Appendix 2. Code listing

```
################################################################################
############# Header of code file. Adding all needed libraries #################
################################################################################

#install.packages("tictoc")
library(readr)
library(dplyr) # for data cleaning
library("stringr",
lib.loc="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")
library("plotly",
lib.loc="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")
library(naniar)
library(VIM)
library(FactoMineR)
library(missMDA)
library(cluster) # for gower similarity and pam
library(Rtsne) # for t-SNE plot
library(ggplot2) # for visualization
library(tictoc)
################################################################################
######################## Import educational dataset  ###########################
################################################################################

education_UKR_2012_2018 <- read_csv("education_UKR_2012-2018.csv")
class(education_UKR_2012_2018) #check the class of data we get
education_UKR_2012_2018 = as.data.frame(education_UKR_2012_2018) #make all data to
be a data.frame
class(education_UKR_2012_2018)

################################################################################
############################ Data understanding  ###############################
################################################################################

# first I want to understand what are the unique statistical units of each of 30
columns that I can analyse.
list_of_unique_values<- list(1:numb_cols)
list_of_unique_values
for (name in 1:numb_cols) {
  unique_value<-education_UKR_2012_2018[name]%>%
    unique()
  list_of_unique_values[[name]]<- unique_value
}
View(list_of_unique_values)

# After checking the unique values we can conclude that columns "Level of
educational attainment",
# "School subject", "Teaching experience", "Type of contract", "Reference area" and
"Time period" are redundant, so we can drop them
```

```
education_UKR_2012_2018 <- select (education_UKR_2012_2018,-c(`Level of educational attainment`,
                                                              `School subject`,
                                                              `Teaching experience`,
                                                              `Type of contract`,
                                                              `Reference area`,
                                                              `Time Period`))
View(education_UKR_2012_2018)

################################################################################
######################### Deal with missing values  ############################
################################################################################
numb_rows <- nrow(education_UKR_2012_2018)
numb_complete_rows <- sum(complete.cases(education_UKR_2012_2018))
numb_rows # there are 3318 rows in the dataset
numb_complete_rows # but only 47 rows with no NA values!
1 - numb_complete_rows/numb_rows # so, if we are going to drop all missing values we'll loose 98% of information, that is not possible
numb_cols <-ncol(education_UKR_2012_2018)
numb_cols

################################################################################
#### Question 1: Discover attendance patterns for urban and rural locations ####
################################################################################

# we want get information that is assosiated with Rural and urban location grouped by wealth level
urban_rural_area <- filter(education_UKR_2012_2018, (
                                                  ((`Location` == "RUR:Rural") | (`Location` == "URB:Urban"))
                                                  #  & (`Unit of measure` == "PT:Percentage")
                                                  #  & (`Sex` !="_T:Total")
                                                  # & (`Grade` !="_T:Total")
                                                  # & (`Wealth quintile` !="_T:Total")
                                                   )
                         )
View(urban_rural_area)

##################################################
########### Males in Rural area   ################
##################################################

# Here I select data about males in Rural area
rural_area_male <-urban_rural_area %>%
  filter(`Sex`=="M:Male")%>%
  filter(`Statistical unit`=="NAR:Net attendance rate")%>%
  filter(str_detect(`Unit of measure`, "Percentage"))%>%
  filter(str_detect(`Location`, "Rural"))%>%
```

```r
  select_if(~ length(unique(.)) > 1)

# Create a general variable  - concatanation of education and wealth
rural_area_male$`Educ_wealth_male` <- paste(rural_area_male$`Level of 
education`,"_",rural_area_male$`Wealth quintile`)

# delete variables education and wealth
rural_area_male <- select(rural_area_male,-c(1,2))

# Order variables
rural_area_male <- rural_area_male[order(rural_area_male$`Educ_wealth_male`),]

#Swap data
rural_area_male <-rural_area_male[ ,c(2,1)]

##################################################
########### Females in Rural area   ################
##################################################

# Here I select data about females in Rural area
rural_area_female <-urban_rural_area %>%
  filter(`Sex`=="F:Female")%>%
  filter(`Statistical unit`=="NAR:Net attendance rate")%>%
  filter(str_detect(`Unit of measure`, "Percentage"))%>%
  filter(str_detect(`Location`, "Rural"))%>%
  select_if(~ length(unique(.)) > 1)

# Create a general variable  - concatanation of education and wealth
rural_area_female$`Educ_wealth_female` <- paste(rural_area_female$`Level of 
education`,"_",rural_area_female$`Wealth quintile`)

# delete variables education and wealth
rural_area_female <-select(rural_area_female,-c(1,2))

# Order variables
rural_area_female <- 
rural_area_female[order(rural_area_female$`Educ_wealth_female`),]

#Swap data
rural_area_female <-rural_area_female[,c(2,1)]
rural_area_female

##################################################
########### Males in Urban area   ################
##################################################

# Here I select data about males in Rural area
urban_area_male <-urban_rural_area %>%
  filter(`Sex`=="M:Male")%>%
  filter(`Statistical unit`=="NAR:Net attendance rate")%>%
  filter(str_detect(`Unit of measure`, "Percentage"))%>%
```

```
  filter(str_detect(`Location`, "Urban"))%>%
  select_if(~ length(unique(.)) > 1)


# Create a general variable  - concatanation of education and wealth
urban_area_male$`Educ_wealth_male` <- paste(urban_area_male$`Level of
education`,"_",urban_area_male$`Wealth quintile`)

# delete variables education and wealth
urban_area_male <- select(urban_area_male,-c(1,2))

# Order variables
urban_area_male <- urban_area_male[order(urban_area_male$`Educ_wealth_male`),]

#Swap data
urban_area_male <-urban_area_male[,c(2,1)]
urban_area_male

##################################################
########### Females in Urban area   ################
##################################################

# Here I select data about females in Rural area
urban_area_female <-urban_rural_area %>%
  filter(`Sex`=="F:Female")%>%
  filter(`Statistical unit`=="NAR:Net attendance rate")%>%
  filter(str_detect(`Unit of measure`, "Percentage"))%>%
  filter(str_detect(`Location`, "Urban"))%>%
  select_if(~ length(unique(.)) > 1)

# Create a general variable  - concatanation of education and wealth
urban_area_female$`Educ_wealth_female` <- paste(urban_area_female$`Level of
education`,"_",urban_area_female$`Wealth quintile`)

# delete variables education and wealth
urban_area_female <-select(urban_area_female,-c(1,2))

# Order variables
urban_area_female <-
urban_area_female[order(urban_area_female$`Educ_wealth_female`),]

#Swap data
urban_area_female <-urban_area_female[,c(2,1)]
urban_area_female

# Create new data frame that combine male and female in urban and rural data
urban_rural_area_male_female <-urban_area_male
urban_rural_area_male_female <-cbind(urban_rural_area_male_female,
rural_area_male[,2])
urban_rural_area_male_female <-cbind(urban_rural_area_male_female,
urban_area_female[,2])
```

```r
urban_rural_area_male_female <-cbind(urban_rural_area_male_female,
rural_area_female[,2])

urban_rural_area_male_female

colnames(urban_rural_area_male_female) <- c("Educ_wealth", "2013-male-urban",
"2013-male-rural","2013-female-urban" , "2013-female-rural")
View(urban_rural_area_male_female)

# Plot male  in urban and rural data
p <- plot_ly(urban_rural_area_male_female, x = ~`Educ_wealth`, y = ~`2013-male-
urban`, type = 'bar', name = 'Males in urban area') %>%
  add_trace(y = ~`2013-male-rural`, name = 'Males in rural area') %>%
  layout(xaxis= list(title = 'Wealth-Education level'), yaxis = list(title =
'Attendance rate in %'), barmode = 'group')
p

# Plot female in urban and rural data
p_fem <- plot_ly(urban_rural_area_male_female, x = ~`Educ_wealth`, y = ~`2013-
female-urban`, type = 'bar', name = 'Females in urban area', marker = list(color =
'rgb(102, 0, 255)')) %>%
  add_trace(y = ~`2013-female-rural`, name = 'Females in rural area', marker =
list(color = 'rgb(204, 0, 153)')) %>%
  layout(xaxis= list(title = 'Wealth-Education level'), yaxis = list(title =
'Attendance rate in %'), barmode = 'group')
p_fem


# Plot female in urban and rural data
p_gen <- plot_ly(urban_rural_area_male_female, x = ~`Educ_wealth`, y = ~`2013-male-
urban`, type = 'bar', name = 'Males in urban area', marker = list(color = 'rgb(51,
153, 255)')) %>%
  add_trace(y = ~`2013-male-rural`, name = 'Males in rural area', marker =
list(color = 'rgb(0, 102, 204)')) %>%
  add_trace(y = ~`2013-female-urban`, name = 'Females in urban area', marker =
list(color = 'rgb(255, 204, 204)')) %>%
  add_trace(y = ~`2013-female-rural`, name = 'Females in rural area', marker =
list(color = 'rgb(255, 102, 102)')) %>%
  layout(xaxis= list(title = 'Wealth-Education level'), yaxis = list(title =
'Attendance rate in %'), barmode = 'group')
p_gen

################ Mean check urban males #######################
#mean attendancy value for primary education of urban males for all classes
m_male_urban <- urban_rural_area_male_female[1:5,2]
mean(m_male_urban[!sapply(m_male_urban, function(x)isTRUE(all.equal(x, 0)))])

#mean attendancy value for lower-secondary education of urban males for all classes
m_male_urban <- urban_rural_area_male_female[7:11,2]
mean(m_male_urban[!sapply(m_male_urban, function(x)isTRUE(all.equal(x, 0)))])
```

```r
#mean attendancy value for upper-secondary education of urban males for all classes
m_male_urban <- urban_rural_area_male_female[13:17,2]
mean(m_male_urban[!sapply(m_male_urban, function(x)isTRUE(all.equal(x, 0)))])

################ Mean check rural males ######################
#mean attendancy value for primary education of rural males for all classes
m_male_rural <- urban_rural_area_male_female[1:5,3]
mean(m_male_rural[!sapply(m_male_rural, function(x)isTRUE(all.equal(x, 0)))])

#mean attendancy value for lower-secondary education of rural males for all classes
m_male_rural <- urban_rural_area_male_female[7:11,3]
mean(m_male_rural[!sapply(m_male_rural, function(x)isTRUE(all.equal(x, 0)))])

#mean attendancy value for upper-secondary education of rural males for all classes
m_male_rural<- urban_rural_area_male_female[13:17,3]
mean(m_male_rural[!sapply(m_male_rural, function(x)isTRUE(all.equal(x, 0)))])



################ Mean check urban females ######################
#mean attendancy value for primary education of urban females for all classes
m_fem_urban <- urban_rural_area_male_female[1:5,4]
mean(m_fem_urban[!sapply(m_fem_urban, function(x)isTRUE(all.equal(x, 0)))])

#mean attendancy value for lower-secondary education of urban females for all
classes
m_fem_urban <- urban_rural_area_male_female[7:11,4]
mean(m_fem_urban[!sapply(m_fem_urban, function(x)isTRUE(all.equal(x, 0)))])

#mean attendancy value for upper-secondary education of urban females for all
classes
m_fem_urban <- urban_rural_area_male_female[13:17,4]
mean(m_fem_urban[!sapply(m_fem_urban, function(x)isTRUE(all.equal(x, 0)))])


################ Mean check rural females ######################
#mean attendancy value for primary education of rural females for all classes
m_fem_rur <- urban_rural_area_male_female[1:5,5]
mean(m_fem_rur[!sapply(m_fem_rur, function(x)isTRUE(all.equal(x, 0)))])

#mean attendancy value for lower-secondary education of rural females for all
classes
m_fem_rur <- urban_rural_area_male_female[7:11,5]
mean(m_fem_rur[!sapply(m_fem_rur, function(x)isTRUE(all.equal(x, 0)))])

#mean attendancy value for upper-secondary education of rural females for all
classes
m_fem_rur <- urban_rural_area_male_female[13:17,5]
mean(m_fem_rur[!sapply(m_fem_rur, function(x)isTRUE(all.equal(x, 0)))])
```

```r
###############################################################################
#### Question 2: Make an analysis of teachers’ qualification over the years ####
############# depending on sex and educational institutions level. #############
###############################################################################

teachers_data <- education_UKR_2012_2018 %>%
  filter(str_detect(`Statistical unit`, "teacher")) %>%
  select_if(~ length(unique(.)) > 1)
View(teachers_data)

# check what unique statistical units we can use for data exploration
unique_value_teachers_data_units<-teachers_data[1]%>%
  unique()
unique_value_teachers_data_units

# select only the data in %
teachers_data_perc <- teachers_data%>%
  filter(str_detect(`Unit of measure`, "PT:Percentage"))
  #select_if(~ length(unique(.)) > 1)
View(teachers_data_perc)

# try to get information about what rows has less missing values
teachers_data_no_missing_vals <-
teachers_data[which.max(rowSums(!is.na(teachers_data))),]
View(teachers_data_no_missing_vals)

# we've discovered that for statistical unit : "PTR:Pupil-teacher ratio" we have
the smallest number
# of missing values, try to get all connected relevant data
pupil_teacher_ratio<- teachers_data %>%
  filter(`Statistical unit`=="PTR:Pupil-teacher ratio")%>%
  filter(`Orientation`=="_T:Total")%>%
  select_if(~ length(unique(.)) > 1)
pupil_teacher_ratio

# Order variables
pupil_teacher_ratio <- pupil_teacher_ratio[order(pupil_teacher_ratio$`Level of
education`),]
pupil_teacher_ratio

# number of missing variables
gg_miss_var(pupil_teacher_ratio)

# aggr calculates and represents the number of missing entries in each variable
# and for certain combinations of variables (which tend to be missing
simultaneously)
res<-summary(aggr(pupil_teacher_ratio, sortVar=TRUE))$combinations

matrixplot(pupil_teacher_ratio, sortby = 1)
```

```
# omit rows where more than 55% of data is missing, select data about all
institutions
pupil_teacher_ratio <-
pupil_teacher_ratio[is.na(pupil_teacher_ratio)%*%rep(1,ncol(pupil_teacher_ratio))<=
ncol(pupil_teacher_ratio)*0.55,]
summary(pupil_teacher_ratio)
pupil_teacher_ratio_all_institutions<-pupil_teacher_ratio %>%
  filter(`Type of institution`=="INST_T:All institutions")%>%
  select_if(~ length(unique(.)) > 1)
pupil_teacher_ratio_all_institutions

#change all NA values to 0 values
pupil_teacher_ratio_all_institutions[is.na(pupil_teacher_ratio_all_institutions)]
<- 0

# re-structure data
final_df <- t(pupil_teacher_ratio_all_institutions)
final_df <- final_df[-c(1), ]
years <-c(2012:2017)
final_df <-data.frame(years,final_df)
colnames(final_df) <-c("years","L0:Early childhood education", "L1:Primary
education", "L2_3:Secondary education", "L5T8:Tertiary education")
final_df

p <- plot_ly(final_df,    type = 'scatter', mode = 'markers') %>%
  add_trace(x = ~`years`,y = ~`L0:Early childhood education`, name = 'L0:Early
childhood education') %>%
  add_trace(x = ~`years`,y = ~`L1:Primary education`, name = 'L1:Primary
education') %>%
  add_trace(x = ~`years`,y = ~`L2_3:Secondary education`, name = 'L2_3:Secondary
education') %>%
  add_trace(x = ~`years`,y = ~`L5T8:Tertiary education`, name = 'L5T8:Tertiary
education') %>%
  layout(
    title = " Students teachers Ratio",
      yaxis = list(title = "Ratio"))
p


################################################################################
##### Question 3: Make cluster analysis of the countries where Ukrainian  ######
############# student will preferably go depending on educational level  #######
################ (bac, license, master) over the years 2012-2018. ##############
################################################################################

students_to_country <- education_UKR_2012_2018 %>%
  filter(`Destination region` != "W00:All countries") %>%
  select_if(~ length(unique(.)) > 1)

# check what unique statistical units we can use for data exploration
unique_value_students_to_country<-students_to_country[1]%>%
```

```
  unique()
unique_value_students_to_country


# I want to check dependencies for "OE:Outbound internationally mobile students"
students_to_country_mobile <- students_to_country %>%
  filter(`Statistical unit` == "OE:Outbound internationally mobile students") %>%
  select_if(~ length(unique(.)) > 1)
students_to_country_mobile


scaled.dat <- scale(t(students_to_country_mobile[,-1]))
scaled.dat
# so, it will be level of similarity to go to specific country over the years. But
why? languages, cultural stuf, money, any kind of educational programs and diplomas
euroclust<-hclust(dist(t(scaled.dat)))
euroclust
plot(euroclust, labels=students_to_country_mobile$`Destination region`)


# re-structure data
final_df_students_mobile <- t(students_to_country_mobile)
final_df_students_mobile
final_df_students_mobile <- apply(final_df_students_mobile[-c(1), ],1,function(x)
log(as.numeric(x)))
final_df_students_mobile <- t(final_df_students_mobile)
final_df_students_mobile[(final_df_students_mobile == -Inf)] <- 0
final_df_students_mobile
years <-c(2012:2017)
final_df_students_mobile <-data.frame(years,final_df_students_mobile)
final_df_students_mobile
colnames(final_df_students_mobile) <-c("years","Sub_Saharan_Africa",
"South_and_West_Asia", "Oceania", "Central_and_Eastern_Europe", "Central_Asia",
"East_Asia", "Latin_America", "North_America_and_Western_Europe",
"East_Asia_and_the_Pacific", "Arab_States","Latin_America_and_the_Caribbean")
final_df_students_mobile

p <- plot_ly(final_df_students_mobile,    type = 'scatter', mode = 'lines') %>%
  add_trace(x = ~`years`,y = ~`Sub_Saharan_Africa`, name = 'Sub-Saharan Africa')
%>%
  add_trace(x = ~`years`,y = ~`South_and_West_Asia`, name = 'South and West Asia')
%>%
  add_trace(x = ~`years`,y = ~`Oceania`, name = 'Oceania') %>%
  add_trace(x = ~`years`,y = ~`Central_and_Eastern_Europe`, name = 'Central and
Eastern Europe') %>%
  add_trace(x = ~`years`,y = ~`Central_Asia`, name = 'Central Asia') %>%
  add_trace(x = ~`years`,y = ~`East_Asia`, name = 'East Asia') %>%
  add_trace(x = ~`years`,y = ~`Latin_America`, name = 'Latin America') %>%
  add_trace(x = ~`years`,y = ~`North_America_and_Western_Europe`, name = 'North
America and Western Europe') %>%
  add_trace(x = ~`years`,y = ~`East_Asia_and_the_Pacific`, name = 'East Asia and
the Pacific') %>%
```

```
  add_trace(x = ~`years`,y = ~`Arab_States`, name = 'Arab States') %>%
  add_trace(x = ~`years`,y = ~`Latin_America_and_the_Caribbean`, name = 'Latin
America and the Caribbean') %>%
  layout(
    title = "International mobility",
    yaxis = list(title = "Number of departing people (log scaled)"))
p


# regression
students_to_country_mobile
students_to_country_mobile_norm <-students_to_country_mobile[-c(1)]
students_to_country_mobile_norm
students_to_country_mobile_norm <-
t(apply(students_to_country_mobile_norm,1,function(x) log(as.numeric(x))))
students_to_country_mobile_norm
scatter.smooth(x=students_to_country_mobile_norm[,1],
y=students_to_country_mobile_norm[,2], main="Latin America ~ Western Europe")

# has no meaning

students_to_country_mobile_norm <- as.data.frame(students_to_country_mobile_norm)
students_to_country_mobile_norm

final_df_students_mobile
library("ggpubr")
require(gridExtra)
plot1 <- ggscatter(final_df_students_mobile, x="years", y = "Oceania",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "years", ylab = "Oceania")
plot2 <- ggscatter(final_df_students_mobile, x="years", y =
"North_America_and_Western_Europe",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "years", ylab = "North_America_and_Western_Europe")
plot3 <- ggscatter(final_df_students_mobile, x="years", y = "Arab_States",
                   add = "reg.line", conf.int = TRUE,
                   cor.coef = TRUE, cor.method = "pearson",
                   xlab = "years", ylab = "East_Asia")
grid.arrange(plot1, plot2, plot3)



################################################################################
### Question 4: Discover patterns about preferred fields of studies for a ######
############# male/female student  over the years (2012-2018) and make a   #######
################ prognosis about 2019.##############
################################################################################

educ_patters <- education_UKR_2012_2018 %>%
```

```
  filter(`Field of education` != "_T:Total") %>%
  filter(`Field of education` != "_X:Unspecified") %>%
  filter(`Field of education` != "_Z:Not applicable") %>%
  filter(`Sex` != "_T:Total") %>%
  select_if(~ length(unique(.)) > 1)

View(educ_patters)

educ_patterns_tertiary <- educ_patters %>%
  filter(`Statistical unit` == "FOSEP:Distribution of students in tertiary
education by field of education") %>%
  select_if(~ length(unique(.)) > 1)

View(educ_patterns_tertiary)

# check what unique statistical units we can use for data exploration
unique_unit_educ_patterns_tertiary<-educ_patterns_tertiary[1]%>%
  unique()
unique_unit_educ_patterns_tertiary

unique_field_educ_patterns_tertiary<-educ_patterns_tertiary[3]%>%
  unique()
unique_field_educ_patterns_tertiary

# change char values to numbers
rows_patterns <- nrow(educ_patterns_tertiary)
rows_patterns

#Swap data
educ_patterns_tertiary <-educ_patterns_tertiary[ ,c(3,1,2,4,5,6)]
educ_patterns_tertiary

# Order variables
educ_patterns_tertiary <-
educ_patterns_tertiary[order(educ_patterns_tertiary$`Field of education`),]
educ_patterns_tertiary

#append extra variable
Value <-c(1:40)
educ_patterns_tertiary <-data.frame(Value,educ_patterns_tertiary)
educ_patterns_tertiary

glimpse(educ_patterns_tertiary)

educ_patterns_tertiary <- educ_patterns_tertiary%>%
  mutate(Field.of.education = factor(Field.of.education)) %>%
  mutate(Level.of.education = factor(Level.of.education)) %>%
  mutate(Sex = factor(Sex))
educ_patterns_tertiary
glimpse(educ_patterns_tertiary)
```

```r
gower_dist <- daisy(educ_patterns_tertiary,
                    metric = "gower")
summary(gower_dist)
gower_mat <- as.matrix(gower_dist)
gower_mat

# Output most similar pair
educ_patterns_tertiary[
  which(gower_mat == min(gower_mat[gower_mat != min(gower_mat)]),
        arr.ind = TRUE)[1, ], ]

# Output most dissimilar pair
educ_patterns_tertiary[
  which(gower_mat == max(gower_mat[gower_mat != max(gower_mat)]),
        arr.ind = TRUE)[1, ], ]

# Calculate silhouette width for many k using PAM

sil_width <- c(NA)
for(i in 2:10){
  pam_fit <- pam(gower_dist,
                 diss = TRUE,
                 k = i)
  sil_width[i] <- pam_fit$silinfo$avg.width
}

# Plot sihouette width (higher is better)
plot(1:10, sil_width,
     xlab = "Number of clusters",
     ylab = "Silhouette Width")
lines(1:10, sil_width)

tic("run clustering")
pam_fit <- pam(gower_dist, diss = TRUE, k = 2)

pam_results <- educ_patterns_tertiary%>%
  mutate(cluster = pam_fit$clustering) %>%
  group_by(cluster) %>%
  do(the_summary = summary(.))
toc()
pam_results$the_summary

educ_patterns_tertiary[pam_fit$medoids,]

tsne_obj <- Rtsne(gower_dist, perplexity = 1.5 ,is_distance = TRUE)

tsne_data <- tsne_obj$Y %>%
  data.frame() %>%
  setNames(c("X", "Y")) %>%
  mutate(cluster = factor(pam_fit$clustering),
         Field.of.education = educ_patterns_tertiary$Field.of.education)
```

```r
ggplot(aes(x = X, y = Y), data = tsne_data) +
  geom_point(aes(color = cluster))
```