Case Study 1

Analyzing the GDP of The Top 190 Countries

Trace Smith

November 4, 2016

Introduction

The objective of this work is to gather two separates data files, read both files into separate data frames, clean up the data, and then perform a horizontal merge of the two data frames. The final task is to then take the merged data and conduct various statistical analysis using R. As an overview of the raw data, the Gross Domestic Product Ranking data set consists of 327 rows and 10 columns while the Education Statistics contains 31 different features with 234 rows. Both data sets were downloaded from the World Bank website and then read into respected data frames. From here, several steps were taken to tidy the data before performing a merge on the key variable "CountryCode". After preprocessing the data, the ultimate goal is to be able to extract information to answer several specific questions. The R code to address the following inquiries can be found in the Analysis directory under the file name `analysis.r`:

1. Merge the data based on the country shortcode. How many of the IDs match?
2. Sort the data frame in ascending order by GDP (so United States is last). What is the 13th country in the resulting data frame?
3. What are the average GDP rankings for the "High income: OECD" and "High income: nonOECD" groups?
4. Plot the GDP for all of the countries. Use ggplot2 to color your plot by Income Group.
5. Cut the GDP ranking into 5 separate quantile groups. Make a table versus Income.Group.
6. How many countries are Lower middle income but among the 38 nations with highest GDP?

R Packages

- The listed R packages below are required to execute the source code. If these packages are not installed, you can install the packages by `install.packages("name")` in the R console.

```
#Load R Library
library(plyr)
library(ggplot2)
library(downloader)
```

Set Working Directory

Note: The working directory should be set to: "/Case Study 1/Data":

```
dir <- "/Users/Case Study 1/Data" #file path
setwd(dir) #set working directory
```

Run Source Code

- In the Analysis directory, gathering of data, data cleansing, merging the data, and code to analyze the data can be executed by the running the file `analysis.r` (i.e. located in the Analysis directory). Using the `source` function, the R files are linked together and thus can be run all at once via the `analysis.r` file. Here is a brief summary of the files implemented in this work:

- `gather.r`: downloads the data from the internet
- `tidy.r`: takes the two data frames and cleans the rows/columns to prepare for statistical analysis
- `merge.r`: takes the cleaned data and then merges two data frames horizontally utilizing the unique key identifier "CountryCode"
- `analysis.r`: takes the merged data and answers the five questions as previously defined in the introduction

```
#Read R code from analysis.r -- linked to gather, tidy, merge, and analysis
#Can execute the entire R script from the code below:
source("../Analysis/analysis.r")
```

Downloading the Dataset

- There are two options when downloading both data sets. First option is shown below where we can specify the URL link to the source and then download the file directly from the website. Furthermore, note that `destfile` refers to directory the file will be stored in along with the name of the file (i.e. GDP.raw.csv). Again, the important piece to remember here is setting the working directory to the working directory to `Case Study 1/Data`. The source code for downloading the data from the World Bank's website and saving to the "Data" directory can be found in the `gather.r` file.

- **Note:** The data can be downloaded by executing the `analysis.r` file as discussed in the previous section. The download function utilized in the `gather.r` file to download the two data sets from the internet is shown below:

```
#Download Data GDP Ranking:
url<-
"https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FEDSTATS_Country.csv"
#URL to data
download(url,destfile="FEDSTATS_Country.raw.csv") #download file and save it
to Data directory

#Download Educational Data:
url<-"https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FGDP.csv" #URL to
```

```
data
download(url,destfile="FGDP.raw.csv") #download file and save it to Data
directory
```

- The second option is to download the file directly from The World Bank website directly and then move the file from the local download directory to the "Data" directory (i.e. a sub-directory of "Case Study 1").

Tyding the Data

- The following code blocks will walk through the R code utilized to clean up the messy data for each data frame. As a note here, the NA's will not be removed during the cleaning process, rather both empty observations (i.e. "") and all NA's will be removed in the following section when merging the data together. Let's briefly explore the two data sets below.

Gross Domestic Product Data

- **Explore the Data**: In this section, we will take a look at the GDP data set (i.e. `FGDP.raw.csv`). Before transforming the data, let's examine the raw data first. The downloaded csv file will need to be read into a data frame, which is a convenient way of storing large data sets in a table format.

```
#Read GDP Dataset into dataframe
gdp.raw <- read.csv("FGDP.raw.csv",header=TRUE,skip=3)
```

- The dimensions of the raw GDP data consist of 10 columns and 327 rows.

```
#Dimensions
dim(gdp.raw)

## [1] 327  10
```

- **Data Cleaning** - Before making any changes to the raw data set, let's now assign the raw data to a new object with the name `gdp` in order to preserve the raw data.

```
#Assign Raw Data to "gdp"
gdp <-gdp.raw
knitr::kable(head(gdp,5))
```

X	Ranking	X.1	Economy	US.dollars.	X.2	X.3	X.4	X.5	X.6
		NA				NA	NA	NA	NA
USA	1	NA	United States	16,244,600		NA	NA	NA	NA
CHN	2	NA	China	8,227,103		NA	NA	NA	NA
JPN	3	NA	Japan	5,959,718		NA	NA	NA	NA
DEU	4	NA	Germany	3,428,131		NA	NA	NA	NA

- Next, as we can see from the output above, there are a total of 6 columns with missing data (i.e. NA), thus we will delete these columns. A quick way of dropping these columns would be to assign `NULL` to the respected index column of the "gdp" data frame.

```
#Drop Columns
gdp[6:10] <-NULL
gdp[[3]] <- NULL
```

- A quick check to see if these columns have been dropped is to list out the header names for the data frame. Only 4 columns remain, thus the code worked as expected.

```
#List the header name for each column
names(gdp)

## [1] "X"           "Ranking"     "Economy"     "US.dollars."
```

- Shown above in the code block that prints out each header name, the first column is labeled "X1" but this is actually the "CountryCode"; therefore the column will be renamed accordingly. Likewise, the "US.dollars." column of the raw data set will also be renamed to "GDP", abbreviated for Gross Domestic Product.

```
#Rename Column
colnames(gdp)[1] <- "CountryCode"
colnames(gdp)[4] <- "GDP"
```

- Similar to as before, to verify if the header has been correctly renamed, call the `names` function:

```
#List the header name for each column
names(gdp)

## [1] "CountryCode" "Ranking"     "Economy"     "GDP"
```

- Examining the raw data set using the following function, `str(gdp.raw)`, it was noticed that rows 217 and beyond were irrelevant to the analysis (i.e. contained unstructured text information and world GDP statistics) and therefore was excluded from the data frame.

```
#Select Certain row and all columns;
#not including the section of text below the data and world GDP info
gdp<-gdp[(2:216),] #select rows 2:215 and all columns (,)
```

- A useful function `str()`, provides information about the structure of the data. Shown below, columns "Ranking" and "GDP" are both factors and will need to be converted to numeric values in order to perform numerical calculations or any type of analysis.

```
str(gdp$GDP) #structure of GDP column

##  Factor w/ 205 levels "",".. 1,008 "," 1,129 ",..: 40 178 143 100 66 63 61
58 57 16 ...

str(gdp$Ranking) #structure of Ranking column

##  Factor w/ 194 levels "",".. Not available.  ",..: 3 104 115 126 137 148
159 170 181 4 ...
```

- The script to convert these columns (i.e. factors) to numeric is as follows:

```r
# pattern "[^[:digit:]]" refers to members of the variable name that start
with digits.
# gsub command to replace them with a blank space
# convert variables to numeric
gdp$GDP <- as.numeric(gsub("[^[:digit:]]","", gdp$GDP))
gdp$Ranking <- as.numeric(gsub("[^[:digit:]]","", gdp$Ranking))
```

- In this code block, the output will allow us to look over the data frame and make sure everything looks correct before moving forward.

```r
#return the first 5 rows of the gdp data frame
knitr::kable(head(gdp,5))
```

	CountryCode	Ranking	Economy	GDP
2	USA	1	United States	16244600
3	CHN	2	China	8227103
4	JPN	3	Japan	5959718
5	DEU	4	Germany	3428131
6	FRA	5	France	2612878

- Note: before removing any rows containing `NA` for the "Ranking" column, the code below will count the total number of NA observations per each column.

```r
#return the number of columns containing NA
head(colSums(is.na(gdp)))

## CountryCode     Ranking     Economy         GDP
##           0          25           0          25
```

Educational Data

- **Explore the Data**: Now, let's take a look at the Educational data (i.e. `FEDSTATS_Country.raw`). Likewise, the downloaded csv file will be read into a data frame; note the raw data file will be preserved by assigning it to a new object "fedstats".

```r
#Read FEDSTATS Dataset into datframe
fedstats.raw <- read.csv("FEDSTATS_Country.raw.csv",header=TRUE)

#Create New DataFrame From Raw
fedstats<-fedstats.raw
```

- The dimensions of the raw data file are 31 columns and 234 rows

```r
#Dimensions (row,columns)
dim(fedstats.raw)

## [1] 234  31
```

- The output from this code block will return the column name of each feature and the corresponding number of observations containing NA (per column).

```r
#The number of N/A per column
knitr::kable(colSums(is.na(fedstats)))
```

CountryCode	0
Long.Name	0
Income.Group	0
Region	0
Lending.category	0
Other.groups	0
Currency.Unit	0
Latest.population.census	0
Latest.household.survey	0
Special.Notes	0
National.accounts.base.year	0
National.accounts.reference.year	197
System.of.National.Accounts	149
SNA.price.valuation	0
Alternative.conversion.factor	0
PPP.survey.year	89
Balance.of.Payments.Manual.in.use	0
External.debt.Reporting.status	0
System.of.trade	0
Government.Accounting.concept	0
IMF.data.dissemination.standard	0
Source.of.most.recent.Income.and.expenditure.data	0
Vital.registration.complete	0
Latest.agricultural.census	0
Latest.industrial.data	139
Latest.trade.data	46
Latest.water.withdrawal.data	82
X2.alpha.code	1
WB.2.code	1
Table.Name	0
Short.Name	0

- **Data Cleaning** - For this data set, there were no major steps necessary to tidy up the data. Although, given that the primary columns of interest are "CountryCode", "Long.Name", and "Income.Group", the remaining attributes will be dropped from the data frame.

```
#Drop column index 4 thru the number of columns in the df
fedstats[4:ncol(fedstats)] <-NULL
```

- After dropping the respected columns, a look at the first five rows of the educational data frame is shown below.

```
#Display the first 5 rows of dataframe
knitr::kable(head(fedstats,5))
```

CountryCode	Long.Name	Income.Group
ABW	Aruba	High income: nonOECD
ADO	Principality of Andorra	High income: nonOECD
AFG	Islamic State of Afghanistan	Low income
AGO	People's Republic of Angola	Lower middle income
ALB	Republic of Albania	Upper middle income

Merge Data

- Now that both data frames, `Gross Domestic Product` and `Educational Data`, is cleaned and ready to merge, the unique ID column that will be used as the key to merge the data on is "CountryCode". The merged data set will be assigned to a new object called "merge.gdp.fedstats".

```
#horiztonal merge on "CountryCode"
merge.gdp.fedstats <- merge(gdp,fedstats,by="CountryCode")
```

- The initial column indexes are from 1:6, however for interpretability purposes, the position of these columns have been re-ordered accordingly:

```
#Change the order of the columns:
merge.gdp.fedstats <-merge.gdp.fedstats[c(1,3,5,4,6,2)]
knitr::kable(head(merge.gdp.fedstats,5))
```

CountryCode	Economy	Long.Name	GDP	Income.Group	Ranking
ABW	Aruba	Aruba	2584	High income: nonOECD	161
ADO	Andorra	Principality of Andorra	NA	High income: nonOECD	NA
AFG	Afghanistan	Islamic State of Afghanistan	20497	Low income	105
AGO	Angola	People's Republic of Angola	114147	Lower middle income	60
ALB	Albania	Republic of Albania	12648	Upper middle income	125

- Note: the answers to the analysis questions in the following section is based on removing each observation with NA in the "Ranking" column. Thus, the **merge.data.final** data frame will be the source of reference in the `analysis.r` file. First, let's replace any observation that is blank with NA:

```
#replace any blank observations with N/A
merge.gdp.fedstats[merge.gdp.fedstats == ""] <- NA
```

- Next, remove any observations in the "Ranking" column of the `merge.gdp.fedstats` data frame.

```
#Remove any Rows Wth NA's; keep all columns (i.e. ",")
merge.data.final<-merge.gdp.fedstats[!(is.na(merge.gdp.fedstats$Ranking)), ]
```

- **Final Merged Data Frame** - the first five rows will be displayed of the `merge.data.final` data frame to check if everything looks correct.

```
#Display the first 5 rows of dataframe
knitr::kable(head(merge.data.final,5))
```

	CountryCode	Economy	Long.Name	GDP	Income.Group	Ranking
1	ABW	Aruba	Aruba	2584	High income: nonOECD	161
3	AFG	Afghanistan	Islamic State of Afghanistan	20497	Low income	105
4	AGO	Angola	People's Republic of Angola	114147	Lower middle income	60
5	ALB	Albania	Republic of Albania	12648	Upper middle income	125
6	ARE	United Arab Emirates	United Arab Emirates	348595	High income: nonOECD	32

Statistical Analysis

Question #1: Merge the data based on the country shortcode. How many of the IDs match?

- After merging the data set, the number of matches can be determined either by visually examining the number of rows in the `merge.gdp.fedstats` data frame or by simply counting the number of matches between the fedstats and gdp data frames by the unique identifier "CountryCode" using the intersect function shown below. There is a total of 210 matches.

```
# Returns the total mataches between fedstats and gpd when merging on
CountryCode
print(paste0("Total Number of ID Matches: ",
             length(intersect(fedstats$CountryCode,gdp$CountryCode))))
```

```
## [1] "Total Number of ID Matches: 210"
```

- **Note:** The total number of matches after removing the NA's in the "Ranking" column of the `merge.data.final` data frame is 189. A total of 21 observations (or countries) were removed.

```
# Dimensions of the data frame
dim(merge.data.final)

## [1] 189   6
```

Question #2: Sort the data frame in ascending order by GDP (so United States is last). What is the 13th country in the resulting data frame?

- The 13th ranked country (in ascending order) of the GDP ranking is Grenada. From the output below, there are actually two countries that are tied for 13th. The other country is St. Kitts and Nevis.

```
#Sort merged data frame by ascending order & select rows 12-13 and all columns
sort.gdp <-
merge.data.final[order(merge.data.final$GDP,decreasing=FALSE,na.last =
TRUE),][12:13,]
sort.gdp[,c("CountryCode","Long.Name","Ranking","GDP")] #only display the
identified columns

##     CountryCode           Long.Name Ranking GDP
## 75          GRD             Grenada     178 767
## 102         KNA St. Kitts and Nevis     178 767
```

Question #3: What are the average GDP rankings for the "High income: OECD" and "High income: nonOECD" groups?

- For this question, the merged data set was grouped by "Income.Group" and then the average of the "Ranking" column was taken for the respected groups. All five income groups and the corresponding average rank of the aggregated data frame are shown in the code output below.

  Average Ranking per Income Group:
- High income:OECD = 32.97
- High income:nonOECD = 91.91

```
#Aggregrate data frame by Income.Group and take the mean rankings
merge.data.agg <- ddply(merge.data.final, .(Income.Group), summarize,
Ranking=mean(Ranking))
knitr::kable(merge.data.agg)
```

Income.Group	Ranking
High income: nonOECD	91.91304
High income: OECD	32.96667
Low income	133.72973

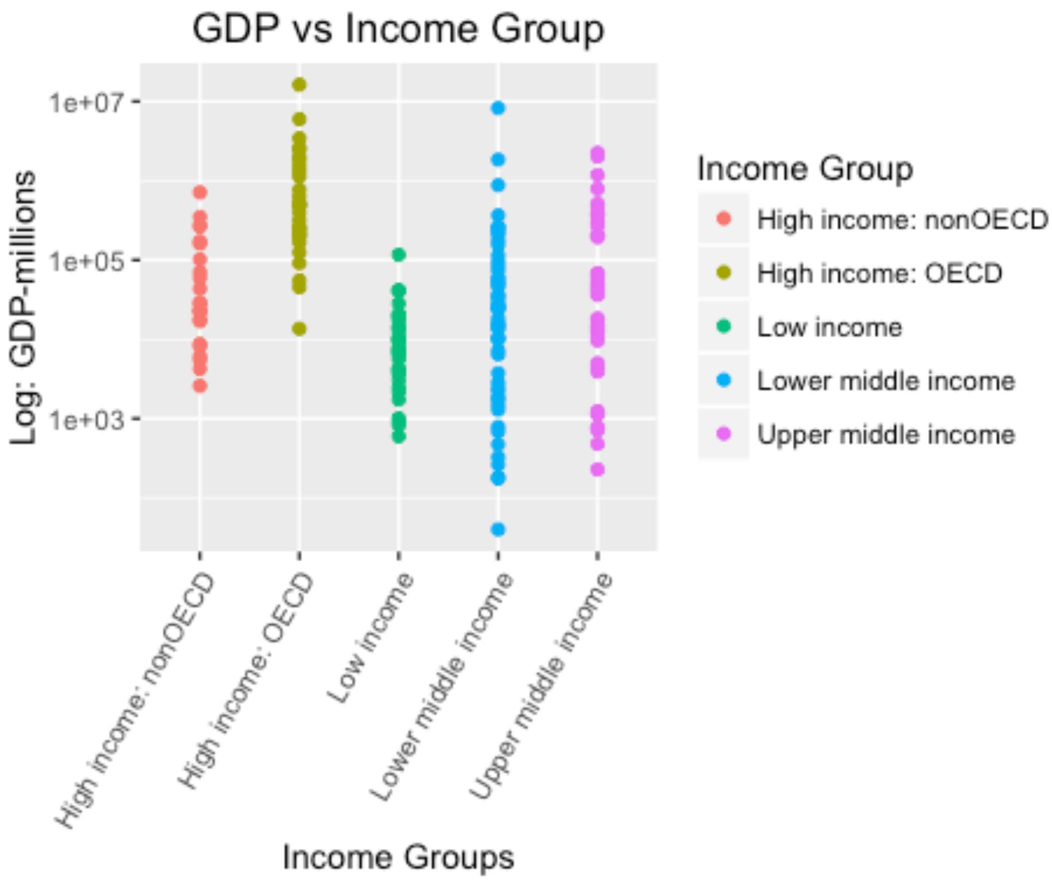
Lower middle income	107.70370
Upper middle income	92.13333

Question #4: Plot the GDP for all of the countries. Use ggplot2 to color your plot by Income Group.

- The Gross Domestic Product vs Income Group scatter plot is generated from the below code block. One observation drawn from the scatter plot (top) or the box plot (bottom) is that the within group standard deviation is the greatest for the lower middle income and upper middle income groups. Furthermore, the interquartile range for the low income group is the smallest.

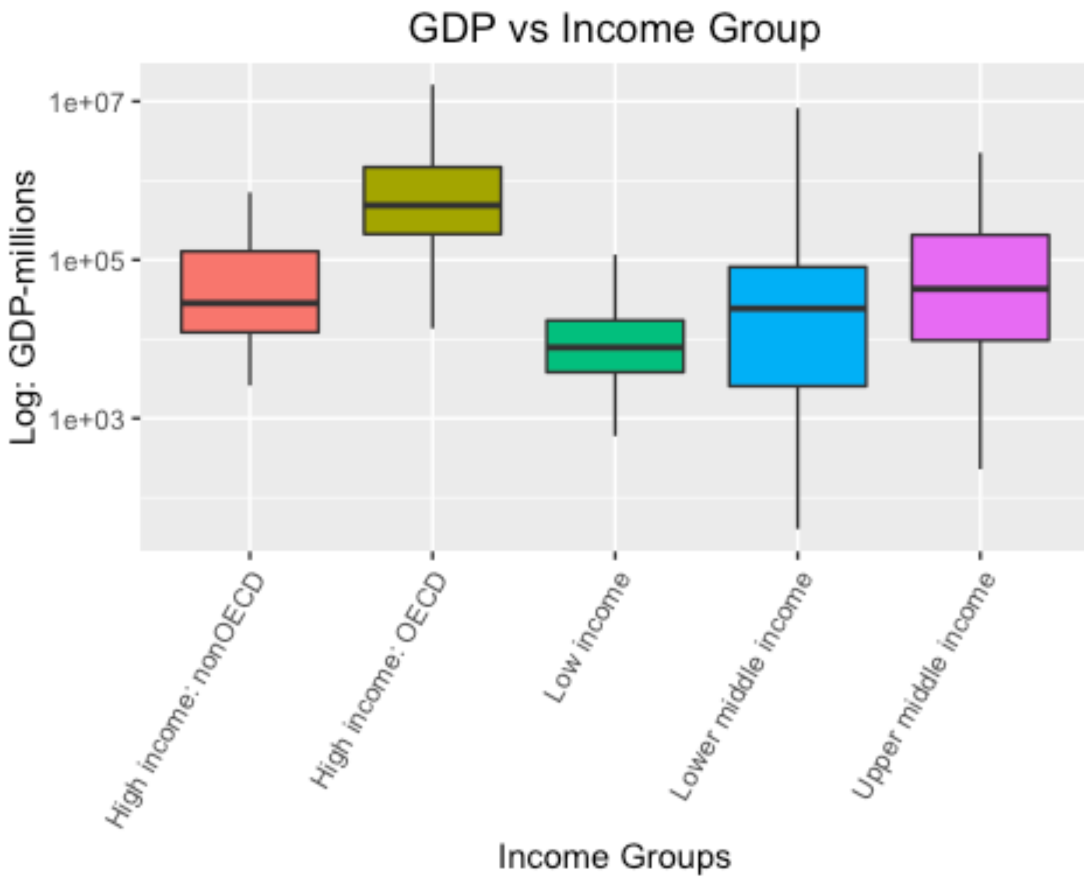
Scatter Plot

```
#Create a scatter plot using ggplot2 to plot Income.Group vs GDP (merged data frame)
p<-ggplot(merge.data.final)+
geom_point(aes(y=GDP,x=Income.Group,colour=Income.Group)) +scale_y_log10()
#change y axis to log scale
p+labs(title="GDP vs Income Group", # add title
       x="Income Groups",y="Log: GDP-millions",colour="Income Group") + #name labels
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) #adjust the x-axis labels (rotate)
```



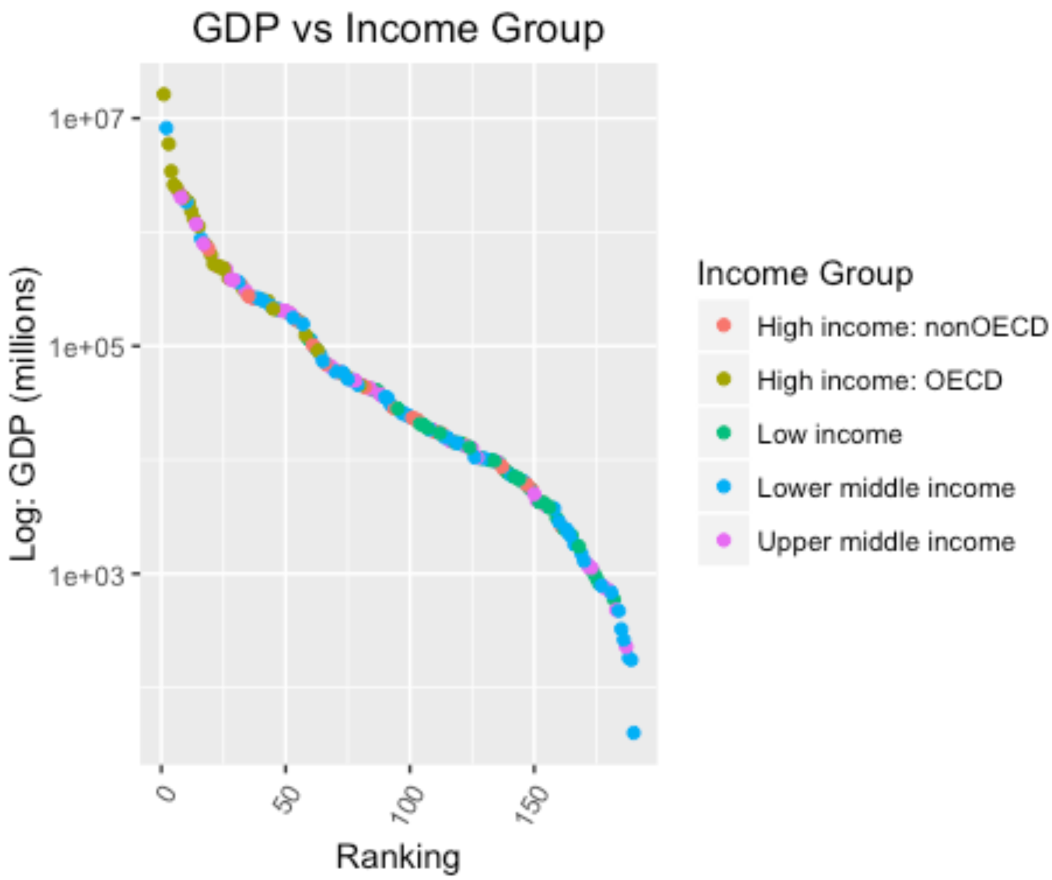
Box Plot

```
#Create a box plot using ggplot2 to plot Income.Group vs GDP (merged data
frame)
p<-ggplot(merge.data.final,aes(x=Income.Group,y=GDP))+
geom_boxplot(aes(fill=Income.Group)) +scale_y_log10() #change y axis to log
scale
p+labs(title="GDP vs Income Group",
       x="Income Groups",y="Log: GDP-millions",colour="Income Group") +
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) + #adjust the x-
axis labels (rotate)
  theme(legend.position = "none") #turn off legend
```



- One other way to analyze the Income Group is to plot the GDP vs the Ranking and group by the Income Groups. In the plot below, it can be observed that the lower middle income and low income has a larger distribution for GDP rankings greater than 100.

```
#Create a scatter plot using ggplot2 to plot Ranking vs GDP and group by
Income (merged data frame)
p<-ggplot(merge.data.final)+
geom_point(aes(y=GDP,x=Ranking,colour=Income.Group)) +scale_y_log10() #change
y axis to log scale
p+labs(title="GDP vs Income Group", # add title
       x="Ranking",y="Log: GDP (millions)",colour="Income Group") + #name
labels
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) #adjust the x-axis
labels (rotate)
```



Question #5: Cut the GDP ranking into 5 separate quantile groups. Make a table versus Income.Group. How many countries are Lower middle income but among the 38 nations with highest GDP?

- The cut function divides a numeric vector into different ranges. The total number of break points to apply on the `merge.data.final$Ranking` column is 5, which represents the different quantiles. The quantile ranges associated with each "Income.Group" is listed in the table below.

```
#convert Ranking column into numeric -- initially a factor
merge.data.final$Ranking <-
as.numeric(as.character(merge.data.final$Ranking))
#divide the numeric vector into 5 break points (i.e. quantiles)
merge.data.final$Group <- cut(merge.data.final$Ranking,breaks=5)
#take the quantiles and income.group from the merge data file and create a
table
quant.table<-table(merge.data.final$Income.Group, merge.data.final$Group)
#return table
knitr::kable(quant.table)
```

	(0.811,38.8]	(38.8,76.6]	(76.6,114]	(114,152]	(152,190]
	0	0	0	0	0

High income: nonOECD	4	5	8	4	2
High income: OECD	18	10	1	1	0
Low income	0	1	9	16	11
Lower middle income	5	13	12	8	16
Upper middle income	11	9	8	8	9

Conclusion

- In summary, this primary objective of this work is to take two unstructured data files, read the data from the a csv file into a data frame, tidy/clean up the data, merge the data frames, and perform some analysis on the final data set. As Data Scientist, it is very rare to receive perfectly formatted data and thus a large percentage of work will be cleaning up messy data sets before processing it as illustrated in this work. After preparing the data, statistical inference can then be made to the data. This Case Study walks through a practical example of cleaning a messy data set using R and the source code can simply be executed by running `analysis.r` which is sourced (i.e. linked) to the merge, tidy, and gather .r files in the Data directory.

Reference

- Adapted from the Case Study Report Help website of the University of New South Wales School of Engineering: Link