POC2_RMarkdown

```
library(readxl)
library(data.table)
library(dplyr)
library(ggplot2)


#1. Create data frame "dfCountries", load countries population file in it.
dfCountries <- read.csv("Countries Population.csv")


#2. Sort countries according to their population in ascending and descending order
# To sort countries according to population in ascending order
head(dfCountries[order(dfCountries$Total.Population.2017),])
```

```
##                 Country.Name Country.Code Total.Population.2017  X X.1
## 204                   Tuvalu          TUV                 11192 NA  NA
## 140                    Nauru          NRU                 13649 NA  NA
## 153                    Palau          PLW                 21729 NA  NA
## 28    British Virgin Islands          VGB                 31196 NA  NA
## 186 St. Martin (French part)          MAF                 32125 NA  NA
## 167               San Marino          SMR                 33400 NA  NA
```

```
#To sort countries according to population in descending order
head(dfCountries[order(-dfCountries$Total.Population.2017),])
```

```
##      Country.Name Country.Code Total.Population.2017  X X.1
## 42          China          CHN            1386395000 NA  NA
## 91          India          IND            1339180127 NA  NA
## 209 United States          USA             325719178 NA  NA
## 92      Indonesia          IDN             263991379 NA  NA
## 27         Brazil          BRA             209288278 NA  NA
## 152      Pakistan          PAK             197015955 NA  NA
```

```
#3. A vector of countries with population more than 10000000
countries <- dfCountries$Country.Name[dfCountries$Total.Population.2017 > 10000000]


#4. Create a dataframe "dfBigAndSmall" that has countries with population greater than 10M and less tha
dfBigAndSmall <- filter(dfCountries, Total.Population.2017 > 1000000 & Total.Population.2017 < 2000000)
head(dfBigAndSmall)
```

```
##        Country.Name Country.Code Total.Population.2017  X X.1
## 1           Bahrain          BHR               1492584 NA  NA
## 2            Cyprus          CYP               1179551 NA  NA
```

```
## 3 Equatorial Guinea          GNQ               1267689 NA  NA
## 4           Estonia          EST               1315480 NA  NA
## 5          Eswatini          SWZ               1367254 NA  NA
## 6     Guinea-Bissau          GNB               1861283 NA  NA
```

```
#5. Create levels of income group from dataset "Countries region mapping" levels: Low, Lower mid, Upper
# To load countries Region Mapping dataset
df2 <- read_excel('Countries Region Mapping.xlsx')
df2$IncomeGroup <- factor(df2$IncomeGroup, levels = c("Low income", "Lower middle income", "Upper middl
head(df2$IncomeGroup)
```

```
## [1] High      Low       Lower mid Upper mid High      High
## Levels: Low Lower mid Upper mid High
```

```
#6. Merge the 3 datasets attached into 1 dataframe : "dfCountryMaster"
# To read countries indicators csv file
df3 <- read.csv("Countries Indicators.csv")
# To merge Countries Population and Countries Region Mapping  data frames
df_merge1 <- merge(x=dfCountries, y=df2, by = "Country.Code", all=TRUE)
head(df_merge1)
```

```
##   Country.Code         Country.Name Total.Population.2017  X X.1
## 1          ABW                Aruba                105264 NA  NA
## 2          AFG          Afghanistan              35530081 NA  NA
## 3          AGO               Angola              29784193 NA  NA
## 4          ALB              Albania               2873457 NA  NA
## 5          AND              Andorra                 76965 NA  NA
## 6          ARE United Arab Emirates               9400145 NA  NA
##                       Region IncomeGroup
## 1  Latin America & Caribbean        High
## 2                South Asia         Low
## 3         Sub-Saharan Africa   Lower mid
## 4      Europe & Central Asia   Upper mid
## 5      Europe & Central Asia        High
## 6 Middle East & North Africa        High
```

```
#To check the class of data frame
class(df_merge1)
```

```
## [1] "data.frame"
```

```
#To merge df_merge1 and Countries Indicators data frames
dfCountryMaster <- merge(x=df_merge1, y=df3, by="Country.Code", all=TRUE)
head(dfCountryMaster)
```

```
##   Country.Code Country.Name Total.Population.2017 X.x X.1.x
## 1          ABW        Aruba                105264  NA    NA
## 2          AFG  Afghanistan              35530081  NA    NA
## 3          AGO       Angola              29784193  NA    NA
## 4          ALB      Albania               2873457  NA    NA
## 5          AND      Andorra                 76965  NA    NA
```

```
## 6          ARB         <NA>                    NA  NA    NA
##                       Region IncomeGroup GDP.per.capita.2017
## 1 Latin America & Caribbean        High        25,655.10202
## 2                South Asia         Low         550.0684588
## 3        Sub-Saharan Africa   Lower mid        4,100.289786
## 4     Europe & Central Asia   Upper mid        4,537.579056
## 5     Europe & Central Asia        High        39,146.54884
## 6                      <NA>        <NA>        6,239.713933
##   Under.5.Mortality.Rate.2017 X.y X.1.y
## 1                          NA  NA    NA
## 2                     67.9000  NA    NA
## 3                     81.1000  NA    NA
## 4                      8.8000  NA    NA
## 5                      3.3000  NA    NA
## 6                     36.6612  NA    NA
```

```
colnames(dfCountryMaster)
```

```
##  [1] "Country.Code"                "Country.Name"
##  [3] "Total.Population.2017"       "X.x"
##  [5] "X.1.x"                       "Region"
##  [7] "IncomeGroup"                 "GDP.per.capita.2017"
##  [9] "Under.5.Mortality.Rate.2017" "X.y"
## [11] "X.1.y"
```

```
#To check dimensions of 3 datasets
dim(dfCountries)
```

```
## [1] 219   5
```

```
dim(df2)
```

```
## [1] 216   3
```

```
dim(df3)
```

```
## [1] 241   5
```

```
# To check dimensions of merged dataset
dim(dfCountryMaster)
```

```
## [1] 260  11
```

```
# To check coloumns of merged dataset
names(dfCountryMaster)
```

```
##  [1] "Country.Code"                "Country.Name"
##  [3] "Total.Population.2017"       "X.x"
##  [5] "X.1.x"                       "Region"
##  [7] "IncomeGroup"                 "GDP.per.capita.2017"
##  [9] "Under.5.Mortality.Rate.2017" "X.y"
## [11] "X.1.y"
```

```r
#7. Summarize dfCountryMaster countries by region.
dfCountryMaster %>% group_by(Region) %>% summarise( num_countries = n())
```

```
## # A tibble: 8 x 2
##   Region                       num_countries
##   <chr>                                <int>
## 1 East Asia & Pacific                     36
## 2 Europe & Central Asia                   58
## 3 Latin America & Caribbean               42
## 4 Middle East & North Africa              21
## 5 North America                            3
## 6 South Asia                               8
## 7 Sub-Saharan Africa                      48
## 8 <NA>                                    44
```

```r
#8. Summarize dfCountryMaster countries by region and income group.
dfCountryMaster %>% group_by(Region, IncomeGroup) %>% summarise(num_countries = n())
```

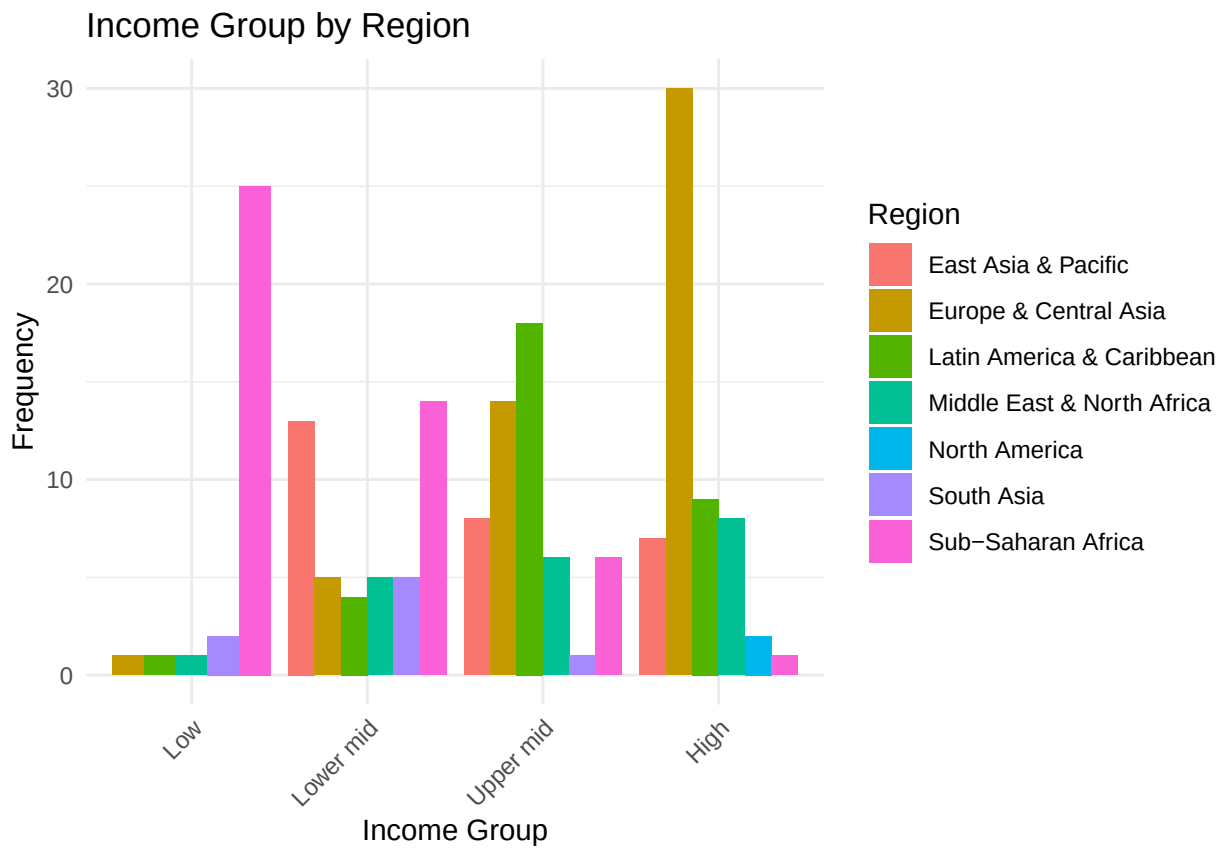
```
## # A tibble: 25 x 3
## # Groups:   Region [8]
##    Region                    IncomeGroup num_countries
##    <chr>                     <fct>               <int>
##  1 East Asia & Pacific       Low                     1
##  2 East Asia & Pacific       Lower mid              13
##  3 East Asia & Pacific       Upper mid               9
##  4 East Asia & Pacific       High                   13
##  5 Europe & Central Asia     Low                     1
##  6 Europe & Central Asia     Lower mid               6
##  7 Europe & Central Asia     Upper mid              14
##  8 Europe & Central Asia     High                   37
##  9 Latin America & Caribbean Low                     1
## 10 Latin America & Caribbean Lower mid               4
## # i 15 more rows
```

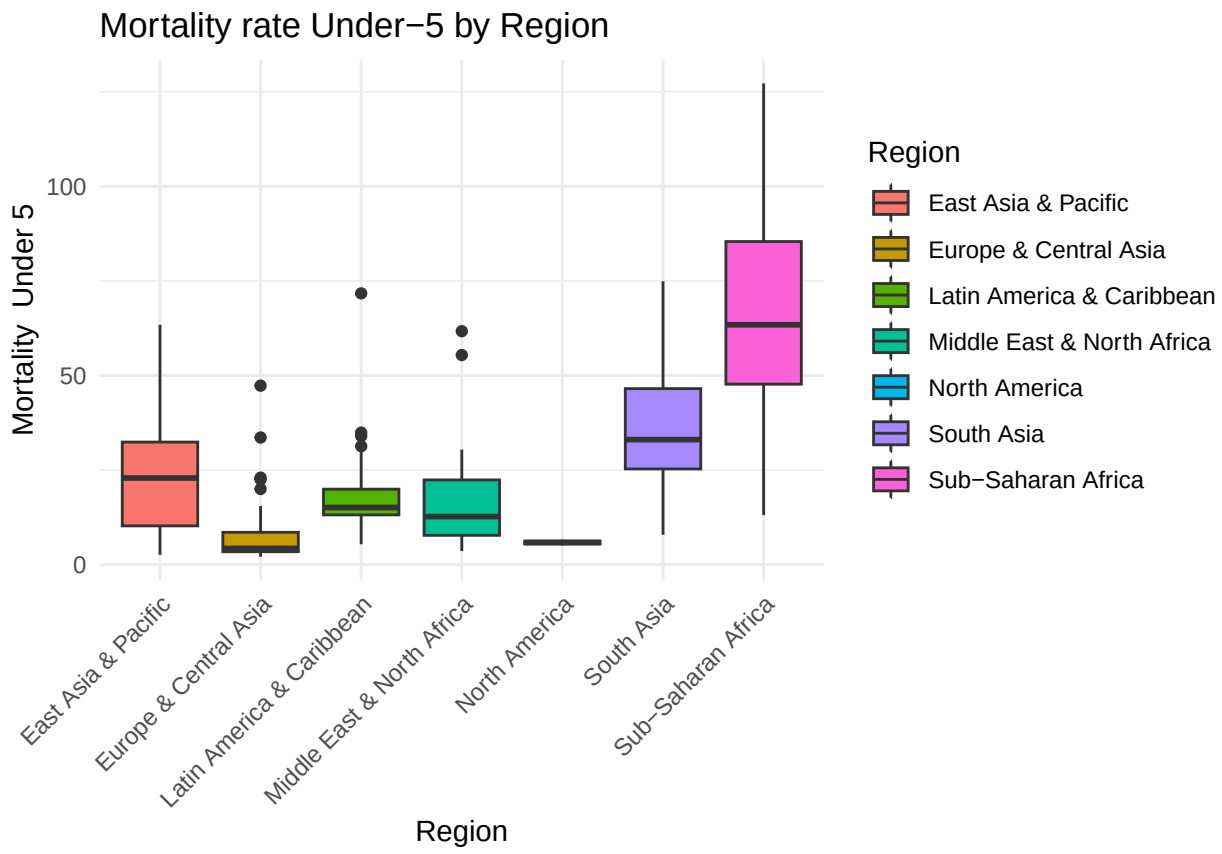
```r
#9. Summarize dfCountryMaster countries by region. Result to have the following columns in it.
#a. Number of countries.
#b. Total population in millions.
#c. Average of GDP per capital
#d. Countries with low income.
#e. Median GDP per capital
#f. minimum and maximum mortality rate under 5.

Summary_countries_by_region  <- dfCountryMaster %>% group_by(Region) %>%
                                summarise( num_countries = n(),
                                           total_population_million = sum(Total.Population.2017, na.rm =
                                           avg_GDP_per_capital = mean(as.numeric(GDP.per.capita.2017), r
                                           low_income_countries = sum(IncomeGroup == "Low income", na.rr
                                           median_gdp_per_capita = median(as.numeric(GDP.per.capita.201
                                           min_mortality_under_5 = min(Under.5.Mortality.Rate.2017, na.r
                                           max_mortality_under_5 = max(Under.5.Mortality.Rate.2017, na.r
                                )
```

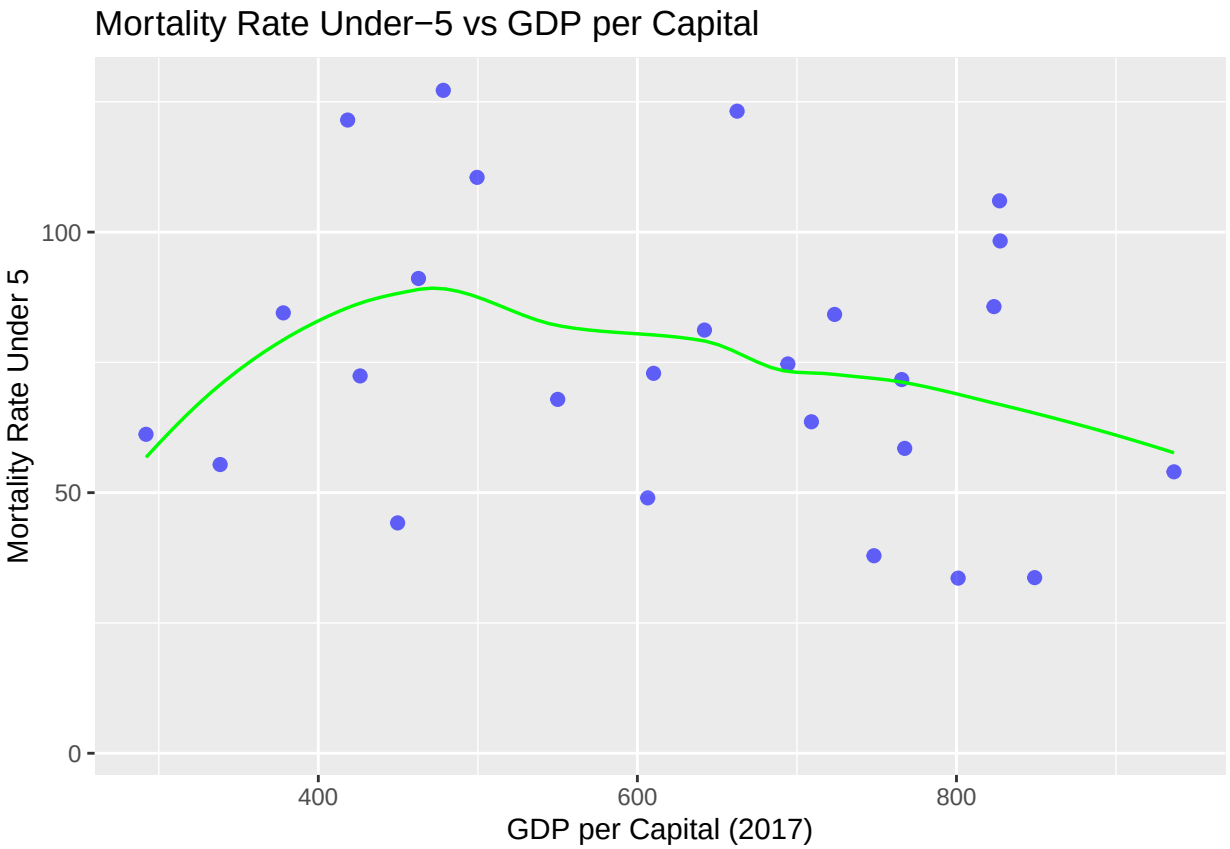
```r
#10. Write the above result in csv.
write.csv(Summary_countries_by_region, file = "Country Summary by Region.csv", row.names = FALSE)
```

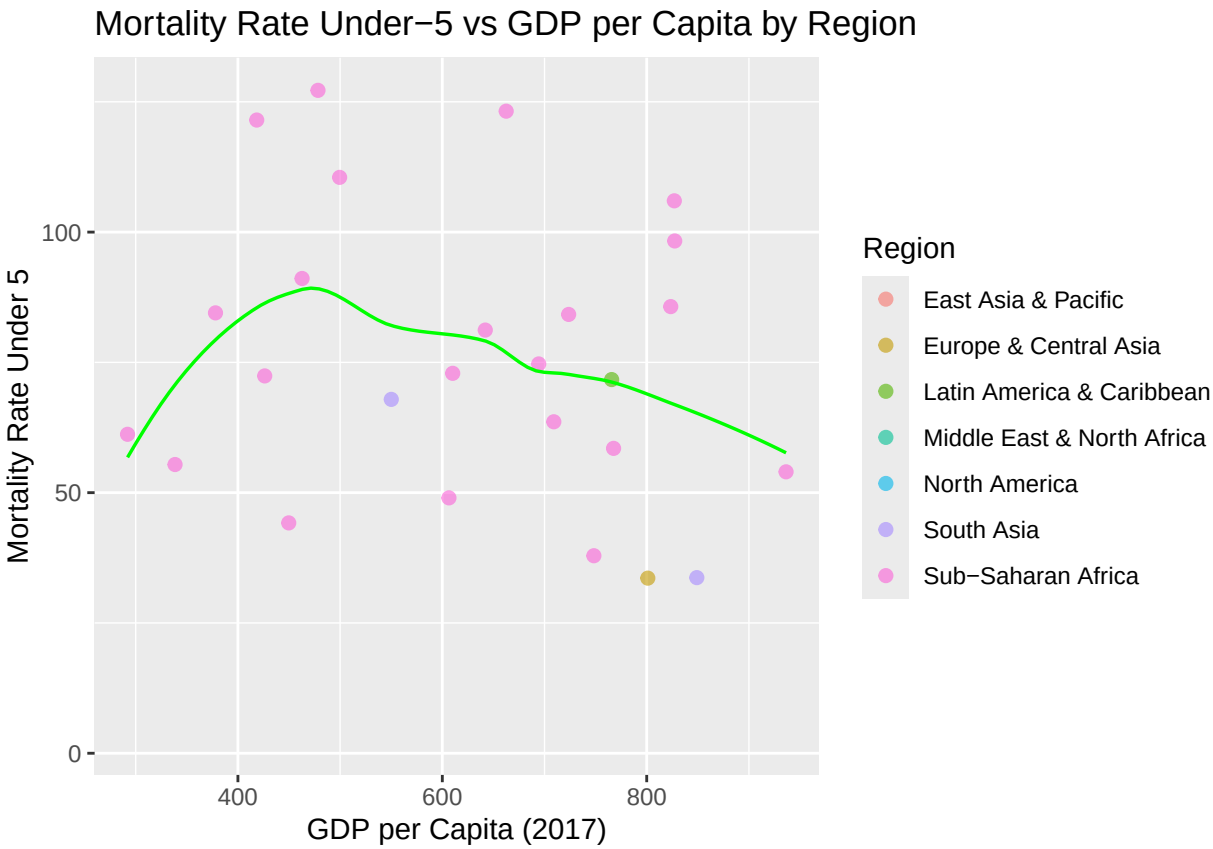
Including Plots

You can also embed plots, for example:




Mortality rate Under−5 by Region
Mortality Under 5
100
50
0
East Asia & Pacific
Europe & Central Asia
Latin America & Caribbean
Middle East & North Africa
North America
South Asia
Sub−Saharan Africa
Region
Region
East Asia & Pacific
Europe & Central Asia
Latin America & Caribbean
Middle East & North Africa
North America
South Asia
Sub−Saharan Africa


Mortality Rate Under−5 vs GDP per Capital
Mortality Rate Under 5
100
50
0
400
600
800
GDP per Capital (2017)


Mortality Rate Under−5 vs GDP per Capita by Region
Mortality Rate Under 5
100
50
0
400
600
800
GDP per Capita (2017)
Region
East Asia & Pacific
Europe & Central Asia
Latin America & Caribbean
Middle East & North Africa
North America
South Asia
Sub−Saharan Africa