```r
## Data sources:
## https://www.kaggle.com/angelmm/healthteethsugar
## https://www.kaggle.com/khsamaha/aviation-accident-database-synopses
## https://www.worlddata.info/downloads/

install.packages('tidyr')
install.packages('lubridate')
install.packages("rworldmap")
install.packages("countrycode")
install.packages("caret")
install.packages("QuantPsyc")
install.packages("DMwR")

library("QuantPsyc")
library("rworldmap")
library("countrycode")
library('tidyr')
library('dplyr')
library('lubridate')
library('ggplot2')
library("ggfortify")
library("caret")
library("DMwR")


## Loading the datasets ------------------------------------------------------------------------------------------

## Per capita general government expenditure on health expressed at average
## exchange rate for that year in US dollar. Current prices. Data from WHO.
health_expend <- read.csv("input/healthexpend.csv", stringsAsFactors = TRUE)

## Gross Domestic Product per capita in constant 2000 US$. The inflation but not
## the differences in the cost of living between countries has been taken into
## account. Data from World Bank.
gdp <- read.csv("input/gdp.csv", stringsAsFactors = TRUE)

## The food consumption quantity (grams per person and day) of sugar and sweeters. Data from FAO.
sugar_consumption <- read.csv("input/sugar_consumption.csv", stringsAsFactors = TRUE)

## UN datasets
children_overweight <- read.csv("input/children_under_5_overweight.csv", stringsAsFactors = TRUE)
adults_morbide <- read.csv("input/adults_morbide.csv", stringsAsFactors = TRUE)
poorness <- read.csv("input/population_under_1_dollar.csv", stringsAsFactors = TRUE)
children_per_woman <- read.csv("input/children_per_woman.csv", stringsAsFactors = TRUE)
population_growth <- read.csv("input/population_growth.csv", stringsAsFactors = TRUE)
population_urban <- read.csv("input/population_urban.csv", stringsAsFactors = TRUE)

## Aviation incidents dataset
aviation_incidents <- read.csv("input/AviationData_NTSB_27_12_2016.csv", stringsAsFactors = TRUE)

## General country data
country_data <- read.csv("input/countries.csv", stringsAsFactors = TRUE, sep = ";")

## Manipulating the datasets so they can be merged -----------------------------------------------------------------

## Cleaning general country data
deleted_columns <- c("Country..de.", "Country..local.", "Capital",
                     "Currency.code", "Dialing.prefix", "Url" )
for(name in deleted_columns) {
  print(name)
  country_data[,name] <- NULL
}
country_data[,8:11] <- NULL
country_data <- rename(country_data, Country = Country..en.)

## cleaning health expense
years <- c(1995:2010)
names(health_expend)[2:17] <- years
health_expend <- rename(health_expend, Country =
                          Per.capita.government.expenditure.on.health.at.average.exchange.rate..US..)
health_expend <- gather(health_expend, Year, Health.Expend, 2:17)

## cleaning gdp
years <- c(1960:2011)
names(gdp)[2:53] <- years
gdp <- rename(gdp, Country = Income.per.person..fixed.2000.US..)
gdp <- gather(gdp, Year, GDP, 2:53)

## cleaning sugar_consumption
sugar_consumption$NA..1 <- NULL
years <- c(1961:2004)
names(sugar_consumption)[2:45] <- years
sugar_consumption <- rename(sugar_consumption, Country = NA.)
sugar_consumption <- gather(sugar_consumption, Year, Sugar.Consumption, 2:45)

## Function for cleaning UN datasets
clean_undata <- function(data, subject, gender = TRUE) {
  if(gender == TRUE) {
    col_names <- c("Country", "Year", "Gender", subject)
    names(data)[1:4] <- col_names
    data$Value.Footnotes <- NULL
  } else {
    col_names <- c("Country", "Year", subject)
    names(data)[1:3] <- col_names
    data$Value.Footnotes <- NULL
  }
  data$Country <- as.factor(data$Country)
  data$Year <- as.factor(data$Year)
  return(data)
}

## Cleaning the UN datasets
children_overweight <- clean_undata(children_overweight, "Children.Overweight")
adults_morbide <- clean_undata(adults_morbide, "Adults.Morbide")
poorness <- clean_undata(poorness, "Population.Under.1.Dollar", gender = FALSE)
children_per_woman <- clean_undata(children_per_woman, "Children.Per.Woman",
                                   gender = FALSE)
population_growth <- clean_undata(population_growth, "Population.Growth",
                                  gender = FALSE)
population_urban <- clean_undata(population_urban, "Population.Urban",
                                 gender = FALSE)

## Computing the variables from the aviation dataset
## Extracting and adding year
aviation_incidents$Event.Date <- ymd(aviation_incidents$Event.Date)
aviation_incidents$Year <- year(aviation_incidents$Event.Date)

## Extracting deaths, injured and uninjured victims
aviation_incidents <- group_by(aviation_incidents, Country, Year)
aviation_deaths <- count(aviation_incidents, Country,
                         wt = Total.Fatal.Injuries) %>%
                          rename( Aviation.Victim.Deaths = n)
aviation_injuries <- count(aviation_incidents, Country,
                            wt = c(Total.Serious.Injuries, Total.Minor.Injuries)) %>%
                          rename( Aviation.Victim.Injuries = n)
aviation_uninjured <- count(aviation_incidents, Country,
                             wt = Total.Uninjured) %>%
```

```
                            rename( Aviation.Victim.Uninjured = n)
aviation_victims <- left_join(aviation_deaths, left_join(aviation_injuries,
                              aviation_uninjured, by = c("Country", "Year")),
                              by = c("Country", "Year"))
aviation_victims$Year <- as.character(aviation_victims$Year)
aviation_victims$Country <- gsub(" ", "", aviation_victims$Country)

## Merging the datasets without gender specification ---------------------------

dataframe <- full_join(children_per_woman, gdp, by = c("Country", "Year")) %>%
  full_join(health_expend, by = c("Country", "Year")) %>%
  full_join(poorness, by = c("Country", "Year")) %>%
  full_join(population_growth, by = c("Country", "Year")) %>%
  full_join(population_urban, by = c("Country", "Year")) %>%
  full_join(sugar_consumption, by = c("Country", "Year")) %>%
  full_join(aviation_victims, by = c("Country", "Year"))

## Adding the sexes column with label "both sexes"
dataframe$Gender <- "Both sexes"

## Merging this with the gender specified datasets -----------------------------

dataframe <- full_join(dataframe, children_overweight, by = c("Country", "Year",
                                                              "Gender")) %>%
  full_join(adults_morbide, by = c("Country", "Year", "Gender"))

## Finally merging that with the general country data ---------------------------
dataframe <- left_join(dataframe, country_data, by = "Country")

## Cleaning the new merged datafile --------------------------------------------
table(dataframe$Country)

dataframe$Country <- gsub("United States of America|UnitedStates", "United States", dataframe$Country)
dataframe$Country <- gsub("Venezuela (Bolivarian Republic of)|Venezuela, RB", "Venezuela", dataframe$Country)
dataframe$Country <- gsub("Viet Nam", "Vietnam", dataframe$Country)
dataframe$Country <- gsub("Yemen, Rep.", "Yemen", dataframe$Country)
dataframe$Country <- gsub("United Republic of Tanzania", "Tanzania", dataframe$Country)
dataframe$Country <- gsub("Bahamas, The", "Bahamas", dataframe$Country)
dataframe$Country <- gsub("Republic of Moldova", "Moldova", dataframe$Country)
dataframe$Country <- gsub("Korea, Dem. Rep.", "Korea, Rep.", dataframe$Country)
dataframe$Country <- gsub("Egypt, Arab Rep.", "Egypt", dataframe$Country)
dataframe$Country <- gsub("Iran (Islamic Republic of)", "Iran, Islamic Rep.", dataframe$Country)
dataframe$Country <- gsub("Congo, Dem. Rep.|Congo, Rep.", "Congo", dataframe$Country)
dataframe$Country <- gsub("[[:digit:]]+", NA , dataframe$Country)
dataframe$Country <- gsub("TurksAndCaicosIslands","Turks and Caicos Islands", dataframe$Country)
dataframe$Country <- gsub("AmericanSamoa","American Samoa", dataframe$Country)
dataframe$Country <- gsub("AntiguaAndBarbuda","Antigua and Barbuda", dataframe$Country)
dataframe$Country <- gsub("Bolivia (Plurinational State of)","Bolivia", dataframe$Country)
dataframe$Country <- gsub("BosniaAndHerzegovina","Bosnia and Herzegovina", dataframe$Country)
dataframe$Country <- gsub("AmericanSamoa","American Samoa", dataframe$Country)
dataframe$Country <- gsub("UnitedArabEmirates","United Arab Emirates", dataframe$Country)
dataframe$Country <- gsub("UnitedKingdom","United Kingdom", dataframe$Country)
dataframe$Country <- gsub("TrinidadAndTobago","Trinidad and Tobago", dataframe$Country)
dataframe$Country <- gsub("SaudiArabia","Saudi Arabia", dataframe$Country)

##checking how many rows do not have a country
which(is.na(dataframe$Country))

##deleting the rows without a country
dataframe<- filter(dataframe, Country != "", Country != "Unknown" )

dataframe$Year <- as.numeric(dataframe$Year)

##removing NAs in Year and Country columns including a function removing NAs in columns.
clean_NA <- function(dataset, column){
  dataset <- filter(dataset, column != "NA")
}

clean_NA(dataframe, dataframe$Country)
clean_NA(dataframe, dataframe$Year)

##cleaned the Adults.Morbide column of the additional numbers in brackets.
dataframe$Adults.Morbide <- gsub("\\[.*?\\]", '', dataframe$Adults.Morbide)

#adding the un-abbreviations of all countries
dataframe$CountryUN <- countrycode(sourcevar = dataframe$Country,
                                   origin = "country.name", destination = "iso3c", warn = FALSE)

##creating a categorical variable with Poverty Rate according to population under 1 dollar.
dataframe$Povertyrate[dataframe$Population.Under.1.Dollar < 5] <- "Low"
dataframe$Povertyrate[dataframe$Population.Under.1.Dollar > 5 & dataframe$Population.Under.1.Dollar < 70] <- "Average"
dataframe$Povertyrate[dataframe$Population.Under.1.Dollar > 70] <- "High"

#creating another categorical variable with Fertility rate
dataframe$FertilityRate[dataframe$Children.Per.Woman < 2] <- "Low"
dataframe$FertilityRate[dataframe$Children.Per.Woman > 2 & dataframe$Children.Per.Woman < 3] <- "Average"
dataframe$FertilityRate[dataframe$Children.Per.Woman > 3 & dataframe$Children.Per.Woman < 5] <- "High"
dataframe$FertilityRate[dataframe$Children.Per.Woman > 5] <- "Very High"

#creating another categorical variable with levels of urbanization
dataframe$CountryUrbanization[dataframe$Population.Urban < 25] <- "Not very urbanized"
dataframe$CountryUrbanization[dataframe$Population.Urban > 25 & dataframe$Population.Urban < 50] <- "Mediumly urbanized"
dataframe$CountryUrbanization[dataframe$Population.Urban > 50 & dataframe$Population.Urban < 75] <- "Highly urbanized"
dataframe$CountryUrbanization[dataframe$Population.Urban > 75 & dataframe$Population.Urban < 100] <- "Extremely urbanized"

#creating a numerical variable of total population in cities
dataframe$Population_number_in_cities <- dataframe$Population*(dataframe$Population.Urban/100)

#starting with providing descriptive results and plots------------------

#creating a function that calculates the minimum, maximum and mean for a particular variable, as
#in this case the main variables of interest are GDP and population growth the below list shows the summaries
mean_min_max <- function(data, variable) {
  summary <- summarize(data, Mean = mean(variable), Minimum = min(variable), Maximum = max(variable))
  summary
}

GDP_populationgrowth_data <- filter(dataframe, !is.na(GDP) & !is.na(Population.Growth)) %>%
  select(Country, Year, GDP, Population.Growth)

list(mean_min_max(GDP_populationgrowth_data, GDP_populationgrowth_data$GDP),
     mean_min_max(GDP_populationgrowth_data, GDP_populationgrowth_data$Population.Growth))

#visualizing population growth on the world map. We can zoom in on particular areas from here.

df_popgrowth <- data.frame(Country = dataframe$CountryUN,
                    Population_growth = dataframe$Population.Growth)

popgrowth_Map <- joinCountryData2Map(df_popgrowth, joinCode = "ISO3",
                              nameJoinColumn = "Country")

mapCountryData(popgrowth_Map, nameColumnToPlot="Population_growth", mapTitle = "Fastest growing countries", catMethod = "categorical",
               missingCountryCol = gray(.8), addLegend = FALSE)

#it seems that in Europe, there is relatively limited population growth compared to for example Asia and Africa.
europe_africa_asia <- filter(dataframe, Continent %in% c('Europe', 'Africa', 'Asia')) %>%
                    select(Population.Growth, Children.Per.Woman, Year, Country, Continent,
                           Government.form, GDP, Population.Under.1.Dollar, Population.Urban)
```

```r
descriptive_europe_africa_asia <- summarise(group_by(europe_africa_asia, Continent), `Mean population growth` = mean(Population.Growth, na.rm = TRUE), `Mean children per woman`
= mean(Children.Per.Woman, na.rm = TRUE))

#this seems about right, on average people in Africa give birth to way more children (of course heavily associated with population growth)
#Although it might be redundant, a simple t-test can be conducted to see whether this difference is significant
europe_growth <- filter(europe_africa_asia, Continent == 'Europe') %>%
  select(Population.Growth)
africa_growth <- filter(europe_africa_asia, Continent == 'Africa') %>%
  select(Population.Growth)
asia_growth <- filter(europe_africa_asia, Continent == 'Asia') %>%
  select(Population.Growth)

difference_growth_africa <- t.test(europe_growth, africa_growth, alternative = 'less', var.equal =  FALSE)
difference_growth_asia <- t.test(europe_growth, asia_growth, alternative = 'less', var.equal =  FALSE)


#visualizing that just comparing continents is a risky business, there are many internal clusters within these continents

data_africa_asia <- filter(europe_africa_asia, Continent %in% c("Africa", "Asia") , Year == 2008) %>%
  select(Country, Continent, Children.Per.Woman, Population.Growth) %>%
  na.omit()

africa_data <- filter(data_africa_asia, Continent == "Africa")
asia_data <- filter(data_africa_asia, Continent == "Asia")

africa_cluster <- hclust(dist(africa_data[-1:-2]))
asia_cluster <- hclust(dist(asia_data[-1:-2]))

africa_dendogram <- plot(africa_cluster, label = africa_data$Country, xlab = 'Clustering in Africa',
     main = 'Cluster Dendogram African countries', hang = -1)
asia_dendogram <- plot(asia_cluster, label = asia_data$Country, xlab = 'Clustering in Asia',
     main = 'Cluster Dendogram Asian countries', hang = -1)

# the two contintent differ significantly. Still, it might be interesting to see whether this was always the case;
# in other words, how were the developments across the years.

difference_accros_years <- europe_africa_asia %>%
  group_by(Year, Continent) %>%
  summarise(population_growth_mean = mean(Population.Growth, na.rm=TRUE)) %>%

  ggplot(aes(x = Year, y = population_growth_mean, color = Continent)) +
  geom_line() +
  ggtitle("Average population growth per year") +
  scale_y_continuous(name = "Average population growth", breaks = c(0, 0.5, 1, 1.5, 2, 2.5)) +
  scale_x_continuous(breaks = c(seq(1960, 2015, by=5)))

difference_accros_years

#okay, so it seems that especially Europe experienced a declining population growth throughout time

#let's find out what role other variables play, for example, government form in 2010

government_form_bar <- filter(europe_africa_asia, Year == 2008) %>%
  ggplot(aes(x = Government.form, fill = Continent)) +
  geom_bar() +
  ggtitle('Government type in 2008 in Europe, Africa and Asia') +
  scale_y_continuous(name = "Total number of countries") +
  scale_x_discrete(name = "Government type") +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))

government_form_bar

# it seems that European countries mostly have parlementary republics, whereas african countries often have presidential republics.
# an explortive analysis shows that also the the organizations financial position differs per continent
descriptive_europe_africa_asia_finance <- select(europe_africa_asia, Year, Continent, Population.Under.1.Dollar, GDP) %>%
  filter(Year == 2008) %>%
  group_by(Continent) %>%
  summarise(`Mean under $1` = mean(Population.Under.1.Dollar, na.rm = TRUE),
            `Mean GDP` = mean(GDP, na.rm = TRUE))

#we see that the GDP developed rather irregulary
ggplot(europe_africa_asia, aes(x = Year, y = GDP, fill = Continent)) +
  geom_bar(stat = 'identity') +
  facet_wrap(~ Continent)

#so, talking about growth, we might espect a relationship between for example GDP and population growth
dollarchildren_scatter <- ggplot(data = europe_africa_asia, aes(x = Population.Under.1.Dollar, y = Children.Per.Woman, color = Continent)) +
  geom_point() +
  geom_smooth() +
  ggtitle('Scatterplot: Population under $1 and Children per woman') +
  scale_y_continuous(name = "Children per woman") +
  scale_x_continuous(name = "Population under $1")

dollarchildren_scatter

#it is also interesting to see wheter urbanization differs per continent; all in preparation for the regressions that will be ran later
Urbanization_boxplot <- filter(europe_africa_asia, Year == 2011) %>%
  ggplot(aes(x = Continent, y = Population.Urban, color = 'blue')) +
  geom_boxplot(position = "dodge", notch = TRUE, notchwidth = 0.5) +
  ggtitle('Boxplot: Urbanization levels per continent') +
  scale_y_continuous(name = "Level of urbanization") +
  theme(legend.position = 'none')
Urbanization_boxplot

Urbanization_boxplot

### Statistical Analysis ------------------------------------------

# creating a training and a test set for 'dataframe' dataframe.
set.seed(1)
trn_indexes <- sample.int(nrow(dataframe), size = 0.8 * nrow(dataframe))

trn_dataframe <- dataframe[trn_indexes, ]
tst_dataframe <- dataframe[-trn_indexes, ]

# We already saw that population growth is different throughout the continents.
# Lets compute the corresponding regression coefficients for population growth per continent.

lm_dataframe_Con <- lm(Population.Growth ~ Continent, data = trn_dataframe)

lm_dataframe_Con_autoplot <- autoplot(lm_dataframe_Con, which = 1:2)

# Continents with more developed countries negatively influence the world population. (Africa is the baseline, so the inter-difference between the coefficient is of interest)

# Predicting the Population Growth as a function of fertility Rate.
lm_dataframe_Fer <- lm(Population.Growth ~ FertilityRate, data = trn_dataframe)

lm_dataframe_Fer_autoplot <- autoplot(lm_dataframe_Fer, which = 1:2)

# Obviously, a low fertility rate means less children per woman impacts the population growth negatively (average fertility as baseline).
# The results from the auto plots suggest that fertility rate is a good predictor of children per woman but the long tails in the QQ-plot suggest that the data is not normally
distributed.
# Let's take a look at Population Growth as a function of Poverty Rate.

lm_dataframe_Pov <- lm(Population.Growth ~ Povertyrate, data = trn_dataframe)

lm_dataframe_Pov_autoplot <- autoplot(lm_dataframe_Pov, which = 1:2)
```

```r
# As seen with the plots, countries with high poverty rates have stronger population growths (baseline: average poverty).
# The diagnostic plots show decent lines with small acceptable deviations, the model is accepted.
# Now let's take a closer look at the effect of urbanization on population growth.

trn_dataframe$CountryUrbanization <- factor(trn_dataframe$CountryUrbanization,
                                            levels = c("Mediumly urbanized", "Not very urbanized", "Highly urbanized", "Extremely urbanized"))

lm_dataframe_Urb <- lm(Population.Growth ~ CountryUrbanization, data = trn_dataframe)

lm_dataframe_Urb_autoplot <- autoplot(lm_dataframe_Urb, which = 1:2)

# As seen in this model, countries that are not very urbanized contribute the most to population growth.
# Highly urbanized countries have a substantial bigger negative effect on population growth than extremely urbanized countries.
# The diagnostic plots reveal that there are outliers in the errors and no normal distribution of the data, The model isn't great
# More developed countries, with people living in cities, have a negative effect on population growth.

# We now try multiple descriptive variables and their interactions to explain population growth
lm_dataframe_Multi <- lm(Population.Growth ~ GDP * Health.Expend * Population.Under.1.Dollar * Sugar.Consumption * Children.Per.Woman, data = trn_dataframe)
summary(lm_dataframe_Multi)

# The R-squared is relatively high. The model suggests that a 80% of variance is explained by the variables.

autoplot(lm_dataframe_Multi, which = 1:2)

# The diagnostic plot show an uneven distribution of the errors and small deviations on the normal distribution.
# Let's try log the dependent variable and check if this improves the distribution of the errors.
lm_dataframe_Multi_log <- lm(log(Population.Growth) ~ GDP * Health.Expend * Population.Under.1.Dollar * Sugar.Consumption * Children.Per.Woman, data = trn_dataframe)
summary(lm_dataframe_Multi_log)

# The R-squared is relatively high. The model suggests that a 80% of variance is explained by the variables.

lm_dataframe_Multi_log_autoplot<- autoplot(lm_dataframe_Multi_log, which = 1:2)

# Using log does not improve the diagnostic plots.
# Now we try to take the square root of Population Growth.

lm_dataframe_Multi_sqrt <- lm(sqrt(Population.Growth) ~ GDP * Health.Expend * Population.Under.1.Dollar * Sugar.Consumption * Children.Per.Woman, data = trn_dataframe)
summary(lm_dataframe_sqrt)

# The R-squared is relatively high. The model suggests that a 80% of variance is explained by the variables.

lm_dataframe_sqrt_autoplot <- autoplot(lm_dataframe_Multi_sqrt, which = 1:2)



# This also is not an improvement for the model we build. We keep the original.
# Our ext step: Check for overfitting with cross-validation (caret package)
set.seed(2)
train_PopGrowth <- train(form = Population.Growth ~ GDP * Health.Expend * Population.Under.1.Dollar * Sugar.Consumption * Children.Per.Woman, data = trn_dataframe,
                         method = "lm", trControl = trainControl(method = "cv"), na.action = na.omit)

train_PopGrowth

# We can see that the R-Squared has dropped but not significantly. The R-Squared is still high.
# Now we use the set-aside test-set with our unseen data to check the R-Squared.
set.seed(3)
train_PopGrowth_test <- train(form = Population.Growth ~ GDP * Health.Expend * Population.Under.1.Dollar * Sugar.Consumption * Children.Per.Woman, data = tst_dataframe,
                              method = "lm", trControl = trainControl(method = "cv"), na.action = na.omit)

train_PopGrowth_test

# The R-Squared on the test-set dropped 10 points, but is still decent considering the sample size.
# So we can conclude that this model is explaining the variance in population growth. And still performs decent on unseen data.

## Overig (moet nog plaatsje krijgen in rapport) -----------------------------------------------------------------

# Standardized Coefficients
mod <- lm(Population.Growth ~ Continent, data = trn_dataframe)
coef_lmbeta <- lm.beta(mod)

coef_lmbeta

# Detecting outliers
dataframe_detection <- dataframe[,3:4]
outlier_detection <- lofactor(dataframe_detection, k=5)
plot(density(outlier_detection))

n <- nrow(outlier_detection)
labels <- 1:n
labels[-outlier_detection] <- "."
biplot(prcomp(outlier_detection), cex=.8, xlabs=labels)

outliers <- order(outlier_detectionm, decreasing= T)
print(outliers)

lm_dataframe_GDP <- lm(Population.Growth ~ GDP, data = trn_dataframe)

lm_dataframe_GDP_autoplot <- autoplot(lm_dataframe_GDP, which = 1:2)

ggplot(data = trn_dataframe, aes(y = Population.Growth, x = log(GDP))) +
  geom_point() +
  geom_smooth(method = "lm")
```