RLisbona_MSDS6306_Unit6Casestudy

Randy Lisbona

6/18/2016

Contents

Unit 6 Case study 1
Download GDP and Income group data sets 1
Clean raw source data 2
Merge the GDP and Income group datasets 4
Analyze the results 5
Conclusion and summary 11
Further analysis 12

Unit 6 Case study

Download, clean, merge, analyze Worldbank GDP and Income group data. The purpose of this study was compare Worldbank GDP Rank with Worldbank Incomegroup datasets by countrycode. Questions this study addresses: 1) How many CountryCodes are in both datasets 2) 13th country by ascending GDP Rank. Note -There is no single country with a "13" ranking, but two that share ranking 12.5. 3) What are average GDP Rankings of High Income: OECD and High Income nonOECD groups? Information on OECD (Organization for Economic Cooperation and Development) countries can be found at http://www.oecd.org/ and a list of OECD countries can be found here http://www.oecd.org/about/membersandpartners/list-oecd-member-countries.htm

Download GDP and Income group data sets

GDP source https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FGDP.csv Income group source https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FEDSTATS_Country.csv

In this code section the data is downloaded from the above URL's and saved to local .csv files At this stage the data is raw, no rows or columns have been modified or removed.

```
source("./Analysis/GatherWorldBankData.R", echo = TRUE,print.eval=TRUE)
```

```
##
## > casestudy.link.GDP <- c("https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FGDP.csv")
##
## > casestudy.link.IncomeGroupByCountry <- c("https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FE
##
## > source.url = casestudy.link.GDP
##
## > relativeSourcePath = c("./Analysis/Data")
##
## > GDP.filename = c("GDPbyCountry_raw.csv")
##
## > GDP.pathtofile <- paste(relativeSourcePath, GDP.filename,
## +     sep = "/")
##
## > paste("Downloading file ", GDP.pathtofile, sep = "")
```

```
## [1] "Downloading file ./Analysis/Data/GDPbyCountry_raw.csv"
##
## > paste("From URL ", source.url, sep = "")
## [1] "From URL https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FGDP.csv"
##
## > if (!file.exists(GDP.pathtofile)) download(source.url,
## +     GDP.pathtofile)
##
## > source.url = casestudy.link.IncomeGroupByCountry
##
## > IG.filename = c("IncomeGroupByCountry_raw.csv")
##
## > IG.pathtofile <- paste(relativeSourcePath, IG.filename,
## +     sep = "/")
##
## > paste("Downloading file ", IG.pathtofile, sep = "")
## [1] "Downloading file ./Analysis/Data/IncomeGroupByCountry_raw.csv"
##
## > paste("From URL ", source.url, sep = "")
## [1] "From URL https://d396qusza40orc.cloudfront.net/getdata%2Fdata%2FEDSTATS_Country.csv"
##
## > if (!file.exists(IG.pathtofile)) download(source.url,
## +     IG.pathtofile)
```

Clean raw source data

Cleaning steps include opening the raw source files, visually browsing the files to get a better understanding of quirks in the formatting, steps to extract and verify rows and columns to exclude. Finally outputing and saving cleaned datasets for further analysis in this project as well as future projects that might make use of the same data sources.

```
source("./Analysis/CleanWorldBankData.R", echo = TRUE,print.eval=TRUE)
```

```
##
## > GDP.raw <- read.csv(GDP.pathtofile, skip = 0, strip.white = TRUE,
## +     blank.lines.skip = TRUE, colClasses = "character")
##
## > paste(nrow(GDP.raw), "Records read from:", GDP.pathtofile,
## +     sep = " ")
## [1] "330 Records read from: ./Analysis/Data/GDPbyCountry_raw.csv"
##
## > IG.raw <- read.csv(IG.pathtofile, strip.white = TRUE,
## +     blank.lines.skip = TRUE, colClasses = "character")
##
## > paste(nrow(IG.raw), "Records read from:", IG.pathtofile,
## +     sep = " ")
## [1] "234 Records read from: ./Analysis/Data/IncomeGroupByCountry_raw.csv"
##
## > GDP_DataRows <- GDP.raw[5:194, ]
##
## > GDP_DataRowsValidCols <- GDP_DataRows[, c(1, 2, 4,
## +     5)]
##
## > names(GDP_DataRowsValidCols) <- c("CountryCode", "Ranking",
```

```
## +     "CountryName", "GDP_Millions_USD")
##
## > GDP_DataRowsValidCols$GDP_Millions_USD <- gsub(",",
## +     "", GDP_DataRowsValidCols$GDP_Millions_USD)
##
## > GDP_DataRowsValidCols$GDP_Millions_USD <- as.numeric(GDP_DataRowsValidCols$GDP_Millions_USD)
##
## > GDP_Clean <- GDP_DataRowsValidCols
##
## > IG.reduced <- IG.raw[c(1, 2, 3)]
##
## > names(IG.reduced) <- c("CountryCode", "CountryName",
## +     "IncomeGroup")
##
## > paste(nrow(IG.reduced), "Rows found in Income group dataset before removing blank IncomeGroups",
## +     sep = " ")
## [1] "234 Rows found in Income group dataset before removing blank IncomeGroups"
##
## > IG_Removed <- IG.reduced[IG.reduced$IncomeGroup ==
## +     "", ]
##
## > IG_Clean <- IG.reduced[!IG.reduced$IncomeGroup ==
## +     "", ]
##
## > paste(nrow(IG_Clean), "Rows in Income group dataset after removing blank IncomeGroups",
## +     sep = " ")
## [1] "210 Rows in Income group dataset after removing blank IncomeGroups"
##
## > RowsRemoved <- nrow(IG.reduced) - nrow(IG_Clean)
##
## > paste(RowsRemoved, "Rows removed", sep = " ")
## [1] "24 Rows removed"
##
## > IG_Clean <- arrange(IG_Clean, CountryCode)
##
## > relativeSourcePath = c("./Analysis/Data")
##
## > filename = c("GDPbyCountry_Clean.csv")
##
## > pathtofile <- paste(relativeSourcePath, filename,
## +     sep = "/")
##
## > paste("Write cleaned GDP data     :", pathtofile,
## +     sep = " ")
## [1] "Write cleaned GDP data     : ./Analysis/Data/GDPbyCountry_Clean.csv"
##
## > write.csv(GDP_Clean, file = pathtofile)
##
## > relativeSourcePath = c("./Analysis/Data")
##
## > filename = c("IncomeGroupByCountry_Clean.csv")
##
## > pathtofile <- paste(relativeSourcePath, filename,
## +     sep = "/")
```

```
##
## > paste("Write cleaned Income Group :", pathtofile,
## +     sep = " ")
## [1] "Write cleaned Income Group : ./Analysis/Data/IncomeGroupByCountry_Clean.csv"
##
## > write.csv(IG_Clean, file = pathtofile)
```

Merge the GDP and Income group datasets

Merging involved identifying a field common to both the GDP cleaned data and Income Group cleaned data files. Countrycode was common to both files and used as the matching field. inner and outer joins were run to understand missing fields in the combined dataset.

```
# Merge the data
source("./Analysis/MergeWorldBankData.R", echo = TRUE, print.eval=TRUE)
```

```
##
## > relativeSourcePath = c("./Analysis/Data")
##
## > filename = c("GDPbyCountry_Clean.csv")
##
## > pathtofile <- paste(relativeSourcePath, filename,
## +     sep = "/")
##
## > GDP = read.csv(pathtofile, stringsAsFactors = FALSE)
##
## > paste(nrow(GDP), "Records read from ", pathtofile,
## +     sep = " ")
## [1] "190 Records read from  ./Analysis/Data/GDPbyCountry_Clean.csv"
##
## > filename = c("IncomeGroupbyCountry_Clean.csv")
##
## > pathtofile <- paste(relativeSourcePath, filename,
## +     sep = "/")
##
## > IG = read.csv(pathtofile, stringsAsFactors = FALSE)
##
## > paste(nrow(IG), "Records read from ", pathtofile,
## +     sep = " ")
## [1] "210 Records read from  ./Analysis/Data/IncomeGroupbyCountry_Clean.csv"
##
## > GDP_sub.set <- GDP
##
## > IG_sub.set <- IG
##
## > fulljoin <- full_join(GDP_sub.set, IG_sub.set, by = "CountryCode")
##
## > innerjoin_GDP_IG <- inner_join(GDP_sub.set, IG_sub.set,
## +     by = "CountryCode")
##
## > unmatched_rows <- fulljoin %>% filter(is.na(CountryName.x) |
## +     is.na(IncomeGroup))
```

Analyze the results

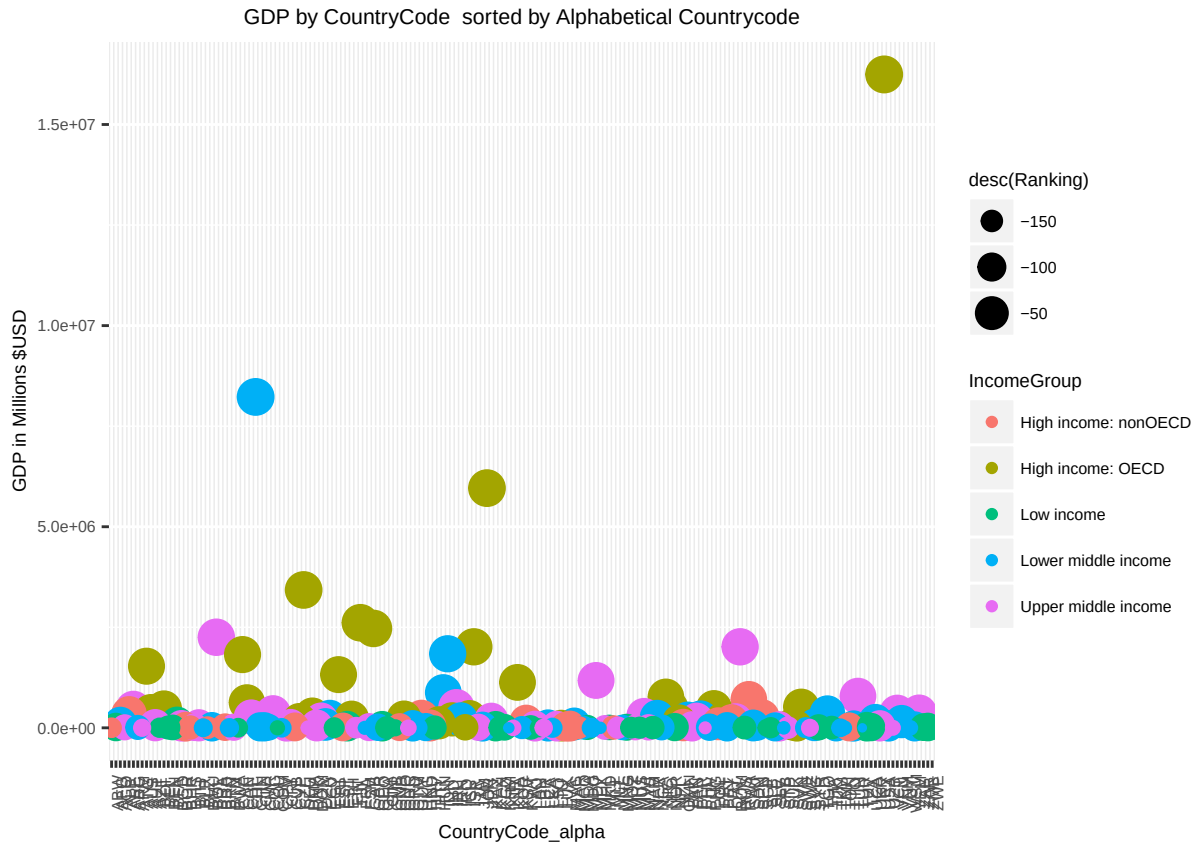
The analysis step consists of code necessary to answer the specific questions for the case study

```
source("./Analysis/AnalyzeWorldBankData.R", echo = TRUE, print.eval=TRUE)
```

```
##
## > paste("Distinct Incomegroups found in dataset", sep = "")
## [1] "Distinct Incomegroups found in dataset"
##
## > distinct(select(innerjoin_GDP_IG, IncomeGroup))
##            IncomeGroup
## 1    High income: OECD
## 2  Lower middle income
## 3  Upper middle income
## 4 High income: nonOECD
## 5           Low income
##
## > paste("Question 1, Match the data based on Countrycodes, how many match both datasets =",
## +     nrow(innerjoin_GDP_IG), sep = "")
## [1] "Question 1, Match the data based on Countrycodes, how many match both datasets =189"
##
## > paste("Count of Countries excluded due to incomplete data = ",
## +     nrow(fulljoin) - nrow(innerjoin_GDP_IG), sep = "")
## [1] "Count of Countries excluded due to incomplete data = 22"
##
## > paste("Table of Unmatched rows", sep = "")
## [1] "Table of Unmatched rows"
##
## > unmatched_rows
##    X.x CountryCode Ranking CountryName.x GDP_Millions_USD X.y                                Countryl
## 1  135         SSD     131   South Sudan             10220  NA
## 2   NA         ADO      NA          <NA>                NA   2                      Principality of An
## 3   NA         ASM      NA          <NA>                NA   9                               American
## 4   NA         CHI      NA          <NA>                NA  35                             Channel Is
## 5   NA         CYM      NA          <NA>                NA  46                              Cayman Is
## 6   NA         DJI      NA          <NA>                NA  50                         Republic of Dji
## 7   NA         FRO      NA          <NA>                NA  64                              Faeroe Is
## 8   NA         GRL      NA          <NA>                NA  76                                   Gree
## 9   NA         GUM      NA          <NA>                NA  78
## 10  NA         IMY      NA          <NA>                NA  86                                 Isle o
## 11  NA         LBY      NA          <NA>                NA 109    Socialist People's Libyan Arab Jamal
## 12  NA         LIE      NA          <NA>                NA 111                Principality of Liechten
## 13  NA         MMR      NA          <NA>                NA 128                             Union of My
## 14  NA         MNP      NA          <NA>                NA 131 Commonwealth of the Northern Mariana Is
## 15  NA         NCL      NA          <NA>                NA 138                                 New Cale
## 16  NA         PRK      NA          <NA>                NA 155        Democratic People's Republic of
## 17  NA         PYF      NA          <NA>                NA 158                             French Poly
## 18  NA         SMR      NA          <NA>                NA 170                      Republic of San M
## 19  NA         SOM      NA          <NA>                NA 171                   Somali Democratic Rep
## 20  NA         TCA      NA          <NA>                NA 181                     Turks and Caicos Is
## 21  NA         VIR      NA          <NA>                NA 201          Virgin Islands of the United S
## 22  NA         WBG      NA          <NA>                NA 204                           West Bank and
##
## > innerjoin_GDP_IG_rank <- mutate(innerjoin_GDP_IG,
```

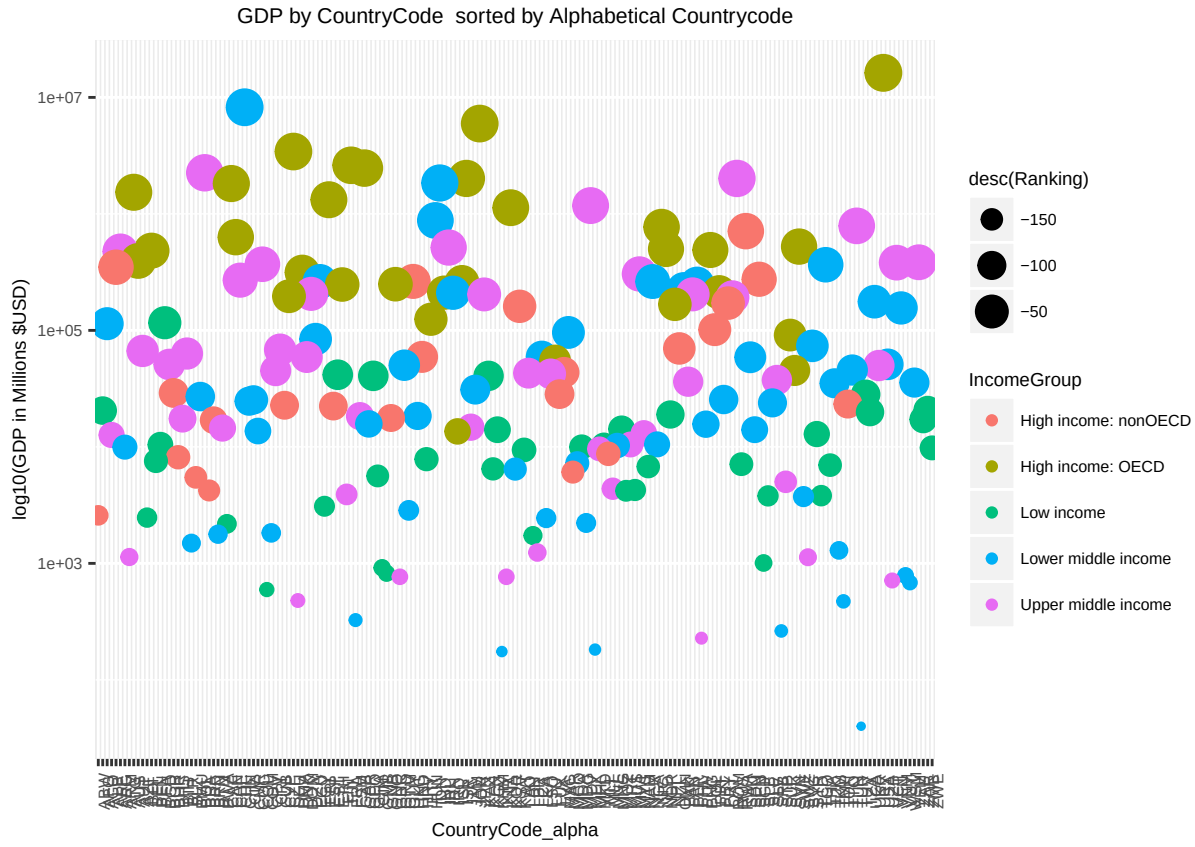
```
## +     GDPrank = rank(desc(Ranking)))
##
## > innerjoin_GDP_IG_rank.sort <- arrange(innerjoin_GDP_IG_rank,
## +     GDPrank)
##
## > newrank <- innerjoin_GDP_IG_rank.sort[c("CountryCode",
## +     "Ranking", "GDPrank", "CountryName.x", "GDP_Millions_USD",
## +     "IncomeGroup")]
##
## > paste("The 13th country by ascending GDP rank is not available",
## +     sep = "")
## [1] "The 13th country by ascending GDP rank is not available"
##
## > paste("Two countries share a rank of 12.5 , there is no 13th country",
## +     sep = "")
## [1] "Two countries share a rank of 12.5 , there is no 13th country"
##
## > newrank %>% select(Ranking, GDPrank, CountryCode,
## +     CountryName.x, GDP_Millions_USD) %>% arrange(GDPrank) %>%
## +     filter(GDPrank == 12.5) %> .... [TRUNCATED]
##   Ranking GDPrank CountryCode CountryName.x       GDP_Millions_USD
## 1 178     12.5    GRD         Grenada             767
## 2 178     12.5    KNA         St. Kitts and Nevis 767
##
## > paste("First 15 countries ranked by GDP", sep = "")
## [1] "First 15 countries ranked by GDP"
##
## > newrank %>% select(Ranking, GDPrank, CountryCode,
## +     CountryName.x, GDP_Millions_USD) %>% arrange(GDPrank) %>%
## +     filter(row_number() <= 15) .... [TRUNCATED]
##    Ranking GDPrank CountryCode CountryName.x                  GDP_Millions_USD
## 1  190      1.0    TUV         Tuvalu                          40
## 2  189      2.0    KIR         Kiribati                       175
## 3  188      3.0    MHL         Marshall Islands               182
## 4  187      4.0    PLW         Palau                          228
## 5  186      5.0    STP         São Tomé and Principe          263
## 6  185      6.0    FSM         Micronesia, Fed. Sts.          326
## 7  184      7.0    TON         Tonga                          472
## 8  183      8.0    DMA         Dominica                       480
## 9  182      9.0    COM         Comoros                        596
## 10 181     10.0    WSM         Samoa                          684
## 11 180     11.0    VCT         St. Vincent and the Grenadines 713
## 12 178     12.5    GRD         Grenada                        767
## 13 178     12.5    KNA         St. Kitts and Nevis            767
## 14 177     14.0    VUT         Vanuatu                        787
## 15 176     15.0    GNB         Guinea-Bissau                  822
##
## > PlotDataset <- arrange(innerjoin_GDP_IG_rank, Ranking)
##
## > HighIncomes <- filter(innerjoin_GDP_IG_rank, grepl(pattern = "High income",
## +     IncomeGroup))
##
## > by_IncomeGroup <- group_by(HighIncomes, IncomeGroup)
##
```

```
## > HighIncomeRanks <- summarise(by_IncomeGroup, meanrank = mean(Ranking),
## +     medianrank = median(Ranking))
##
## > print("Question3: average GDP rankings for High Income: OECD and High Income: nonOECD",
## +     sep = "")
## [1] "Question3: average GDP rankings for High Income: OECD and High Income: nonOECD"
##
## > print(HighIncomeRanks, digits = 2)
## Source: local data frame [2 x 3]
##
##            IncomeGroup meanrank medianrank
##                  (chr)    (dbl)      (dbl)
## 1 High income: nonOECD 91.91304       94.0
## 2    High income: OECD 32.96667       24.5
##
## > PlotDataset$CountryCode_alpha <- PlotDataset$CountryCode
##
## > PlotDataset$CountryCode <- factor(PlotDataset$CountryCode,
## +     levels = PlotDataset$CountryCode[order(PlotDataset$Ranking)])
##
## > p <- ggplot(data = PlotDataset, aes(x = CountryCode_alpha,
## +     y = GDP_Millions_USD, colour = IncomeGroup)) + geom_point(aes(size = desc(Ranking) .... [TRUNCA
##
## > print(p)
```

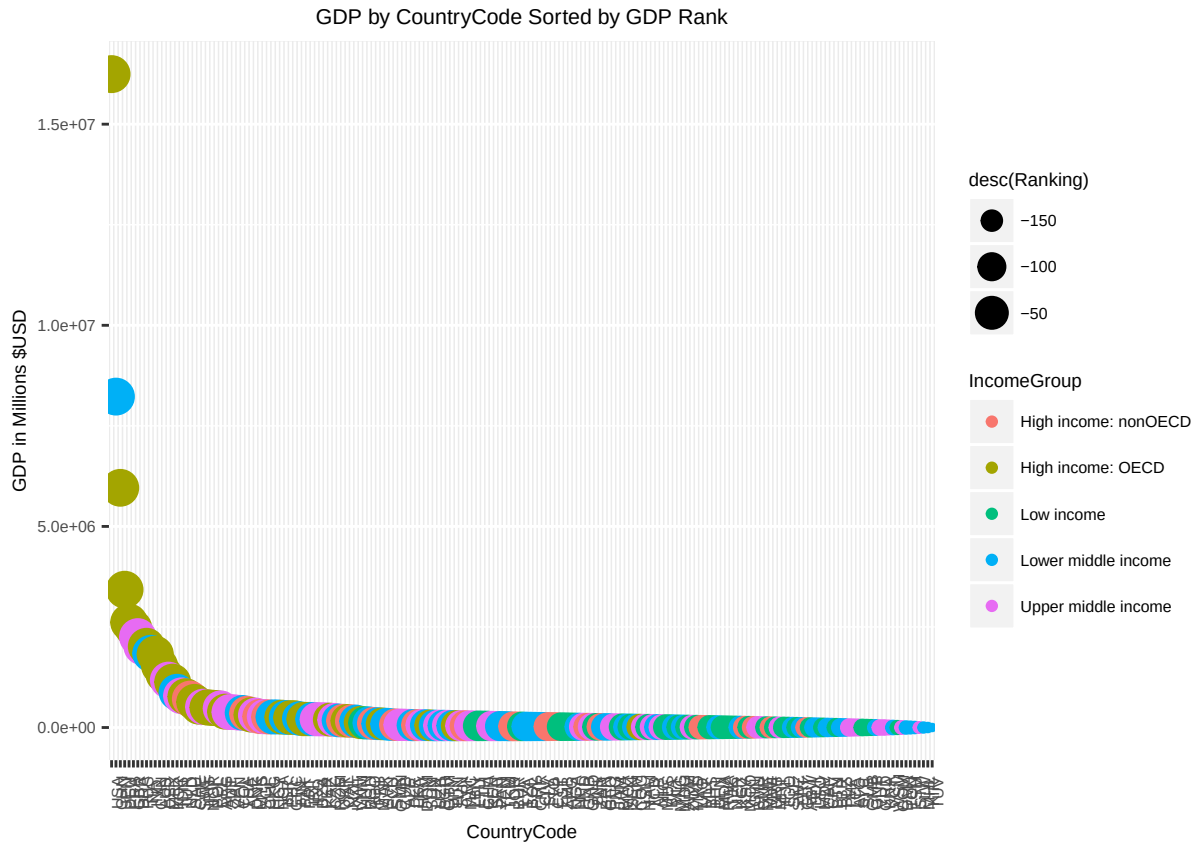



```
##
```

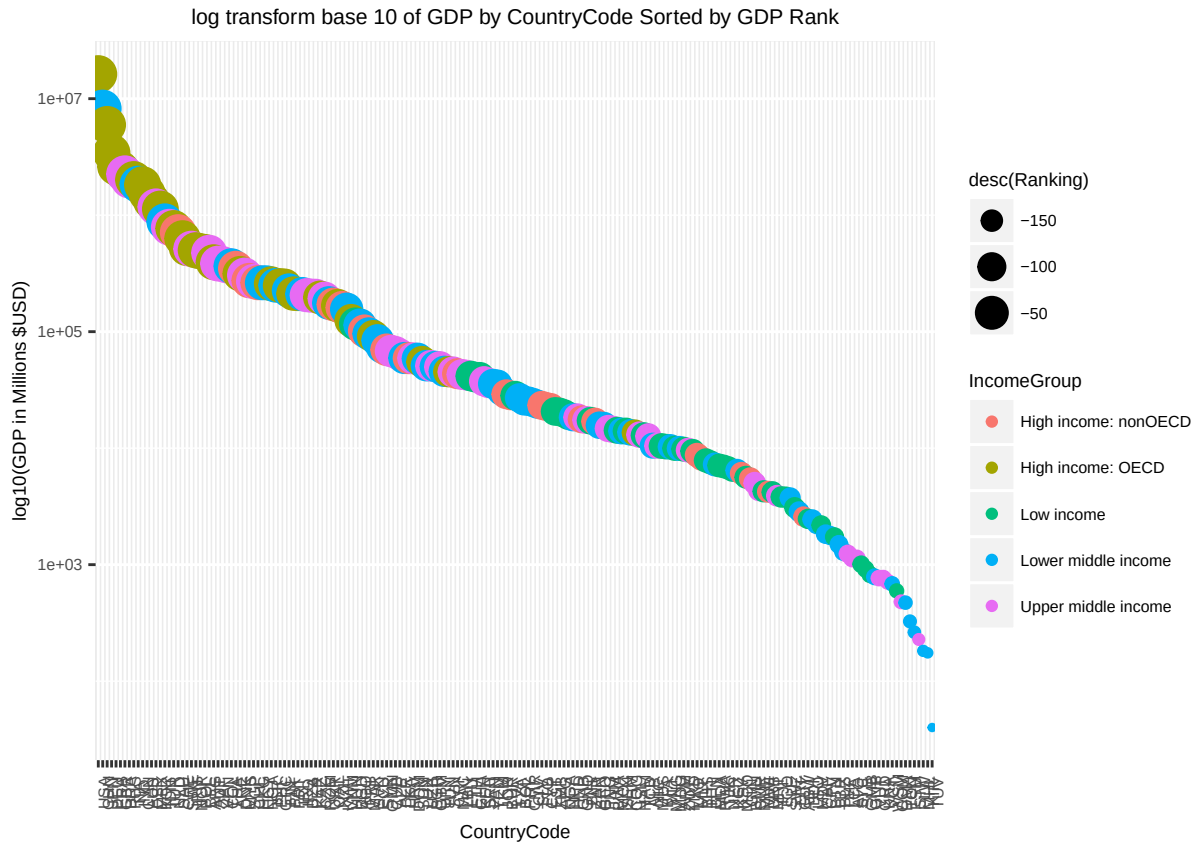
```
## > p <- ggplot(data = PlotDataset, aes(x = CountryCode_alpha,
## +     y = GDP_Millions_USD, colour = IncomeGroup)) + geom_point(aes(size = desc(Ranking) .... [TRUNCA
##
## > print(p)
```




```
##
## > p <- ggplot(data = PlotDataset, aes(x = CountryCode,
## +     y = GDP_Millions_USD, colour = IncomeGroup)) + geom_point(aes(size = desc(Ranking))) +
##  .... [TRUNCATED]
##
## > print(p)
```




```
##
## > p <- ggplot(data = PlotDataset, aes(x = CountryCode,
## +     y = GDP_Millions_USD, colour = IncomeGroup)) + geom_point(aes(size = desc(Ranking))) +
##  .... [TRUNCATED]
##
## > print(p)
```




```
##
## > PlotDataset$Quantile <- ntile(PlotDataset$Ranking,
## +     5)
##
## > summary_tbl2 <- ddply(PlotDataset, c("Quantile", "IncomeGroup"),
## +     summarise, N = length(X.x), mean_rank = mean(Ranking), sd_GDP_Millions = sd( .... [TRUNCATED]
##
## > summary_tbl2
##    Quantile          IncomeGroup  N mean_rank sd_GDP_Millions min_GDP    mean_GDP  max_GDP
## 1         1 High income: nonOECD  4  30.75000      211180.083  263259  399401.250   711050
## 2         1    High income: OECD 18  15.00000     3742064.198  314887 2369223.833 16244600
## 3         1  Lower middle income  5  19.40000     3363478.024  262832 2315130.800  8227103
## 4         1  Upper middle income 11  22.81818      705252.385  269869  812226.455  2252664
## 5         2 High income: nonOECD  5  61.60000       51452.754   59228  112617.000   171476
## 6         2    High income: OECD 10  51.70000       70239.523   55178  181262.700   258217
## 7         2           Low income  1  59.00000             NaN  116355  116355.000   116355
## 8         2  Lower middle income 13  57.69231       77489.648   51113  139805.923   262597
## 9         2  Upper middle income  9  61.22222       73946.529   50972  123770.667   205789
## 10        3 High income: nonOECD  8  99.75000        8481.791   16954   25515.875    43582
## 11        3    High income: OECD  1  80.00000             NaN   45279   45279.000    45279
## 12        3           Low income  9  98.55556       10491.696   17204   27608.667    41605
## 13        3  Lower middle income 12  95.08333       10303.698   15747   29858.750    50234
## 14        3  Upper middle income  8  90.37500       12077.362   17466   36237.250    49920
## 15        4 High income: nonOECD  5 144.80000        1874.914    4225    6529.000     8722
## 16        4    High income: OECD  1 122.00000             NaN   13579   13579.000    13579
## 17        4           Low income 16 135.87500        2941.448    4264    8983.250    14244
```

```
## 18        4  Lower middle income  8 128.87500        3268.195    6445   10975.625    15654
## 19        4  Upper middle income  8 130.50000        4031.741    4373   10557.875    14755
## 20        5 High income: nonOECD  1 161.00000             NaN    2584    2584.000     2584
## 21        5           Low income 11 166.18182        1332.547     596    2239.455     4199
## 22        5  Lower middle income 16 174.81250        1111.897      40    1286.688     3744
## 23        5  Upper middle income  9 175.22222        1083.887     228    1151.667     3908
##
## > PlotDataset <- mutate(PlotDataset, LMincomeTop38 = (Ranking <=
## +     38) * (IncomeGroup == "Lower middle income"))
##
## > LMincomeTop38.tbl <- PlotDataset[PlotDataset$LMincomeTop38 ==
## +     1, ]
##
## > paste(nrow(LMincomeTop38.tbl), "Countries are in the top 38 by GDP with Lower Middle Income",
## +     sep = " ")
## [1] "5 Countries are in the top 38 by GDP with Lower Middle Income"
##
## > paste("Countries with Lower middle Income groups in top 38 by GDP rank",
## +     sep = "")
## [1] "Countries with Lower middle Income groups in top 38 by GDP rank"
##
## > kable(LMincomeTop38.tbl[, c(2, 4, 7, 5, 11, 8, 3,
## +     12)], caption = "Countries with Lower middle Income groups in top 38 by GDP rank")
##
##
## Table: Countries with Lower middle Income groups in top 38 by GDP rank
##
##      CountryCode   CountryName.x      CountryName.y                GDP_Millions_USD   Quantile  Inc
## ---  ------------  -----------------  --------------------------  ------------------  ---------  ---
## 2    CHN           China              People's Republic of China             8227103          1  Low
## 10   IND           India              Republic of India                       1841710          1  Low
## 16   IDN           Indonesia          Republic of Indonesia                    878043          1  Low
## 31   THA           Thailand           Kingdom of Thailand                      365966          1  Low
## 38   EGY           Egypt, Arab Rep.   Arab Republic of Egypt                   262832          1  Low
```

Conclusion and summary

This study started with a list of 210 countries with GDP ranking data and 190 countries with Income group classifications, The two datasets were merged on "CountryCode" (USA, CAN, etc). 211 unique countries were found between the two datasets, 189 countries were found in both datasets, 22 were excluded due to missing data (NA's) necessary for the analysis in one or the other dataset.

There was interest in finding the 13th country, ranked by Ascending GDP rank (so USA is last) likely because there is no 13th country. Two countries, Grenada (GRD), and St. Kitts and Nevis (KNA), share rank 12 (denoted as 12.5). Average GDP rankings were calculated for High Income: OECD and High Income: nonOECD countries, OECD countries show a higher GDP rank (lower numbers are higher ranks)

Plots were produced to visually show the GDP rank of the income groups. Significant effort, bordering on ridiculous, was spent learning the idiosyncrasies of ggplot involved with getting the X axis country codes to sort by descending $GDP rather than alphabetically. This groups the OECD countries and higher GDP in the upper left quadrant of charts 3 and 4. Log transform plots were also included to bring the high GDP and low GDP countries closer together on the Y axis plots.

A table of GDP rankings (5 quantiles) was created, along with Frequency of income groups, SD/min/mean/max. There appears to be a significant GDP advantage to OECD member countries, further analysis would need

to be performed to verify this through acceptable statistical methods using tedious sofware such as R or SAS.

Five countries were found in the top 38 countries by $GDP with Lower middle income groups. Likely due to low wages in these countries.

Further analysis

Output graphics from R were disappointing, due to time constraints, it was not possible to fit the plots to the page width or rotate the page to landscape orientation so the country codes on the X axis did not land on top of each other. Graphics output was better in R studio than in the HTML output. Seemingly simple tasks are exceedingly difficult in R. Table formatting would need to be improved for publication quality graphics. This was a challenging project.