) # I check whether there are 21 unique values for GDP per capita (PPP, in $), since there are 21 countries in the cleaned ESS Wave 6 data set

## [1] 42290.76 16327.86 31923.94 29254.73 43359.61 44803.97 25996.05 31720.12  
## [9] 40873.10 37677.73 38514.03 23275.23 46361.46 36486.30 24703.69 47272.10  
## [17] 23589.15 26438.02 45432.43 29042.82 26888.35

# I merge the 'X2018' values into the corresponding main data frame with a left  
# join, keeping all observations from the cleaned ESS Wave 9 data set  
d9 <- left\_join(d9, gdp[, c("cntry", "X2018")], by = "cntry")  
  
unique(d9$X2018) # I check whether there are 23 unique values for GDP per capita (PPP, in $), since there are 23 countries in the cleaned ESS Wave 9 data set

## [1] 52535.23 23016.07 40676.62 41135.49 55235.37 57482.96 36249.36 40696.39  
## [9] 49580.61 46537.47 47573.49 28909.41 31913.07 84555.63 43036.24 36376.19  
## [17] 30877.04 57826.63 32027.60 34928.54 53521.63 38961.54 31214.59

colnames(d4)[colnames(d4) == "X2008"] <- "gdp\_wave" # Renaming the newly merged 'X2008' variable with a meaningful label (gdp\_wave)  
  
colnames(d6)[colnames(d6) == "X2012"] <- "gdp\_wave" # Renaming the newly merged 'X2012' variable with a meaningful label (gdp\_wave)  
  
colnames(d9)[colnames(d9) == "X2018"] <- "gdp\_wave" # Renaming the newly merged 'X2018' variable with a meaningful label (gdp\_wave)  
  
rm(gdp) # I remove the GDP data set to keep the environment clean  
  
# 5)  
  
d4$gdp\_z\_wave <- as.vector(scale(d4$gdp\_wave)) # Wave 4 (2008)  
d6$gdp\_z\_wave <- as.vector(scale(d6$gdp\_wave)) # Wave 6 (2012)  
d9$gdp\_z\_wave <- as.vector(scale(d9$gdp\_wave)) # Wave 9 (2018)

## Corruption Perception Index data set

I must take care of the following sub-tasks:

1. Drop all the Level-2 variables that do not concern my analysis. This means that I will only keep *Jurisdiction* - i.e., the country’s name - *X2008*, and *X2012* - i.e., the data for the Corruption Perception Index that corresponds to the years in which the respective ESS Waves were published.
2. Drop all the countries that do not concern my analysis. This means that I will only keep the 23 European countries of substantive interest: Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Hungary, Ireland, Italy, Latvia, Lithuania, Netherlands, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, and the United Kingdom.
3. Create a new *cntry* column with country codes that match the main ESS data sets.
4. Since the 2018 data was never registered in a .csv format, I must unfortunately code it by hand in a new *X2018* column. The official data from Transparency International is only provided in a machine-unfriendly .xlsx format, so it is just quicker to directly assign the 23 separate values inside this Corruption Perception Index data set.
5. Since the CPI’s scale was changed from 1-to-10 decimals to 1-to-100 integers starting from 2012, I must recode the values contained in the *X2008* column by multiplying them by 10.
6. Run a left join, merging the Corruption Perception Index data for the corresponding survey year (*X2008*, *X2012*, or *X2018*) into the main ESS data sets. This can be safely done by merging on the *cntry* values, because the main data sets are still separate.
7. Generate the *cpi\_wave* variable by grand mean-centering the Corruption Perception Index values within each main ESS data set.

# Setting the working directory  
setwd("C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper")  
  
# Checking if the working directory was appropriately set  
getwd()

## [1] "C:/Users/Mattia aka Mario/Desktop/UvA/First Year/Fixed and Random Effects Modelling (T. van der Meer)/Final Paper"

# # Loading my data set of interest (GDP per capita, PPP adjusted, N = 266)  
cpi = read.csv("cpi.csv", sep = ",")  
  
# 1)  
  
# Specifying the variables I will be keeping for my analysis  
vars <- c("Jurisdiction", "X2008", "X2012")  
  
# Selecting the variables of interest by conditioning on the specified column  
# labels  
cpi <- cpi[, vars]  
  
# 2) Drop all the countries that do not concern my analysis. This means that I  
# will only keep the 23 European countries of substantive interest: Belgium,  
# Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France,  
# Germany, Hungary, Ireland, Italy, Latvia, Lithuania, Netherlands, Poland,  
# Portugal, Slovakia, Slovenia, Spain, Sweden, and the United Kingdom.  
  
cpi <- subset(cpi, cpi$Jurisdiction == "Belgium" | cpi$Jurisdiction == "Bulgaria" |  
 cpi$Jurisdiction == "Croatia" | cpi$Jurisdiction == "Cyprus" | cpi$Jurisdiction ==  
 "Czech Republic" | cpi$Jurisdiction == "Denmark" | cpi$Jurisdiction == "Estonia" |  
 cpi$Jurisdiction == "Finland" | cpi$Jurisdiction == "France" | cpi$Jurisdiction ==  
 "Germany" | cpi$Jurisdiction == "Hungary" | cpi$Jurisdiction == "Ireland" | cpi$Jurisdiction ==  
 "Italy" | cpi$Jurisdiction == "Latvia" | cpi$Jurisdiction == "Lithuania" | cpi$Jurisdiction ==  
 "Netherlands" | cpi$Jurisdiction == "Poland" | cpi$Jurisdiction == "Portugal" |  
 cpi$Jurisdiction == "Slovakia" | cpi$Jurisdiction == "Slovenia" | cpi$Jurisdiction ==  
 "Spain" | cpi$Jurisdiction == "Sweden" | cpi$Jurisdiction == "United Kingdom")  
  
unique(cpi$cntry) # Checking if all 23 country codes were properly set

## NULL

# 3)  
  
cpi$cntry <- car::recode(cpi$Jurisdiction, "'Belgium' = 'BE'; 'Bulgaria' = 'BG'; 'Croatia' = 'HR'; 'Cyprus' = 'CY'; 'Czech Republic' = 'CZ'; 'Denmark' = 'DK'; 'Estonia' = 'EE'; 'Finland' = 'FI'; 'France' = 'FR'; 'Germany' = 'DE'; 'Hungary' = 'HU'; 'Ireland' = 'IE'; 'Italy' = 'IT'; 'Latvia' = 'LV'; 'Lithuania' = 'LT'; 'Netherlands' = 'NL'; 'Poland' = 'PL'; 'Portugal' = 'PT'; 'Slovakia' = 'SK'; 'Slovenia' = 'SI'; 'Spain' = 'ES'; 'Sweden' = 'SE'; 'United Kingdom' = 'GB'")  
  
# 4)  
  
cpi$X2018 <- car::recode(cpi$Jurisdiction, "'Belgium' = 75; 'Bulgaria' = 42; 'Croatia' = 48; 'Cyprus' = 59; 'Czech Republic' = 59; 'Denmark' = 88; 'Estonia' = 73; 'Finland' = 85; 'France' = 72; 'Germany' = 80; 'Hungary' = 46; 'Ireland' = 73; 'Italy' = 52; 'Latvia' = 58; 'Lithuania' = 59; 'Netherlands' = 82; 'Poland' = 60; 'Portugal' = 64; 'Slovakia' = 50; 'Slovenia' = 60; 'Spain' = 58; 'Sweden' = 85; 'United Kingdom' = 80")  
  
# 5)  
  
cpi$X2008 <- as.numeric(cpi$X2008) \* 10 # Converting the data for 2008 to the numeric type and multiplying its scale by 10  
cpi$X2012 <- as.numeric(cpi$X2012) # Converting the data for 2012 to the numeric type  
  
# 6)  
  
# I merge the 'X2008' values into the corresponding main data frame with a left  
# join, keeping all observations from the cleaned ESS Wave 4 data set  
d4 <- left\_join(d4, cpi[, c("cntry", "X2008")], by = "cntry")  
  
# I merge the 'X2012' values into the corresponding main data frame with a left  
# join, keeping all observations from the cleaned ESS Wave 6 data set  
d6 <- left\_join(d6, cpi[, c("cntry", "X2012")], by = "cntry")  
  
# I merge the 'X2018' values into the corresponding main data frame with a left  
# join, keeping all observations from the cleaned ESS Wave 9 data set  
d9 <- left\_join(d9, cpi[, c("cntry", "X2018")], by = "cntry")  
  
# Checking for missing values in the newly merged columns to see whether this  
# cleaning step went smoothly  
  
print(sum(is.na(d4$X2008))) # ESS Wave 4 (2008)

## [1] 0

print(sum(is.na(d6$X2012))) # ESS Wave 6 (2012)

## [1] 0

print(sum(is.na(d9$X2018))) # ESS Wave 9 (2018)

## [1] 0

# 7)  
  
colnames(d4)[colnames(d4) == "X2008"] <- "cpi\_wave" # Renaming the newly merged 'X2008' variable with a meaningful label (cpi\_wave)  
  
colnames(d6)[colnames(d6) == "X2012"] <- "cpi\_wave" # Renaming the newly merged 'X2012' variable with a meaningful label (cpi\_wave)  
  
colnames(d9)[colnames(d9) == "X2018"] <- "cpi\_wave" # Renaming the newly merged 'X2018' variable with a meaningful label (cpi\_wave)  
  
rm(cpi) # I remove the GDP data set to keep the environment clean  
  
# I grand mean-centre the Level-2 (time-variant) CPIs in all three main data  
# sets  
  
d4$cpi\_wave <- d4$cpi\_wave - mean(d4$cpi\_wave, na.rm = TRUE) # Wave 4 (2008)  
d6$cpi\_wave <- d6$cpi\_wave - mean(d6$cpi\_wave, na.rm = TRUE) # Wave 6 (2012)  
d9$cpi\_wave <- d9$cpi\_wave - mean(d9$cpi\_wave, na.rm = TRUE) # Wave 9 (2018)

## Final steps

1. Recode and select the columns of each main ESS data set, to perfectly match their names and only keep the data needed for the analysis.
2. Attach the rows of each main ESS data set into one final data set. This is not problematic since the *cntry\_wave* unique identifiers are kept intact, and *cntry* becomes the Level-3 (time-invariant) unique identifier.
3. Generate the Level-3 (time-invariant) variables *gdp\_z*, *cntry\_lrscale*, *cntry\_stfeco*, and *cpiscale* by taking the overall country means of *gdp\_z\_wave*, *cntry\_lrscale\_wave*, and *cpi\_wave*.
4. Prepare the Level-2 data for the within-between model specification by subtracting values for *gdp\_z*, *cntry\_lrscale*, and *cpi* from *gdp\_z\_wave*, *cntry\_lrscale\_wave*, and *cpi\_wave*, respectively.
5. Generate the cross-level interactions for within-country, time-variant, and between-country, time-invariant effects, *within\_lr\_int* and *between\_lr\_int*.
6. Save the cleaned version of the final data set in the Stata (.dta) format.

# 1)  
  
# Specifying the variables I will be keeping for my analysis  
vars <- c("idno", "cntry\_wave", "cntry\_wave\_char", "cntry", "cntry\_num", "euftf",  
 "female", "agea\_m", "eduyrs\_m", "lrscale\_m", "cntry\_lrscale\_wave", "gdp\_z\_wave",  
 "cpi\_wave")  
  
# Selecting the variables of interest by conditioning on the specified column  
# labels  
d4 <- d4[, vars]  
d6 <- d6[, vars]  
d9 <- d9[, vars]  
  
# 2)  
  
d <- rbind(d4, d6) # Attaching the 2012 data to the 2008 data  
d <- rbind(d, d9) # Attaching the 2018 data  
  
# I am left with a final number N = 123011 Level-1 observations...  
  
length(unique(d$cntry\_wave)) # ...and J = 66 Level-2 (waves \* countries) clusters.

## [1] 66

length(unique(d$cntry)) # The Level-3 clusters are countries (time-invariant), and they are C = 23.

## [1] 23

d <- d[order(d$cntry\_num, d$cntry\_wave, d$idno), ] # Re-ordering the data set by the Level-3, Level-2, and Level-1 identifiers, ascending  
  
# 3) + 4)  
  
# In all these instances, I group the observation at Level-3 (country,  
# time-invariant), calculating the time-invariant mean. Then, I ungroup the  
# observations, and I subtract the latter from the Level-2 variable to get the  
# time-variant divergence from that mean.  
  
# Left-right political scale  
d <- d %>%  
 group\_by(cntry) %>%  
 mutate(cntry\_lrscale = mean(cntry\_lrscale\_wave, na.rm = TRUE)) %>%  
 ungroup()  
  
length(unique(d$cntry\_lrscale)) # I check whether the newly generated column has 23 unique values - i.e., the number of Level-3 clusters (countries, time-invariant)

## [1] 23

print(sum(is.na(d$cntry\_lrscale))) # I check whether the newly generated column has missing values

## [1] 0

d <- d %>%  
 mutate(cntry\_lrscale\_diff = cntry\_lrscale\_wave - cntry\_lrscale)  
  
length(unique(d$cntry\_lrscale\_diff)) # I check whether the newly generated column has 66 unique values - i.e., the number of Level-3 clusters (countries, time-variant)

## [1] 66

print(sum(is.na(d$cntry\_lrscale\_diff))) # I check whether the newly generated column has missing values

## [1] 0

# GDP per capita (PPP)  
d <- d %>%  
 group\_by(cntry) %>%  
 mutate(gdp\_z = mean(gdp\_z\_wave, na.rm = TRUE)) %>%  
 ungroup()  
  
length(unique(d$gdp\_z)) # I check whether the newly generated column has 23 unique values - i.e., the number of Level-3 clusters (countries, time-invariant)

## [1] 23

print(sum(is.na(d$gdp\_z))) # I check whether the newly generated column has missing values

## [1] 0

d <- d %>%  
 mutate(gdp\_z\_diff = gdp\_z\_wave - gdp\_z)  
  
length(unique(d$gdp\_z\_diff)) # I check whether the newly generated column has 66 unique values - i.e., the number of Level-3 clusters (countries, time-variant)

## [1] 66

print(sum(is.na(d$gdp\_z\_diff))) # I check whether the newly generated column has missing values

## [1] 0

# Corruption Perception Index  
d <- d %>%  
 group\_by(cntry) %>%  
 mutate(cpi = mean(cpi\_wave, na.rm = TRUE)) %>%  
 ungroup()  
  
length(unique(d$cpi)) # I check whether the newly generated column has 23 unique values - i.e., the number of Level-3 clusters (countries, time-invariant)

## [1] 23

print(sum(is.na(d$cpi))) # I check whether the newly generated column has missing values

## [1] 0

d <- d %>%  
 mutate(cpi\_diff = cpi\_wave - cpi)  
  
length(unique(d$cpi\_diff)) # I check whether the newly generated column has 66 unique values - i.e., the number of Level-3 clusters (countries, time-variant)

## [1] 66

print(sum(is.na(d$cpi\_diff))) # I check whether the newly generated column has missing values

## [1] 0

# 5)  
  
d$within\_lr\_int = d$lrscale\_m \* d$cntry\_lrscale\_diff # Interaction of individual-level effect with within-country, time-variant effect   
  
d$between\_lr\_int = d$lrscale\_m \* d$cntry\_lrscale # Interaction of individual-level effect with between-country, time-invariant effect  
  
# 6)  
  
# I now assign meaningful labels to all the unlabelled variables  
  
# Unit and cluster identifiers  
Hmisc::label(d$idno) <- "Respondent's ID number"  
Hmisc::label(d$cntry\_wave) <- "Respondent's country \* wave (Level-2) ID number"  
Hmisc::label(d$cntry\_wave\_char) <- "Country (year) identifier"  
Hmisc::label(d$cntry) <- "Respondent's country code"  
Hmisc::label(d$cntry\_num) <- "Respondent's country ID number"  
  
# Level-1 variables  
Hmisc::label(d$female) <- "Respondent is female"  
Hmisc::label(d$agea\_m) <- "Age of respondent (in years, grand mean-centred)"  
Hmisc::label(d$eduyrs\_m) <- "Years of full-time education completed (grand mean-centred)"  
Hmisc::label(d$lrscale\_m) <- "Respondent's placement on the left right scale (grand-mean centred)"  
  
# Level-2 variables  
Hmisc::label(d$cntry\_lrscale\_wave) <- "Average placement on left right scale at the country level (time-variant)"  
Hmisc::label(d$gdp\_z\_wave) <- "GDP per capita, PPP (current international $, time-variant, standardized)"  
Hmisc::label(d$cpi\_wave) <- "Corruption Perception Index (time-variant, grand mean-centred)"  
  
Hmisc::label(d$cntry\_lrscale\_diff) <- "Average placement on left right scale at the country level (time-variant divergence from time-invariant mean)"  
Hmisc::label(d$gdp\_z\_diff) <- "GDP per capita, PPP (time-variant divergence from time-invariant mean)"  
Hmisc::label(d$cpi\_diff) <- "Corruption Perception Index (time-variant divergence from time-invariant mean)"  
  
# Level-3 variables  
Hmisc::label(d$cntry\_lrscale) <- "Average placement on left right scale at the country level (time-invariant)"  
Hmisc::label(d$gdp\_z) <- "GDP per capita, PPP (current international $, time-invariant)"  
Hmisc::label(d$cpi) <- "Corruption Perception Index (time-invariant)"  
  
Hmisc::label(d$within\_lr\_int) <- "Interaction of individual-level  
political positioning with within-country, time-variant component"  
Hmisc::label(d$between\_lr\_int) <- "Interaction of individual-level  
political positioning with between-country, time-invariant component"  
  
# By looking in the ESS4 data set, I found that a respondent from the United  
# Kingdom is listed as a 123 year old man. I believe there was an error in  
# transcription because a person has never been recorded as being older than  
# 115 years old, thus I cannot be sure that the data for this individual is  
# listed correctly. For safety reasons, I delete this observation.  
  
d <- d[d$idno != 211915, ]  
  
# Saving the complete version of the final data set as a Stata data file  
# (.dta), with a number of observations of N = 123010  
  
haven::write\_dta(d, "ESS469\_complete.dta")  
  
d <- na.omit(d) # I now apply casewise deletion, by dropping all observations with missing values  
  
# Saving the version without missing values of the final data set as a Stata  
# data file (.dta), with a number of observations of N = 98664  
  
haven::write\_dta(d, "ESS469.dta")