Data Preprocessing of Population and Life expectancy based on Gender

Code

Deepthi Suresh

13 Oct 2023

Setup

Hide

```
# Load the necessary packages required to reproduce the report. For example:

library(readr)
library(dplyr)
library(tidyr)
library(magrittr)
library(openxlsx)
library(readxl)
library(Hmisc)
library(editrules)
library (MVN)
library(forecast)
library(kableExtra)
library(outliers)
```

Executive Summary

The report is based on four datasets - Population and Life Expectancy (LE) for both males and females. As a part of the pre-processing the first step was to read in the data into R and then make adjustments to the data especially removing the ‘world’ observation to avoid skewness. The data was then tidied by converting the year into a variable with its own column. Then the male and female datasets of population and LE was combined using rbind. The structure of the dataset was then looked at and necessary conversion of data type was done. Then, the two data was ready to be merged using join function to form the merged dataset. Once the dataset was ready, a new derived variable called relative Life expectancy was created, which was based on the average global life expectancy. The final data was then scanned for missing values and imputation or deletion was performed based on the variable to ensure there was no missing values. Then the dataset was checked for any outliers and the numeric variable had some outliers which were dealt by capping or Winsorising method by using a special function. Once the missing values are outliers are handled, the data was now ready for transformation. Several data transformation techniques were performed to identify the best one to convert the data into a normal distribution. Scaling and Centering was also performed to enhance the comparability

Data

The first dataset considered here is of the population of 265 countries in the world and the total world population divided into two datasets based on the gender under the names pop_female (female population across countries) and pop_male (male population across countries). The second dataset has the details of the life expectancy at birth of males and females across different countries, divided into two datsets called le_female(Life expectancy at birth, female (years)) and le_male (Life expectancy at birth, male (years)).

The variables in all 4 datasets are given below:

Country Name - Name of the countries that are assessed

Country Code - abbreviation given to each country

Indicator Name - The variable for which the data is collected.

Population or life expectancy of the female or males

Indicator Code - code for identifying the dataset. most likely used internally

Years from 1960 - 2022- the data is assessed for the years 1960-2022.

The data is sourced from the world bank website with the below URL:

Population, female -https://data.worldbank.org/indicator/SP.POP.TOTL.FE.IN?locations=AU (https://data.worldbank.org/indicator/SP.POP.TOTL.FE.IN?locations=AU)

Population, male - https://data.worldbank.org/indicator/SP.POP.TOTL.MA.IN?locations=AU (https://data.worldbank.org/indicator/SP.POP.TOTL.MA.IN?locations=AU)

Life expectancy, female -https://data.worldbank.org/indicator/SP.DYN.LE00.FE.IN?locations=AU (https://data.worldbank.org/indicator/SP.DYN.LE00.FE.IN?locations=AU)

Life expectancy, male-https://data.worldbank.org/indicator/SP.DYN.LE00.MA.IN?locations=AU (https://data.worldbank.org/indicator/SP.DYN.LE00.MA.IN?locations=AU)

Since the dataset contains the total world as one of the observation, I have removed it using the filter function as the main focus of the study is to understand the country-wise variable distribution (it avoids the skewness of data).

Head function is used to make sure the data is correctly loaded.

Hide

```
# reading in population data for males and females
pop_female <- read_excel("Population_female.xls",skip = 3)
pop_male <- read_excel("Population male.xls",skip = 3)
# reading in Life expectancy data for males and females
le_male <- read_excel("Life expectancy male.xls",skip = 3)
le_female <- read_excel("Life expectancy female.xls",skip = 3)

#removing the world observation to avoid skewness of data
pop_female <- pop_female %>% filter(`Country Name` != "World")

pop_male <- pop_male %>% filter(`Country Name` != "World")

le_female <- le_female %>% filter(`Country Name` != "World")

le_male <- le_male %>% filter(`Country Name` != "World")

# checking if the data is loaded correctly
head(pop_female, n=3)
head(pop_male, n=3)
head(le_female,n=3)
head(le_male,n=3)
```

Tidy & Manipulate Data I

Hide

```
#tidying population data based on year
PFemale_tidy <- pop_female %>% pivot_longer(names_to = "Year", values_to = "Populat
ion", cols = 5:67)
head(PFemale_tidy,n=3)
PMale_tidy <- pop_male %>% pivot_longer(names_to = "Year", values_to = "Populatio
n", cols = 5:67)
head(PMale_tidy,n=3)
#combining male and female population datset into 1
Population <- rbind(PMale_tidy,PFemale_tidy)
head(Population,n=3)
#tidying life expectancy data based on year
le_female_tidy <- le_female %>% pivot_longer(names_to = "Year", values_to = "Life e
xpectancy at birth", cols = 5:67)
head(le_female_tidy,n=3)
le_male_tidy <- le_male %>% pivot_longer(names_to = "Year", values_to = "Life expec
tancy at birth", cols = 5:67)
head(le_male_tidy,n=3)
#combining male and female population and LE dataset into 1
Population <- rbind(PMale_tidy,PFemale_tidy)
LE <- rbind(le_female_tidy,le_male_tidy)
head(LE,n=3)
```

All the datasets in this study has all the years from 1960-2022 in separate columns which indicated that the data is untidy, as it doesn't meet the principle of 'every variable must have its own column'. In order to tidy the dataset, pivot_longer() function has been used on both population and LE of females and males separately. The head() function shows that the dataset is now tidy and ready to be combined.

The Population and LE data is created by combining the female and male population and life expectancy data respectively using the rbind() function. the head() function shows that the process of combining is successful. LE and Population now has 33390 observations with 6 variables each

Understand

Hide

```
#checking and updating the structure of Population dataset
str(Population)
#converting country code into factor variable and checking the levels
Population$`Country Code` <- as.factor(Population$`Country Code`)
#converting Indicator name into factor variable and checking the levels and convert
ing the labels to Male and "Female"
unique(Population$`Indicator Name`)
Population$`Indicator Name`<- factor(Population$`Indicator Name`,levels= c("Populat
ion, male","Population, female"),
              labels=c("Male","Female"),ordered = TRUE)
levels(Population$`Indicator Name`)
class(Population$`Indicator Name`)
#converting country Name into factor variable and checking the levels
Population$`Country Name` <- as.factor(Population$`Country Name`)
class(Population$`Country Name`)
#checking the updated structure of the dataset
str(Population)
#checking and updating the structure of Life Expectancy dataset
str(LE)
#converting country code into factor variable and checking the levels
LE$`Country Code` <- as.factor(LE$`Country Code`)
#converting Indicator name into factor variable and checking the levels and convert
ing the labels to Male and "Female"
unique(LE$`Indicator Name`)
LE$`Indicator Name`<- factor(LE$`Indicator Name`,levels= c("Life expectancy at birt
h, male (years)","Life expectancy at birth, female (years)"),
                      labels=c("Male","Female"),ordered = TRUE)
levels(LE$`Indicator Name`)
class(LE$`Indicator Name`)
#converting country Name into factor variable and checking the levels
LE$`Country Name` <- as.factor(LE$`Country Name`)
class(LE$`Country Name`)
#checking the updated structure of the dataset
str(LE)
#Merging the population and LE dataset together to form Merged dataset
Merged <- inner_join(Population,LE, by = c ("Country Code", "Indicator Name", "Yea
r", "Country Name"))
# Removing the indicator code variable as it is irrelevant to the study and create
a cleaner datset
Merged <- Merged %>% select(-`Indicator Code.x`,-`Indicator Code.y`)
```

The structure of the Population data (using str())shows that all variables except Population is Character, Population variable is in numeric. here the as.factor function is used to convert the Country code, Indicator Name and Country Name into factor variables by checking the unique values using the unique() function. The level() functions shows the different levels for the updated factor variables. The indicator name variable is further labelled as Male and Female for “Population,male and”Population,female” and ordered. the class() indicates an ordered variable which shows that the conversion is successful.

The same steps are followed for the LE data as they have the same variables.

The str() function shows the updated structure of both population and LE and shows that the variables are successfully converted into factor where necessary.

Once the structure of the data is corrected, the population and LE data is merged to form the ‘Merged’ data using inner join function as inner join help in identifying the common countries and variables between the two data. In this case since the observations were same for both, there were no deletion of data. Using the left or right join was also giving the same result because of the common observations and variables.

To further tidy the data the indicator code column has been removed as they are irrelevant to the study and most likely used as an internal reference.

Tidy & Manipulate Data II

Hide

```
#Creating a relative life expectancy variable

# calculating the mean global life expectancy
global_average_life_expectancy <- round(mean(Merged$`Life expectancy at birth`,na.r
m= TRUE),2)

Merged$Relative_Life_Expectancy <- round(with(Merged,
                                              `Life expectancy at birth` / global_averag
e_life_expectancy),2)
```

A new variable Relative Life expectancy is created here using the global average life expectancy. the global average LE is calculated by the mean of the LE across all countries at birth (na.rm - removes all NA in the data). The relative frequency is the calculated by dividing the life expectancy at birth for each country by the global average life expectancy. This derived variable provides a measure of how each country’s life expectancy compares to the average. A value greater than 1 indicates a above average life expectancy for the country, a value less than 1 indicates below average LE and equal to 1 indicates the LE of the country is equal to the global average.

Now the merged dataset has 7 variables instead of the 6 including the new Relative life expectancy variable

Scan I

Hide

```
#scanning for missing values

colSums(is.na(Merged))

#There is missing values in Population , life expectancy at birth and relative life
expectancy

# to see the na values in population
missing_population <- Merged %>% filter (., is.na(`Population`)== TRUE)

#Since there are only 186 observation and is for the country 'not classified' and '
West bank and gaza', we can exclude these values. It also only accounts for 0.5% of
the total data

Merged <- Merged[complete.cases(Merged$Population), ]

colSums(is.na(Merged))

#missing values in life expectancy column
missing_LE <- Merged %>% filter (., is.na(Merged$`Life expectancy at birth`)== TRU
E)

#Imputing values for LE column by using the mean
Merged$`Life expectancy at birth` <-impute(Merged$`Life expectancy at birth`, fun =
mean)

colSums(is.na(Merged))
#Recalculating the Relative LE
global_average_life_expectancy <- round(mean(Merged$`Life expectancy at birth`,na.r
m= TRUE),2)

Merged$Relative_Life_Expectancy <- round(with(Merged,
                                              `Life expectancy at birth` / global_averag
e_life_expectancy),2)

colSums(is.na(Merged))

# Checking for infinite or NaN or NA values using a function called is.specialorNA

is.specialorNA <- function(x){
  if (is.numeric(x)) (is.infinite(x) | is.nan(x) | is.na(x))
}
sapply(Merged, function(x) sum(is.specialorNA(x) ))

#no infinite or NAN values

#checking for obvious inconsistencies
# Relative life expectancy has to be between 0 and 2
(Rule1 <- editset(c("`Relative_Life_Expectancy` >= 0", "`Relative_Life_Expectancy`
<= 2")))

#Population of a country is expected to be more than 0
```

```
Violation1 <-violatedEdits(Rule1, Merged)
summary(Violation1)

(Rule2 <- editset(c("`Population` >= 0")))
Violation2 <-violatedEdits(Rule2, Merged)
summary(Violation2)
```

To identify if the data has any missing values colsums(is.na()) is used. the results show that there are 186 missing values in population and 1782 values in LE and Relative LE. Upon close inspection of the missing values in population by using a filter function, it can be seen that the 186 observations are in the country “not classified” and “West bank and Gaza” and accounts for only 0.5% of the total data. there we can exclude these values by using the complete.cases function.

The colsums(is.na()) function now shows that there are no more missing values in Population. however there are still 1596 missing values in Life expectancy at birth and Relative LE.To have a better understanding of the missing value a variable called missing_LE is created which shows the NAs are spread across countries. Since its not ideal to waste all the information because of missing values, the best approach to dealing with it will be to impute it with the mean value.

The colsums(is.na()) function now shows that there are no more missing values in Life expectancy, however there are still 1596 missing values in Relative LE. Since Relative LE is a derived variable, we can recalculate the values using the formula used previously. The colsums(is.na()) function now shows that there are no more missing values

In order to check for infinite or NaN values a function called is.specialorNA is created and then applied to the data using the sapply function. The function shows that there are no NaN or special values in the data.

In order to check for obvious inconsistencies, we have set up two rules using the editset() function. One pertaining to the population stating that the population of a country has to greater than 0 and the second related to the relative life expectancy - stating that the Relative LE should be between 0 and 2. The summary() function shows that there are no violation detected for both the rules.

Scan II

Hide

```
Merged$Population %>%  boxplot(main="Boxplot of Population", ylab="Population", col
= "red")
Merged$`Life expectancy at birth` %>%  boxplot(main="Boxplot of Life expectancy", y
lab="Population", col = "green")
Merged$Relative_Life_Expectancy%>%  boxplot(main="Boxplot of Relative Life expectan
cy", ylab="Population", col = "blue")

#checking for outliers using the Z-scores
z.scores <- Merged$Population %>%  scores(type = "z")
z.scores %>% summary()
length (which(abs(z.scores) >3 ))

z.scores <- Merged$`Life expectancy at birth` %>%  scores(type = "z")
z.scores %>% summary()
length (which(abs(z.scores) >3 ))

z.scores <- Merged$Relative_Life_Expectancy %>%  scores(type = "z")
z.scores %>% summary()
length (which(abs(z.scores) >3 ))

#scatter plot beterrn Life Expectancy and Population
Merged %>% plot( Population ~`Life expectancy at birth`,data = ., ylab="Populatio
n", xlab="LE", main="Population by Life expectancy")

# Imputing the outliers by capping
cap <- function(x) {
  quantiles <- quantile(x, c(0.05, 0.25, 0.75, 0.95))
  x[x < quantiles[2] - 1.5 * IQR(x)] <- quantiles[1]
  x[x > quantiles[3] + 1.5 * IQR(x)] <- quantiles[4]
  (x)
}

Subset <- Merged %>% dplyr::select(Population,`Life expectancy at birth`,Relative_L
ife_Expectancy)
head(Subset)

Merged_capped <- sapply(Subset, FUN = cap)
summary(Merged_capped)
```

One of the ways of detecting outlier is using the boxplot, in which “Tukey’s method of outlier detection” is used to detect outliers. The box plot of population, life expectancy and Relative life expectancy shows that they have outliers. In order to identify the number of outliers z score method is used. According to this method if the absolute value of its z-score is greater than 3, then it is regarded as an outlier. From the summary of Population, we can see that the z-score has a minimum of -0.3121 and maximum of 10.5034.The length() function shows that the population variable has 776 outliers. Similarly, the Life expectancy variable has 86 outliers and the Relative Life expectancy variable has 81 outliers.

The scatter plot also shows that there are some possible outliers for the pair of population and Life expectancy variables.

Since the number of outliers doesn’t seem like they were due to any data processing errors, deleting or imputing them with mean is not the right strategy to handle it. In order to handle the outliers the capping or Winsorising method is being used. A function called cap is created to cap the values outside the limits.

Since this function can be performed only on numerical values, a subset is created for population, LE and relative LE variables. Merged_capped data is created by applying the cap function on the Subset data.The summary of the capped data shows the updated summary statistics.

Transform

Hide

```
# data transformation - to decrease the skewness and convert the distribution into
a normal distribution
hist(Merged$Population)

#the log transformation
log10_population <- log10(Merged$Population)
hist(log10_population) # approximately normal

log_population <- log(Merged$Population)
hist(log_population) # log10 works better

sqrt_population <- sqrt(Merged$Population)
hist(sqrt_population) #still skewed

sqr_population <- Merged$Population^2
hist(sqr_population)  #still skewed

recip_population <- 1/Merged$Population
hist(recip_population)  #still skewed

BoxCox_population<- BoxCox(Merged$Population,lambda = "auto")
hist(BoxCox_population)  #still skewed

#centring and scaling or z score standardisation
z_Population <- scale(Merged$Population, center = TRUE, scale = TRUE)
hist(z_Population)

z_LE <- scale(Merged$`Life expectancy at birth`, center = TRUE, scale = TRUE)
hist(z_LE)

z_relative_LE<- scale(Merged$Relative_Life_Expectancy, center = TRUE, scale = TRUE)
hist(z_relative_LE)
```

The Transformation are done only the population variable since there is a page limit on this project.The initial histogram of population shows that its left skewed Different types of transformations are being done to convert it to normal distribution such as log, lo10, sqrt,sqr, BoxCox etc. It can be seen that the log10 and log transformations works best. Log 10 is slightly better than the log transformations in this case.

The Z-score standardization is also done on all three variables to preprocess the numerical variables. With centering, the distributions are centered around a mean of 0 and with scaling has brought the scales of variables between -4 and 2 for easy comparability

Link to Presentation

The link to the recording is given below: https://rmit-arc.instructuremedia.com/embed/d481f6ea-e77e-4556-859a-0c0e4c8b2a4d (https://rmit-arc.instructuremedia.com/embed/d481f6ea-e77e-4556-859a-0c0e4c8b2a4d)

References

Sona Taheri (2023) Data Wrangling [Module 3 - 9 Demo and Notes], RMIT University,Melbourne Studio. (n.d.). Retrieved from Canvas, accessed on 19 September 2023:https://rmit.instructure.com/accounts/1/external_tools/38?launch_type=global_navigation(https://rmit.instructure.com/accounts/1/external_tools/38?launch_type=global_navigation (https://rmit.instructure.com/accounts/1/external_tools/38?launch_type=global_navigation(https://rmit.instructure.com/accounts/1/external_tools/38?launch_type=global_navigation))

Bache S, Wickham H (2022). magrittr: A Forward-Pipe Operator for R R package version 2.0.3,https://CRAN.R-project.org/package=magrittr(https://CRAN.R-project.org/package=magrittr (https://CRAN.R-project.org/package=magrittr(https://CRAN.R-project.org/package=magrittr))

Damian W. Betebenner (2021). randomNames: Function for Generating Random Names and a Dataset. (Rpackage version 1.5-0.0 URLhttps://cran.r-project.org/package=randomNames(https://cran.rproject.org/package=randomNames (https://cran.rproject.org/package=randomNames))

Harrell Jr F (2023).Hmisc: Harrell MiscellaneousR package version 5.1-1,https://R (https://R) Core Team (2023)R: A Language and Environment for Statistical Computing R Foundation for StatisticalComputing, Vienna, Austria. https://www.R-project.org/ (https://www.R-project.org/) (https://www.R-project.org/ (https://www.R-project.org/))

Wickham H, François R, Henry L, Müller K, Vaughan D (2023). dplyr: A Grammar of Data Manipulation. Rpackage version 1.1.2,https://CRAN.R-project.org/package=dplyr(https://CRAN.Rproject.org/package=dplyr (https://CRAN.R-project.org/package=dplyr(https://CRAN.Rproject.org/package=dplyr))

bank, W. (n.d.). Life expectancy at birth,male. Retrieved from data.worldbank: https://data.worldbank.org/indicator/SP.DYN.LE00.MA.IN?locations=AU (https://data.worldbank.org/indicator/SP.DYN.LE00.MA.IN?locations=AU)

bank, W. (n.d.). Population,male. Retrieved from data.worldbank: https://data.worldbank.org/indicator/SP.POP.TOTL.MA.IN?locations=AU (https://data.worldbank.org/indicator/SP.POP.TOTL.MA.IN?locations=AU)

bank, W. (n.d.). Life expectancy at birth,female. Retrieved from data.worldbank:https://data.worldbank.org/indicator/SP.DYN.LE00.FE.IN?locations=AU (https://data.worldbank.org/indicator/SP.DYN.LE00.FE.IN?locations=AU)

bank, W. (n.d.). Population,female. Retrieved from data.worldbank: https://data.worldbank.org/indicator/SP.POP.TOTL.FE.IN?locations=AU (https://data.worldbank.org/indicator/SP.POP.TOTL.FE.IN?locations=AU)