MATH2349 Data Wrangling

Assignment 2

Rajib Debnath (3856291)

Required packages

```
# Loading required required packages

library(readr)
library(Hmisc)
library(tidyr)
library(dplyr)
library(writexl)
library(editrules)
library(outliers)
library(forecast)
```

Executive Summary

- The objective of this report is to demonstrate the data preprocessing knowledge and skill that I have gained throughout this course.
- I wanted to investigate if there was any relationship between a country's GDP per Capita and its health spending? To achieve that, I wanted to perform a regression analysis of the variable of interest to investigate whether the outcome is statistically significant or not.
- To have broad coverage, instead of G7 or G20, I have focused my investigation on OECD (Organisation for Economic Co-operation and Development) countries which is an association of 37 nations. Further, considered a wide timeframe of 20 years to make the data robust enough to perform complex analysis.
- Three different datasets from two different sources were imported.
- Then, inspected the datasets and explored the structures and attributes. Checked each dataset with a proper understanding of each variable.
- After that, filtered out the necessary data required for my analysis and converted it to an appropriate data type.
- One of the dataset was untidy. So, reshaped the data to ensure that the dataset follows tidy data rules.
- After that I merged all three datasets together to prepare a holistic database that can meet my entire analysis requirement.
- Next, created one new variable that provided additional insights and enabled further analysis.
- After that data were scanned for or missing values, special values, and obvious errors and suitable steps were taken to handle those values.
- Numeric data variables were checked for any potential outlier and provided detailed analysis and rationale to keep those identified outliers intact.
- Finally applied different data transformation methods and shortlisted the approach that showed relatively better performance in terms of reducing the skewness in the numeric data.

Data

A detailed description of datasets considered for data preprocessing, their sources, and variable descriptions are as follows:

Dataset Source

I have collected GDP per Capita (dataset 1) and additional country details data (dataset 2) from the following website: https://data.worldbank.org/indicator/NY.GDP.PCAP.CD (https://data.worldbank.org/indicator/NY.GDP.PCAP.CD)

I have collected Health spending data of the OECD countries for the period 2000 to 2019 (dataset 3) from the following website: https://data.oecd.org/healthres/health-spending.htm (https://data.oecd.org/healthres/health-spending.htm)

Dataset 1

GDP per Capita

While reading the dataset, I had skipped the top four lines as those did not contain any relevant information.

The dataset contained 65 variables and 264 observations. The variables include the following:

- 'Country Name': Name of the Country
- 'Country Code': Three-letter country codes defined in ISO, a standardized way to represent a country name.
- 'Indicator Name': Indicator information - GDP per capita in current USD
- 'Indicator code': Code for the Indicator used by the Worldbank database
- 60 columns for GDP data by Years; from year 1960 to 2020
- Last two columns are empty, '2020' and 'X66' .

```
gdp <- read_csv("API_NY.GDP.PCAP.CD_DS2_en_csv_v2_1495171.csv", col_names = TRUE, skip = 4)
```

```
## Warning: Missing column names filled in: 'X66' [66]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `Country Name` = col_character(),
##   `Country Code` = col_character(),
##   `Indicator Name` = col_character(),
##   `Indicator Code` = col_character(),
##   `2020` = col_logical(),
##   X66 = col_logical()
## )
```

```
## See spec(...) for full column specifications.
```

```
head (gdp)
```

Country Name <chr>	**Country Code** <chr>	**Indicator Name** <chr>	**Indicator Code** <chr>	**1960** <dbl>
Aruba	ABW	GDP per capita (current US$)	NY.GDP.PCAP.CD	*NA*
Afghanistan	AFG	GDP per capita (current US$)	NY.GDP.PCAP.CD	59.77319
Angola	AGO	GDP per capita (current US$)	NY.GDP.PCAP.CD	*NA*
Albania	ALB	GDP per capita (current US$)	NY.GDP.PCAP.CD	*NA*

Country Name <chr>	Country Code <chr>	Indicator Name <chr>	Indicator Code <chr>	1960 <dbl>
Andorra	AND	GDP per capita (current US$)	NY.GDP.PCAP.CD	*NA*
Arab World	ARB	GDP per capita (current US$)	NY.GDP.PCAP.CD	*NA*

6 rows | 1-7 of 66 columns

- To select only the variables that I would need for my analysis, I considered only ‘Country Name’, ‘Country Code’, and columns from 2000 to 2019 years.
- Renamed ‘Country Code’, and ‘Country Name’.

```
gdp <- gdp %>%  select('Country Name', 'Country Code', '2000':'2019')
gdp <- gdp %>% rename('country_code' = 'Country Code', 'country_name'='Country Name')

head(gdp)
```

country_name <chr>	country_code <chr>	2000 <dbl>	2001 <dbl>	2002 <dbl>	2003 <dbl>	2004 <dbl>
Aruba	ABW	20620.7006	20669.0320	20436.8871	20833.7616	22569.9750
Afghanistan	AFG	*NA*	*NA*	179.4266	190.6838	211.3821
Angola	AGO	556.8363	527.3335	872.4945	982.9609	1255.5640
Albania	ALB	1126.6833	1281.6594	1425.1248	1846.1188	2373.5798
Andorra	AND	21854.2468	22971.5355	25066.8822	32271.9639	37969.1750
Arab World	ARB	2606.0766	2510.2619	2476.6960	2736.4821	3134.6735

6 rows | 1-8 of 22 columns

Dataset 2

Country details

The dataset contained:

- ‘Country Code’: Three-letter country codes defined in ISO, a standardized way to represent a country name.
- ‘Region’: Region of the country ‘IncomeGroup’: Classification of the country based on income as defined by the Worldbank
- ‘SpecialNotes’: Special notes about the country
- ‘TableName’: Contains name of the Country; and
- last column ‘X6’ was empty.

```
country <- read_csv("Metadata_Country_API_NY.GDP.PCAP.CD_DS2_en_csv_v2_1495171.csv", col_name
s = TRUE)
```

```
## Warning: Missing column names filled in: 'X6' [6]
```

```
## Parsed with column specification:
## cols(
##   `Country Code` = col_character(),
##   Region = col_character(),
##   IncomeGroup = col_character(),
##   SpecialNotes = col_character(),
##   TableName = col_character(),
##   X6 = col_logical()
## )
```

```
head (country)
```

Country Code <chr>	Region <chr>	IncomeGroup <chr>
ABW	Latin America & Caribbean	High income
AFG	South Asia	Low income
AGO	Sub-Saharan Africa	Lower middle income
ALB	Europe & Central Asia	Upper middle income
AND	Europe & Central Asia	High income
ARB	*NA*	*NA*

6 rows | 1-3 of 6 columns

To Select only the variable that I would need for my analysis, I had considered only ‘Country Code’, ‘Region’ and ‘IncomeGroup’.

```
country <- country %>%  select('Country Code', 'Region', 'IncomeGroup')
country <- country %>% rename('country_code' = 'Country Code')
head(country)
```

country_code <chr>	Region <chr>	IncomeGroup <chr>
ABW	Latin America & Caribbean	High income
AFG	South Asia	Low income
AGO	Sub-Saharan Africa	Lower middle income
ALB	Europe & Central Asia	Upper middle income
AND	Europe & Central Asia	High income
ARB	*NA*	*NA*

6 rows

Dataset 3

Health Spending of OECD Countries

The dataset contained:

- ‘LOCATION’: Three-letter country codes defined in ISO, a standardized way to represent a country name.
- ‘INDICATOR’: Indicator information i.e. HEALTHEXP- Health spend by a country in a year in USD per capita
- ‘SUBJECT’: Total level
- ‘MEASURE’: USD_CAP i.e. in USD per Capita
- ‘FREQUENCY’: A i.e. Yearly measure
- ‘TIME’: Year of measurement
- ‘Value’: Health spend value in USD
- ‘Flag Codes’: Contains code for each country

```
health <- read_csv("Health spending OECD.csv", col_names = TRUE)
```

```
## Parsed with column specification:
## cols(
##   LOCATION = col_character(),
##   INDICATOR = col_character(),
##   SUBJECT = col_character(),
##   MEASURE = col_character(),
##   FREQUENCY = col_character(),
##   TIME = col_double(),
##   Value = col_double(),
##   `Flag Codes` = col_character()
## )
```

```
head (health)
```

LOCATION <chr>	INDICATOR <chr>	SUBJE… <chr>	MEASU… <chr>	FREQUEN… <chr>	TI… <dbl>	Value <dbl>	Flag Codes <chr>
AUS	HEALTHEXP	TOT	USD_CAP	A	2000	2153.657	D
AUS	HEALTHEXP	TOT	USD_CAP	A	2001	2289.340	D
AUS	HEALTHEXP	TOT	USD_CAP	A	2002	2481.046	D
AUS	HEALTHEXP	TOT	USD_CAP	A	2003	2577.124	D
AUS	HEALTHEXP	TOT	USD_CAP	A	2004	2816.707	D
AUS	HEALTHEXP	TOT	USD_CAP	A	2005	2872.440	D

6 rows

- Considering the analysis objective, I considered only ‘LOCATION’, ‘TIME’ and ‘Value’.
- To maintain consistency across datasets and naming simplicity, I renamed ‘LOCATION’=‘country_code’, ‘TIME’=‘Year’ and ‘’Value’=‘health_expense’.

```
health <- health %>%  select('LOCATION', 'TIME', 'Value')

health <- health %>% rename('country_code' = 'LOCATION', 'Year'='TIME', 'health_expense'='Val
ue')

head(health)
```

country_code <chr>	Year <dbl>	health_expense <dbl>
AUS	2000	2153.657
AUS	2001	2289.340
AUS	2002	2481.046
AUS	2003	2577.124
AUS	2004	2816.707
AUS	2005	2872.440

6 rows

Understand

Dataset 1

GDP per Capita

Inspection of dataset 1 structure and attribute revealed that:

- 'country_name' and 'country_code' were of character class which was suitable. GDP values in each year columns from 2000 to 2019 were in numeric class and this was suitable for the analysis.
- So no data type conversion was required for dataset 1.

```
str (gdp)
```

```
## tibble [264 x 22] (S3: tbl_df/tbl/data.frame)
##  $ country_name: chr [1:264] "Aruba" "Afghanistan" "Angola" "Albania" ...
##  $ country_code: chr [1:264] "ABW" "AFG" "AGO" "ALB" ...
##  $ 2000        : num [1:264] 20621 NA 557 1127 21854 ...
##  $ 2001        : num [1:264] 20669 NA 527 1282 22972 ...
##  $ 2002        : num [1:264] 20437 179 872 1425 25067 ...
##  $ 2003        : num [1:264] 20834 191 983 1846 32272 ...
##  $ 2004        : num [1:264] 22570 211 1256 2374 37969 ...
##  $ 2005        : num [1:264] 23300 242 1902 2674 40066 ...
##  $ 2006        : num [1:264] 24045 264 2600 2973 42676 ...
##  $ 2007        : num [1:264] 25835 360 3122 3595 47804 ...
##  $ 2008        : num [1:264] 27085 365 4081 4371 48718 ...
##  $ 2009        : num [1:264] 24630 438 3123 4114 43503 ...
##  $ 2010        : num [1:264] 23513 543 3588 4094 40853 ...
##  $ 2011        : num [1:264] 24986 591 4615 4437 43335 ...
##  $ 2012        : num [1:264] 24714 642 5100 4248 38686 ...
##  $ 2013        : num [1:264] 26189 637 5255 4413 39539 ...
##  $ 2014        : num [1:264] 26648 614 5408 4579 41304 ...
##  $ 2015        : num [1:264] 27981 578 4167 3953 35763 ...
##  $ 2016        : num [1:264] 28281 547 3506 4124 37475 ...
##  $ 2017        : num [1:264] 29008 556 4096 4531 38963 ...
##  $ 2018        : num [1:264] NA 524 3290 5284 41793 ...
##  $ 2019        : num [1:264] NA 502 2974 5353 40886 ...
```

```
attributes(gdp)
```

```
## $names
##  [1] "country_name" "country_code" "2000"         "2001"         "2002"
##  [6] "2003"         "2004"         "2005"         "2006"         "2007"
## [11] "2008"         "2009"         "2010"         "2011"         "2012"
## [16] "2013"         "2014"         "2015"         "2016"         "2017"
## [21] "2018"         "2019"
##
## $row.names
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
## [109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## [127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
## [145] 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
## [163] 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
## [181] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
## [199] 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
## [217] 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234
## [235] 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
## [253] 253 254 255 256 257 258 259 260 261 262 263 264
##
## $class
## [1] "tbl_df"     "tbl"        "data.frame"
```

Dataset 2

Country details

Inspected dataset 2 structure and attribute.

```
str (country)
```

```
## tibble [263 x 3] (S3: tbl_df/tbl/data.frame)
##  $ country_code: chr [1:263] "ABW" "AFG" "AGO" "ALB" ...
##  $ Region      : chr [1:263] "Latin America & Caribbean" "South Asia" "Sub-Saharan Africa"
"Europe & Central Asia" ...
##  $ IncomeGroup : chr [1:263] "High income" "Low income" "Lower middle income" "Upper middl
e income" ...
```

```
attributes(country)
```

```
## $names
## [1] "country_code" "Region"       "IncomeGroup"
##
## $row.names
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
## [109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## [127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
## [145] 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
## [163] 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
## [181] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
## [199] 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
## [217] 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234
## [235] 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
## [253] 253 254 255 256 257 258 259 260 261 262 263
##
## $class
## [1] "tbl_df"     "tbl"        "data.frame"
```

Data inspection revealed that ‘country_code’ was of character class which was suitable. However, ‘Region’ should be in qualitative data and as I would need it to be ordered just alphabetically without any priority, so I had converted it to an unordered factor.

```
country$Region <- country$Region %>% as.factor()
class(country$Region)
```

```
## [1] "factor"
```

```
levels(country$Region)
```

```
## [1] "East Asia & Pacific"        "Europe & Central Asia"
## [3] "Latin America & Caribbean"  "Middle East & North Africa"
## [5] "North America"              "South Asia"
## [7] "Sub-Saharan Africa"
```

‘IncomeGroup’ should be in qualitative data and with a meaningful order. So, I had converted it to an ordered factored (from low to high income).

```
country$IncomeGroup <- factor(country$IncomeGroup, levels=c("Low income", "Lower middle incom
e",
                    "Upper middle income", "High income"), ordered=TRUE)
class(country$IncomeGroup)
```

```
## [1] "ordered" "factor"
```

```
levels(country$IncomeGroup)
```

```
## [1] "Low income"          "Lower middle income" "Upper middle income"
## [4] "High income"
```

Dataset 3

Health Spending of OECD Countries

- Inspected dataset 3 structure and attribute (*due to page constraint, not showed dataset attribute*).
- 'health_expense' was in numeric class and this was suitable for the analysis.

```
str (health)
```

```
## tibble [729 x 3] (S3: tbl_df/tbl/data.frame)
##  $ country_code  : chr [1:729] "AUS" "AUS" "AUS" "AUS" ...
##  $ Year          : num [1:729] 2000 2001 2002 2003 2004 ...
##  $ health_expense: num [1:729] 2154 2289 2481 2577 2817 ...
```

```
#attributes(health) - Due to page constraint, not showed dataset attribute.
```

- I had converted the 'country_code' and 'Year' to factor to keep these variable data types class compatible across datasets. That would also help me to analyse data by years and country_code at a later stage.
- Further, rounded off GDP values.

```
health$country_code <- health$country_code %>% as.factor()
health$Year <- health$Year %>% factor(levels = c('2000','2001','2002','2003','2004','2005','2006','2007','2008','2009','2010','2011','2012','2013','2014','2015','2016','2017','2018','2019'),ordered = TRUE)
class(health$country_code)
```

```
## [1] "factor"
```

```
class(health$Year)
```

```
## [1] "ordered" "factor"
```

```
levels(health$Year)
```

```
##  [1] "2000" "2001" "2002" "2003" "2004" "2005" "2006" "2007" "2008" "2009"
## [11] "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017" "2018" "2019"
```

Tidy & Manipulate Data I

There are three interrelated rules which make a dataset tidy (Wickham and Grolemund (2016)). In tidy data:

- Each variable must have its own column.
- Each observation must have its own row.
- Each value must have its own cell.

Data Inspection revealed that GDP dataset was untidy because GDP values' column headers were actually year values, not variable names. Therefore, the 'Year' variable did not had its own column. So each row represented twenty observations, not one. So I have transformed the data from wide to long format using the gather() function.

```
gdp <- gdp %>% gather(`2000`:`2019`, key = "Year",value = GDP)
head(gdp)
```

country_name <chr>	country_code <chr>	Year <chr>	GDP <dbl>
Aruba	ABW	2000	20620.7006
Afghanistan	AFG	2000	*NA*
Angola	AGO	2000	556.8363
Albania	ALB	2000	1126.6833
Andorra	AND	2000	21854.2468
Arab World	ARB	2000	2606.0766

6 rows

Dataset 2 ('country' - Country details) and Dataset 3 ('health' - Health Spending of OECD Countries) were fulfilling the Tidy data rules (Wickham and Grolemund (2016)). Hence, no further action needed to reshape these two datasets into a tidy format.

Merge Data

All three datasets were in the tidy format then and contained only the required variable in each dataset for the mentioned data analysis objective. After that I have merged three datasets :

- At first I merged Dataset 2 ('country' - Country details) to Dataset 1 ('gdp'- GDP per Capita) using left_join function.
- I have converted the 'country_code'. 'country_name' and 'Year' to factor to keep these variable data types class compatible across datasets.

```
gdp_country <- left_join(gdp, country, by = 'country_code')

gdp_country$country_code <- gdp_country$country_code %>% as.factor()
gdp_country$country_name <- gdp_country$country_name %>% as.factor()
gdp_country$Year <- gdp_country$Year %>% factor(levels = c('2000','2001','2002','2003','2004'
,'2005','2006','2007','2008','2009','2010','2011','2012','2013','2014','2015','2016','2017',
'2018','2019'),ordered = TRUE)

head(gdp_country)
```

country_name <fctr>	country_code <fctr>	Y… <ord>	GDP <dbl>	Region <fctr>	Incon
Aruba	ABW	2000	20620.7006	Latin America & Caribbean	Hig
Afghanistan	AFG	2000	*NA*	South Asia	Lov
Angola	AGO	2000	556.8363	Sub-Saharan Africa	Lower middl

country_name <fctr>	country_code <fctr>	Y… <ord>	GDP <dbl>	Region <fctr>	Incon
Albania	ALB	2000	1126.6833	Europe & Central Asia	Upper middl
Andorra	AND	2000	21854.2468	Europe & Central Asia	Hig
Arab World	ARB	2000	2606.0766	*NA*	

6 rows

After that, I wanted to keep only the observations related to the OECD countries. To achieve that, I had filtered rows of the 'country_code' column of the gdp_country dataset based on the values of the 'country_code' column of the health dataset (which contained only the county codes of OECD countries).

```
gdp_country <- gdp_country %>% filter(country_code %in% health$country_code)

head(gdp_country)
```

country_name <fctr>	country_code <fctr>	Y… <ord>	GDP <dbl>	Region <fctr>	IncomeGroup <ord>
Australia	AUS	2000	21679.248	East Asia & Pacific	High income
Austria	AUT	2000	24564.458	Europe & Central Asia	High income
Belgium	BEL	2000	23041.535	Europe & Central Asia	High income
Canada	CAN	2000	24190.250	North America	High income
Switzerland	CHE	2000	37868.323	Europe & Central Asia	High income
Chile	CHL	2000	5074.902	Latin America & Caribbean	High income

6 rows

- Next step, I merged Dataset 3 ('health' - Health Spending of OECD Countries) to gdp_country (merged data for Dataset 1 and 2) using left_join function using both 'country_code' and 'Year' as the key.
- Further, rounded off GDP and health_expense values to reduce clutter.

```
gdp_country_health <- left_join(gdp_country, health, by = c("country_code", "Year"))

gdp_country_health$GDP <- gdp_country_health$GDP %>% round()
gdp_country_health$health_expense <- gdp_country_health$health_expense %>% round()

head(gdp_country_health)
```

country_name <fctr>	country_code <fctr>	Y… <ord>	GDP <dbl>	Region <fctr>	IncomeGro… <ord>	health
Australia	AUS	2000	21679	East Asia & Pacific	High income	
Austria	AUT	2000	24564	Europe & Central Asia	High income	
Belgium	BEL	2000	23042	Europe & Central Asia	High income	

country_name <fctr>	country_code <fctr>	Y… <ord>	GDP <dbl>	Region <fctr>	IncomeGro… <ord>	health
Canada	CAN	2000	24190	North America	High income	
Switzerland	CHE	2000	37868	Europe & Central Asia	High income	
Chile	CHL	2000	5075	Latin America & Caribbean	High income	

6 rows

Tidy & Manipulate Data II

- Next, I wanted to calculate Health spending for each OECD country as a percentage of their GDP. To achieve that, I had divided GDP per capita ('GDP') by Health spending per capita ('health_expense'). This was an important indicator to provide insights on health economics that could enable further investigation of relationships with other health status indicators of a country.
- I had used the mutate() function to create that new variable ('Health_spending_GDP_share') and represented it as - Health spending percentage of GDP.

```
gdp_country_health<-gdp_country_health %>% mutate("Health_spending_GDP_share"= (health_expens
e/GDP))

gdp_country_health$Health_spending_GDP_share <- gdp_country_health$Health_spending_GDP_share*
100

gdp_country_health$Health_spending_GDP_share <- gdp_country_health$Health_spending_GDP_share
 %>% round(2)
head(gdp_country_health)
```

country_name <fctr>	country_code <fctr>	Y… <ord>	GDP <dbl>	Region <fctr>	IncomeGro… <ord>	health
Australia	AUS	2000	21679	East Asia & Pacific	High income	
Austria	AUT	2000	24564	Europe & Central Asia	High income	
Belgium	BEL	2000	23042	Europe & Central Asia	High income	
Canada	CAN	2000	24190	North America	High income	
Switzerland	CHE	2000	37868	Europe & Central Asia	High income	
Chile	CHL	2000	5075	Latin America & Caribbean	High income	

6 rows | 1-7 of 8 columns

Scan I

Checking for Missing values

Checked data for missing values in each column using colSums() function.

```
colSums(is.na(gdp_country_health))
```

```
##              country_name              country_code                      Year
##                         0                         0                         0
##                       GDP                    Region               IncomeGroup
##                         0                         0                         0
##            health_expense Health_spending_GDP_share
##                        11                        11
```

- The combined dataset () had 11 missing values in the 'health_expense' and 'Health_spending_GDP_share' column.
- To deal with these missing values, I first needed to understand the nature of these missing values. So I had filtered rows with any missing value:

```
data_missingvalues <- gdp_country_health[rowSums(is.na(gdp_country_health)) > 0,]
data_missingvalues
```

country_name <fctr>	country_code <fctr>	Y… <ord>	G… <dbl>	Region <fctr>	IncomeGroup <ord>
Colombia	COL	2000	2520	Latin America & Caribbean	Upper middle income
Colombia	COL	2001	2440	Latin America & Caribbean	Upper middle income
Colombia	COL	2002	2397	Latin America & Caribbean	Upper middle income
Colombia	COL	2003	2281	Latin America & Caribbean	Upper middle income
Colombia	COL	2004	2783	Latin America & Caribbean	Upper middle income
Colombia	COL	2005	3414	Latin America & Caribbean	Upper middle income
Colombia	COL	2006	3741	Latin America & Caribbean	Upper middle income
Colombia	COL	2007	4714	Latin America & Caribbean	Upper middle income
Colombia	COL	2008	5473	Latin America & Caribbean	Upper middle income
Colombia	COL	2009	5193	Latin America & Caribbean	Upper middle income

1-10 of 11 rows | 1-7 of 8 columns Previous **1** 2 Next

- So, the dataset did not contain values in 'health expense' and corresponding 'Health_spending_GDP_share' columns for Colombia from the period 2000 to 2010.
- One way to deal with these missing values would be to recode these missing values with the mean values of Colombia country with the available data; i.e. 2011 to 2020. I tried to investigate if that would be an appropriate thing to do.
- So, calculated mean health expense for the period 2000 to 2010 and 2011 to 2020 for each country.

```
#mean expense for the period 2000 to 2010
summary1 <- gdp_country_health %>% filter( Year < 2011) %>% group_by(country_name) %>% summar
ise(
                                    Mean = mean(health_expense, na.rm = TRUE))
```

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

```
summary1
```

country_name <fctr>	Mean <dbl>
Australia	2886.3636
Austria	3512.6364
Belgium	3109.8182
Canada	3294.8182
Chile	896.7273
Colombia	NaN
Czech Republic	1539.2727
Denmark	3181.8182
Estonia	924.0909
Finland	2640.7273

1-10 of 37 rows Previous **1** 2 3 4 Next

```
#mean expense for the period 2011 to 2020
summary2 <- gdp_country_health %>% filter( Year > 2010) %>% group_by(country_name) %>% summar
ise(
                                        Mean = mean(health_expense, na.rm = TRUE))
```

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

```
summary2
```

country_name <fctr>	Mean <dbl>
Australia	4421.778
Austria	5049.667
Belgium	4693.667
Canada	4768.111
Chile	1807.444
Colombia	1024.000
Czech Republic	2650.111
Denmark	4774.556
Estonia	1929.667

country_name <fctr>	Mean <dbl>
Finland	4051.333

1-10 of 37 rows Previous **1** 2 3 4 Next

- Comparing the mean values of each country for the above mentioned two periods, it is clear that it had changed drastically over the years. Hence I should not recode the missing values with the mean values of available data for Colombia. Therefore, I had decided to omit the records with missing values from the analysis.
- Further, as it would impact only 11 observations out of 740 observations, the loss of data would not be significant and would not result in a biased subset of the data.
- So, all the missing values had been dealt with.

```
gdp_country_health <-na.omit(gdp_country_health)
colSums(is.na(gdp_country_health))
```

```
##              country_name              country_code                      Year
##                         0                         0                         0
##                       GDP                    Region               IncomeGroup
##                         0                         0                         0
##           health_expense Health_spending_GDP_share
##                         0                         0
```

Checking for Special values

After that, I had checked each numerical column for any special values like inf, NaN using a function called is.special.

```
is.special <- function(x){
if (is.numeric(x)) (is.infinite(x) | is.nan(x))
}

# apply this function to the dataset and calculate the total missing values for each column
sapply(df, is.special)
```

```
## $x
## NULL
##
## $df1
## NULL
##
## $df2
## NULL
##
## $ncp
## NULL
##
## $log
## NULL
##
## [[6]]
## NULL
```

```
gdp_country_health %>% sapply(function(x) sum(is.special(x)))
```

```
##              country_name              country_code                      Year
##                         0                         0                         0
##                       GDP                    Region               IncomeGroup
##                         0                         0                         0
##            health_expense Health_spending_GDP_share
##                         0                         0
```

So there are no special values in the dataset.

Checking for obvious Inconsistencies or Errors

An obvious inconsistency occurs when a data record contains a value that cannot correspond to a real-world situation. I had defined rules to identify obvious inconsistencies or errors on the variables using editset functions and after that checked the datasets against these rules using the violatedEdits function.

In this case, GDP and Health Expense could not be negative.

```
rule1 <- editset(c("GDP >= 0", "health_expense >= 0"))

sum(violatedEdits(rule1, gdp_country_health))
```

```
## [1] 0
```

- So, no observation violated these rules set for GDP and Health Expense.
- Next check, Health Expense could not be less than or equal to GDP as ‘health expense’ is a sub-set of ‘GDP’. So ‘Health_spending_GDP_share’ cannot be greater than or equal to 100.

```
rule2 <- editset(c("health_expense<GDP","Health_spending_GDP_share < 100"))
sum(violatedEdits(rule2, gdp_country_health))
```

```
## [1] 0
```

I had written rules in a text file to ensure ‘IncomeGroup’, ‘Region’ and ‘Year’ variable contain only expected values and used editfile function to read rules directly from the text file.

```
Rules <- editfile("editrules.txt", type = "all")
Rules
```

```
##
## Data model:
## dat1 : IncomeGroup %in% c('High income', 'Low income', 'Lower middle income', 'Upper middl
e income')
## dat2 : Region %in% c('East Asia & Pacific', 'Europe & Central Asia', 'Latin America & Cari
bbean', 'Middle East & North Africa', 'North America', 'South Asia', 'Sub-Saharan Africa')
## dat3 : Year %in% c('2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008',
'2009', '2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019')
##
## Edit set:
## NULL :
```

and, checked summary of violated rules. So no errors detected.

```
Violated <- violatedEdits(Rules, gdp_country_health)
summary(Violated)
```

```
## No violations detected, 0 checks evaluated to NA
```

```
## NULL
```

- The OECD is an economic organisation with 37 member countries, so checked if I had exactly the same numbers of countries and corresponding country codes in the dataset.
- I had considered 20 years (2000 to 2019) of data, so checked if I had exactly 20 unique values in the 'Year' column.
- No errors detected.

```
length(unique(gdp_country_health[["country_name"]])) == 37
```

```
## [1] TRUE
```

```
length(unique(gdp_country_health[["country_code"]])) == 37
```

```
## [1] TRUE
```

```
length(unique(gdp_country_health[["Year"]])) == 20
```

```
## [1] TRUE
```

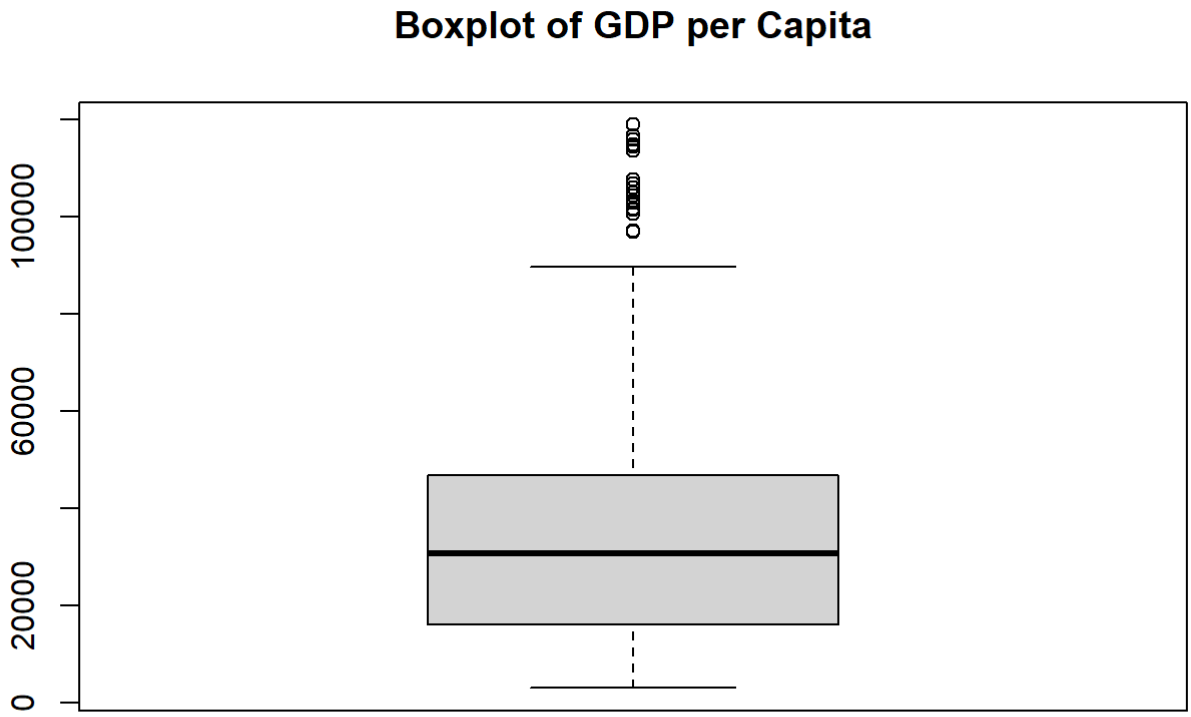
Scan II

Next, I scanned numeric data 'health_expense' and 'GDP' for any potential outlier. I did not need to scan 'Health_spending_GDP_share' variable as it was a derived ratio and would have acceptable values as long as underlying 'health_expense' and 'GDP' data were issue-free.

To understand the presence of univariate outliers (assuming the population distribution of the underlying data is unknown):

- Plotted Boxplot to understand the data and identify potential outliers with −1.5×IQR and 1.5×IQR outlier fences. In that, any value lying outside the fences were depicted as an outlier.
- Identified the values of the potential outliers using the boxplot.stats()$out function

```
boxplot(gdp_country_health$GDP, main="Boxplot of GDP per Capita")
```

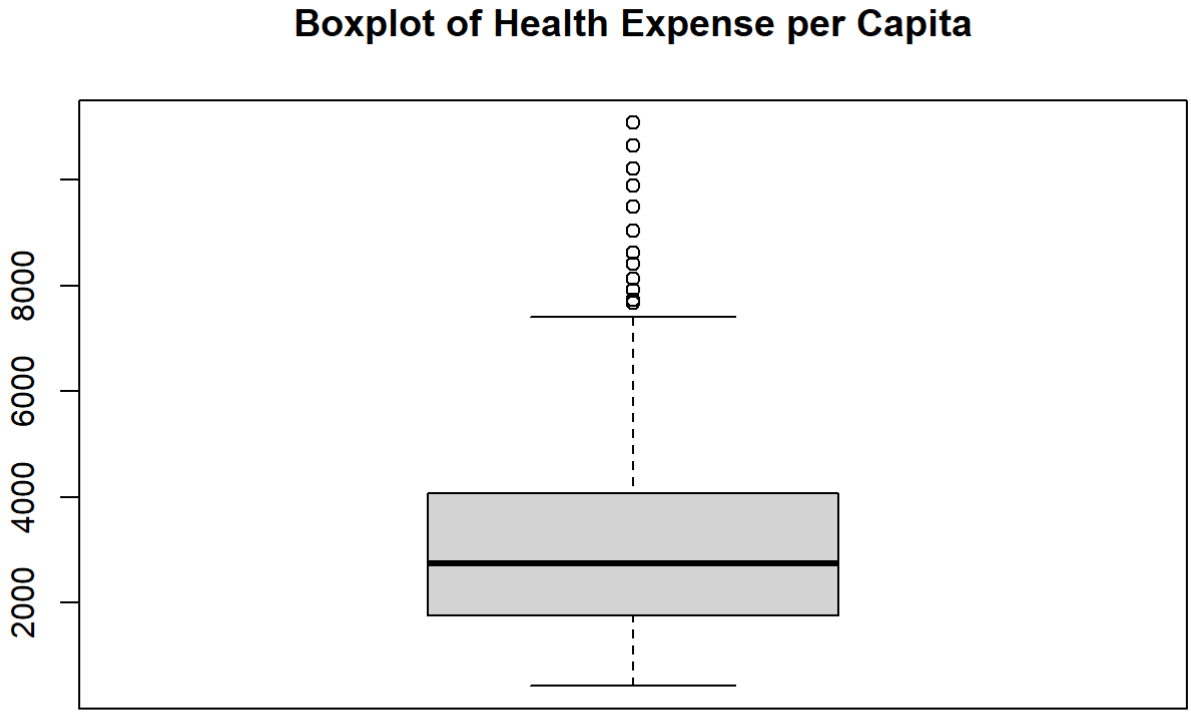



```
#Inspecting outliers in the GDP data
potential_GDP_outliers <- boxplot.stats(gdp_country_health$GDP)$out %>% round ()
potential_GDP_outliers
```

```
##  [1] 106018 114294  96944 103199 104965 115762 100601 106749 101524 113625
## [11] 102913 118824  97019 101376 104278 107627 116654 114705
```

Plotted boxplot for potential Health Expense per Capita outliers.

```
boxplot(gdp_country_health$health_expense, main="Boxplot of Health Expense per Capita")
```




```
##Inspecting outliers in GDP per Capita
potential_health_outliers <- boxplot.stats(gdp_country_health$health_expense)$out %>% round
 ()
potential_health_outliers
```

```
##  [1]  7670  7922  8131  8405  8611  9034  9498  9880 10213 10637  7732 11072
```

- So there could be some potential abnormal cases.
- I had used which() function to extract the row number corresponding to these potential GDP outliers and inspected the specific rows in the dataset to verify them.
- It was interesting as we know Luxembourg and Norway both are energy exporters, regional financial centers, and export business powerhouses with relatively small populations, might resulted in high GDP per Capita.

```
GDP_out <- which(gdp_country_health$GDP %in% c(potential_GDP_outliers))
GDP_out_dataset <- gdp_country_health[GDP_out, ]
GDP_out_dataset
```

country_name <fctr>	country_code <fctr>	Y… <ord>	GDP <dbl>	Region <fctr>	IncomeGro… <ord>	health_exp
Luxembourg	LUX	2007	106018	Europe & Central Asia	High income	
Luxembourg	LUX	2008	114294	Europe & Central Asia	High income	
Norway	NOR	2008	96944	Europe & Central Asia	High income	

country_name <fctr>	country_code <fctr>	Y… <ord>	GDP <dbl>	Region <fctr>	IncomeGro… <ord>	health_ex
Luxembourg	LUX	2009	103199	Europe & Central Asia	High income	
Luxembourg	LUX	2010	104965	Europe & Central Asia	High income	
Luxembourg	LUX	2011	115762	Europe & Central Asia	High income	
Norway	NOR	2011	100601	Europe & Central Asia	High income	
Luxembourg	LUX	2012	106749	Europe & Central Asia	High income	
Norway	NOR	2012	101524	Europe & Central Asia	High income	
Luxembourg	LUX	2013	113625	Europe & Central Asia	High income	

1-10 of 18 rows | 1-7 of 8 columns Previous **1** 2 Next

- Conducted the same investigation for potential health expense outliers and inspected the specific rows in the dataset to verify them.
- It revealed that potential Health Expense per Capita outliers were present only in data points associated with the United States and Switzerland. It is common knowledge that both the mentioned countries spend huge amount in the health sector.

```
health_out <- which(gdp_country_health$health_expense %in% c(potential_health_outliers))
gdp_country_health[health_out, ]
```

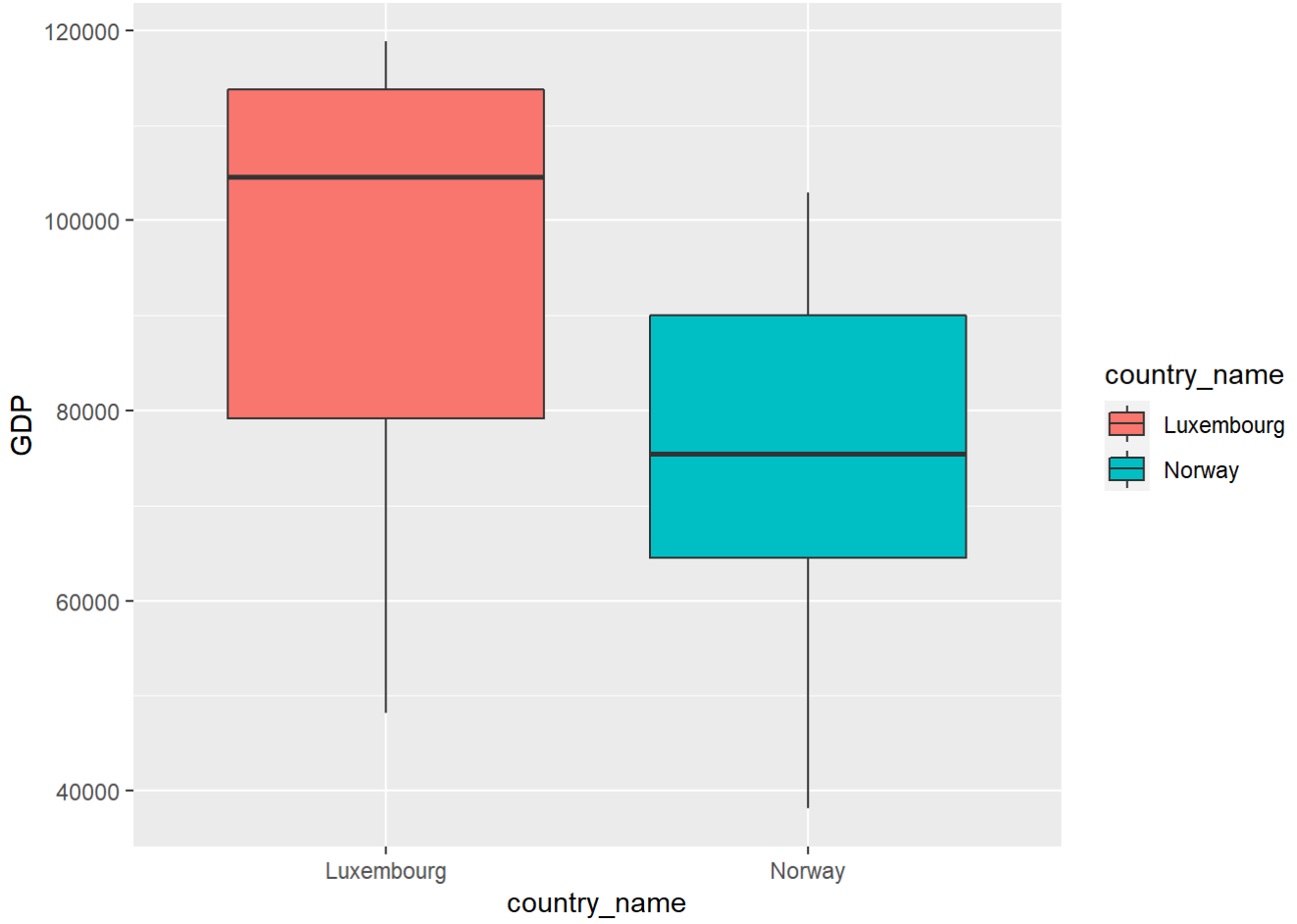
country_name <fctr>	country_code <fctr>	Y… <ord>	GDP <dbl>	Region <fctr>	IncomeGro… <ord>	health_ex
United States	USA	2009	47100	North America	High income	
United States	USA	2010	48468	North America	High income	
United States	USA	2011	49887	North America	High income	
United States	USA	2012	51611	North America	High income	
United States	USA	2013	53118	North America	High income	
United States	USA	2014	55048	North America	High income	
United States	USA	2015	56823	North America	High income	
United States	USA	2016	57928	North America	High income	
United States	USA	2017	59958	North America	High income	
United States	USA	2018	62840	North America	High income	

1-10 of 12 rows | 1-7 of 8 columns Previous **1** 2 Next

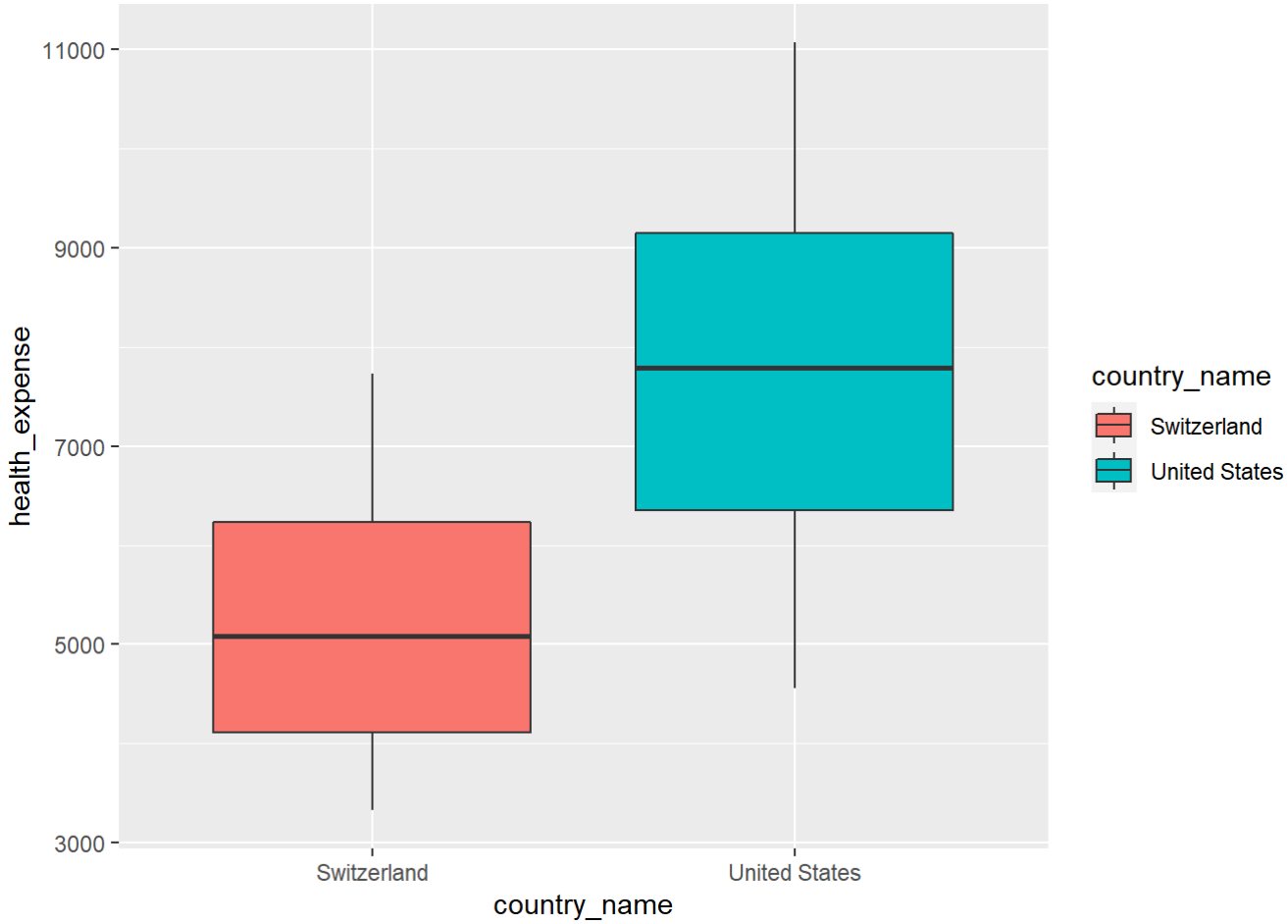
- This called for further investigation using a bivariate box plot.
- It revealed that there were no obvious outliers or abnormal observations when we included the respective country into consideration.

```
GDP_out <- gdp_country_health %>%  filter( country_name %in% c('Luxembourg', 'Norway'))
health_out <- gdp_country_health %>%  filter( country_name %in% c('United States', 'Switzerla
nd'))

library(ggplot2)
ggplot(GDP_out, aes(country_name, GDP, fill=country_name)) +
  geom_boxplot()
```




```
ggplot(health_out, aes(country_name, health_expense, fill=country_name)) +
  geom_boxplot()
```



- Upon inspecting each of the potential outliers' values, it was clear that none of those were extreme cases. Those outlier's values were legitimate observations, very much possible in the real world, and would provide greater insights during analysis and so I should leave it in the dataset. It would not be appropriate practice to remove valid data points simply to produce a better fitting model or statistically significant results in the later analysis stage. These potential outliers were caused more likely due to natural variation instead of data entry or measurement errors, sampling problems, or unusual conditions.
- Hence, I reached a conclusion that there was no justifiable reason to remove these outliers or replace these outliers with mean or median. Replacing the outliers with the nearest neighbours that were not outliers (Capping or winsorising) would also not be appropriate.
- Next, using Central Limit Theorem for a large sample size, I could assume that the distribution for GDP and Health expense data were approximately normal. So I could use the z-score approach to detect outliers with the help of 'outliers' package.

```
z_score_GDP <- gdp_country_health$GDP %>%  scores(type = "z")
z_score_GDP %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.3974 -0.8089 -0.1436  0.0000  0.5827  3.8544
```

```
which( abs(z_score_GDP) >3 )
```

```
##  [1] 276 312 348 384 421 425 458 462 495 499 532 569 606 643 680 717
```

```
z_score_health <- gdp_country_health$health_expense %>%  scores(type = "z")
z_score_health %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.5295 -0.7301 -0.1463  0.0000  0.6459  4.8416
```

```
which( abs(z_score_health) >3 )
```

```
## [1] 433 470 507 544 581 618 655 692 729
```

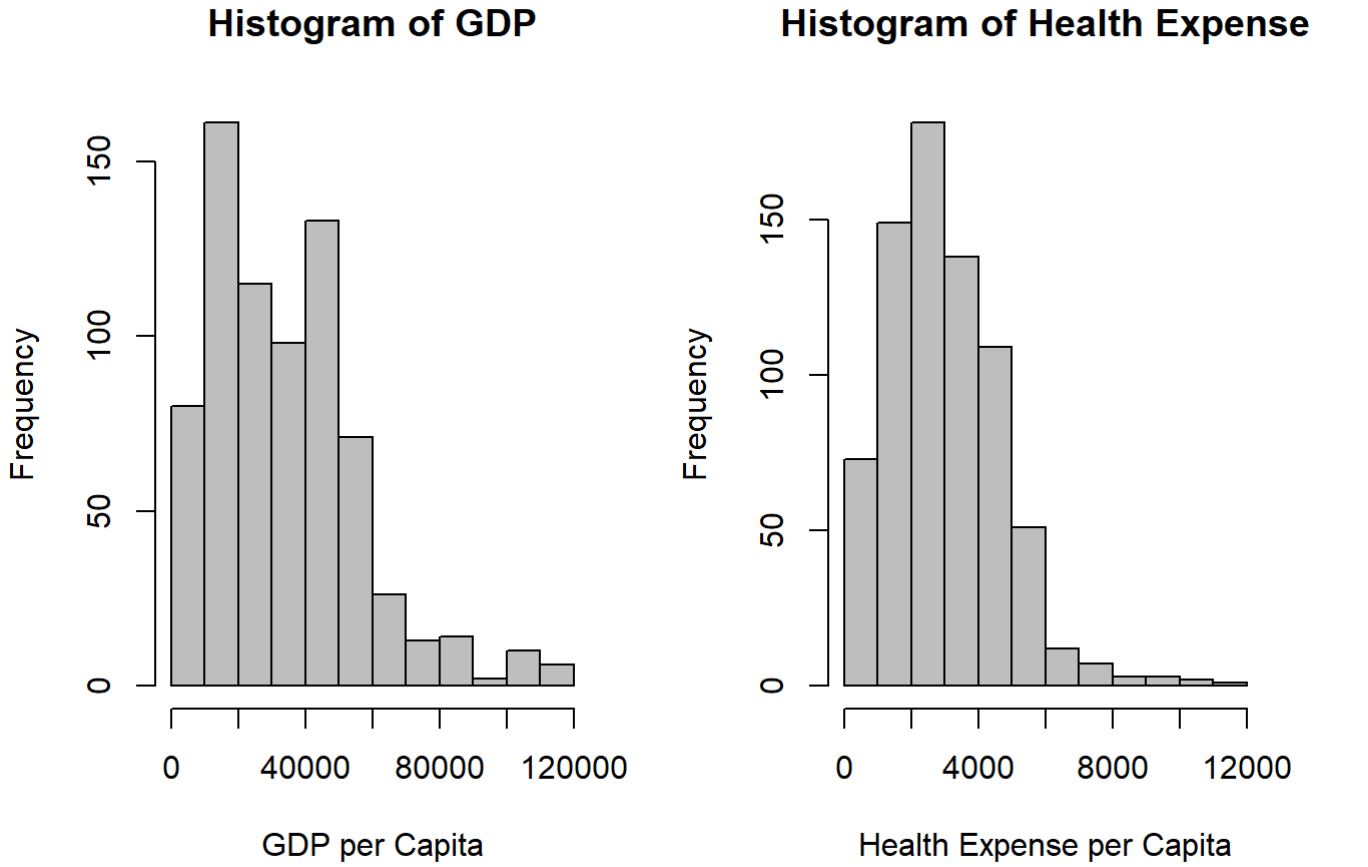
From the summary() output Max. values, I could see the presence of outliers. Further, which() function showed the locations of the z-scores whose absolute value is greater than 3.

However, in this case, as well, using the above mentioned rationale and analysis, I could conclude that there was no justifiable reason to remove or replace those outliers. So I had kept the outliers unchanged.

Transform

As one of the key objectives of my data analysis was to understand the relationship between GDP and Health Spend, I would prefer the distribution of those variables to be normal. So I checked the type of skewness in the data before using any transformation to reduce the skewness.

```
par(mfrow=c(1,2))
gdp_country_health$GDP %>%  hist(col="grey", xlab="GDP per Capita", main="Histogram of GDP")
gdp_country_health$health_expense %>%  hist(col="grey", xlab="Health Expense per Capita", mai
n="Histogram of Health Expense")
```



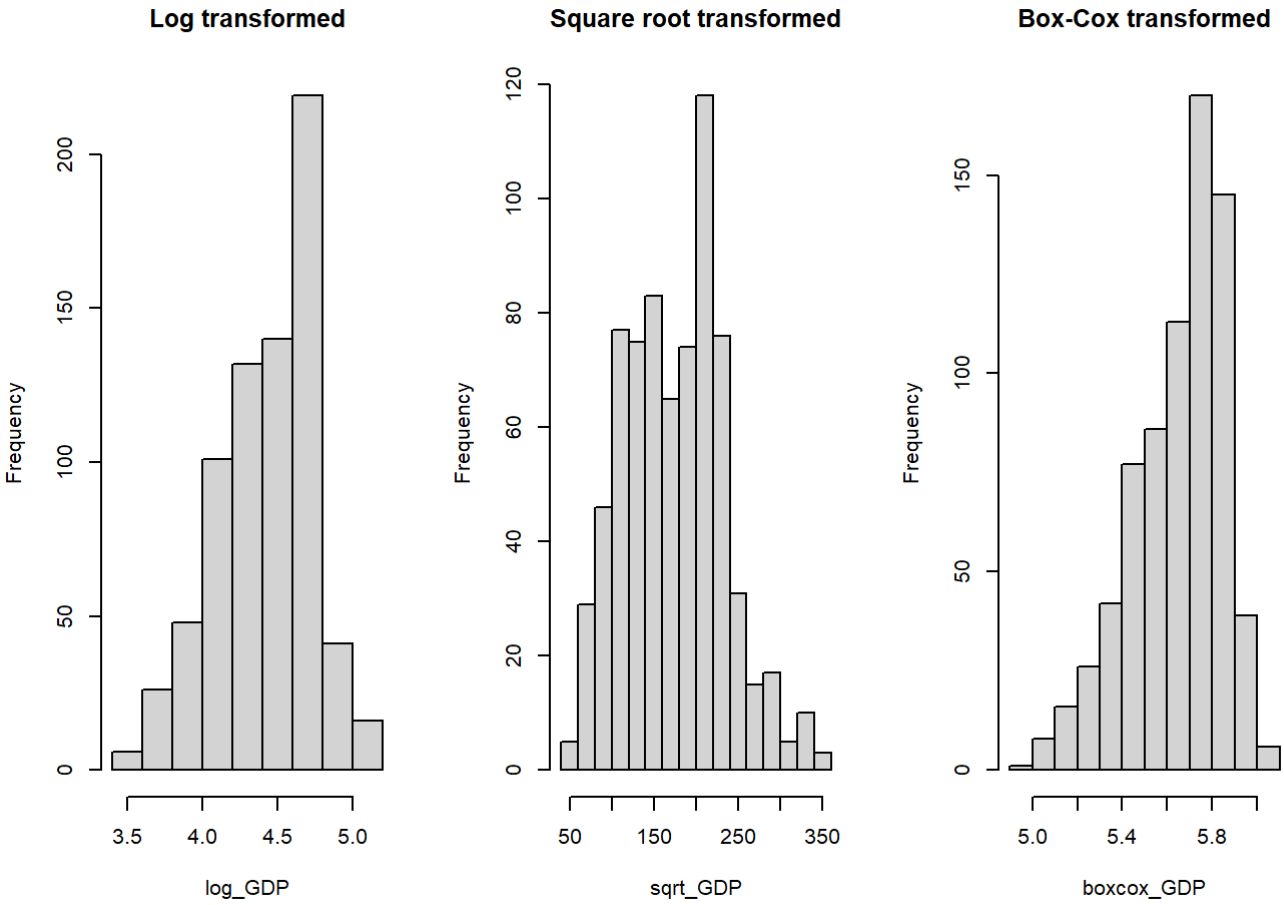
- I observed that both GDP and Health expense had a right-skewed distribution.
- So, I had applied different data transformation methods and showed below the top three outcomes (i.e. Log transformation, Square root transformation, and Box-Cox Transformation).

```
par(mfrow=c(1,3))

log_GDP <- log10(gdp_country_health$GDP)
hist(log_GDP, main="Log transformed")

sqrt_GDP <- sqrt(gdp_country_health$GDP)
hist(sqrt_GDP, main="Square root transformed")

boxcox_GDP<- BoxCox(gdp_country_health$GDP,lambda = "auto")
hist(boxcox_GDP, main="Box-Cox transformed")
```



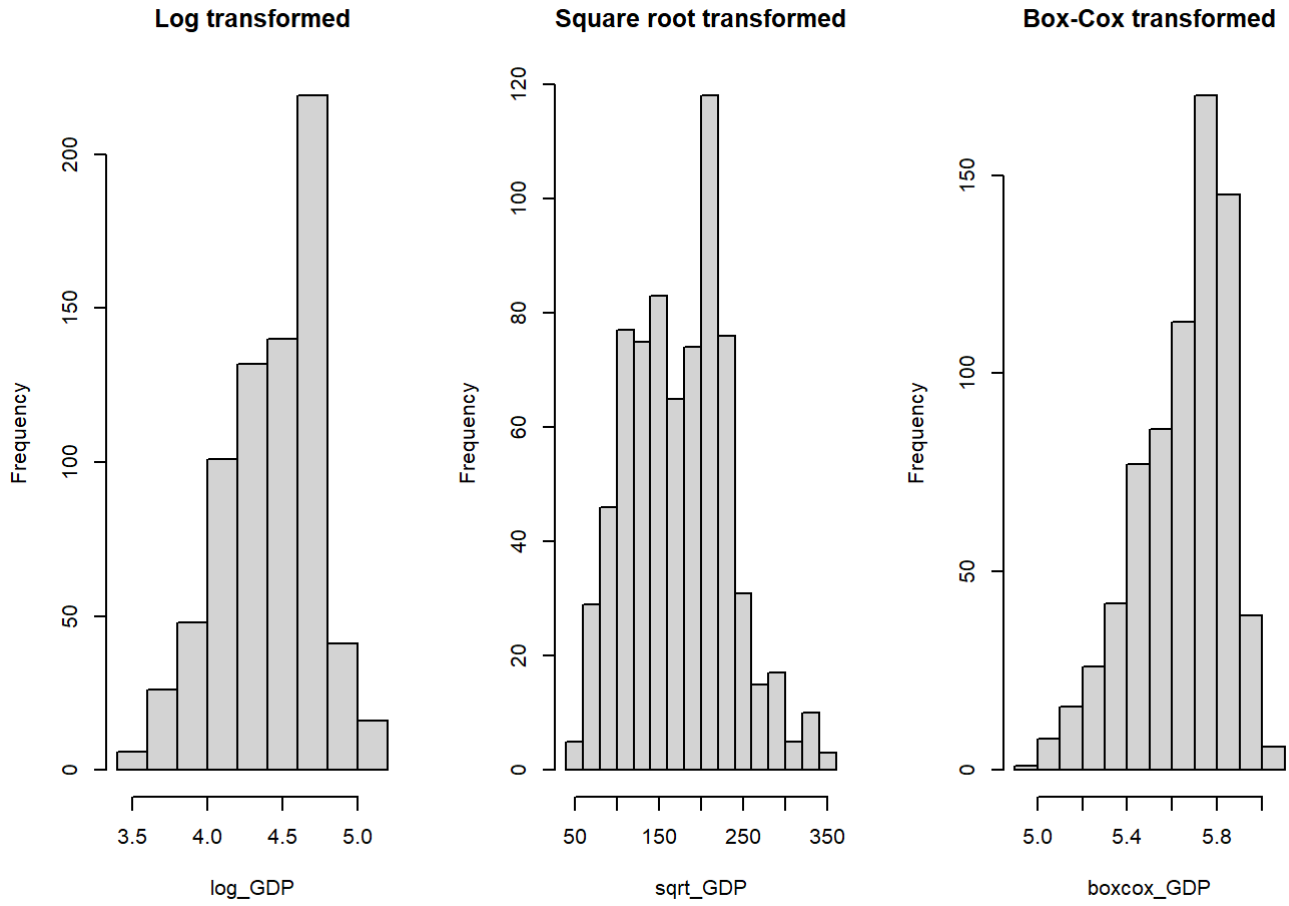
- The Log transformation had shown relatively better performance in terms of reducing the skewness in the GDP per capita data.
- Similarly, I had applied different transformation on the health expense variable and showed below the top three output:

```
par(mfrow=c(1,3))

log_health <- log10(gdp_country_health$health_expense)
hist(log_GDP, main="Log transformed")

sqrt_health <- sqrt(gdp_country_health$health_expense)
hist(sqrt_GDP, main="Square root transformed")

boxcox_health <- BoxCox(gdp_country_health$health_expense,lambda = "auto")
hist(boxcox_GDP, main="Box-Cox transformed")
```



In this case, as well, the Log transformation had shown relatively better performance in terms of reducing the skewness in the Health Expense per capita data. So, during the analysis stage, if needed, I would use the Log transformation.

So, the final dataset is ready for analysis.