Programming and Data Analysis for Business exam

S129049

2/27/2022

Installing necessary Libraries

```
library(tidyverse)
library(ggstatsplot)
library(ggalt)
library(gridExtra)
library(broom)
library(janitor)
library(ggthemes)
library(RColorBrewer)
library(ggrepel)
library(gridExtra)
library(extrafont)
library(dendextend)
library(factoextra)
library(magrittr)
library(DT)
library(knitr)
library(rmarkdown)
library(reshape2)
```

Importing data sets

```
library(readr)
master <-
  read_csv("master.csv", col_types = cols(`HDI for year` = col_double()))
glimpse(master)
```

```
## Rows: 27,820
## Columns: 12
## $ country              <chr> "Albania", "Albania", "Albania", "Albania", "Alba…
## $ year                 <dbl> 1987, 1987, 1987, 1987, 1987, 1987, 1987, 1987, 1…
## $ sex                  <chr> "male", "male", "female", "male", "male", "female…
## $ age                  <chr> "15-24 years", "35-54 years", "15-24 years", "75+…
## $ suicides_no          <dbl> 21, 16, 14, 1, 9, 1, 6, 4, 1, 0, 0, 0, 2, 17, 1, …
## $ population           <dbl> 312900, 308000, 289700, 21800, 274300, 35600, 278…
## $ `suicides/100k pop`  <dbl> 6.71, 5.19, 4.83, 4.59, 3.28, 2.81, 2.15, 1.56, 0…
## $ `country-year`       <chr> "Albania1987", "Albania1987", "Albania1987", "Alb…
## $ `HDI for year`       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…
## $ `gdp_for_year ($)`   <dbl> 2156624900, 2156624900, 2156624900, 2156624900, 2…
## $ `gdp_per_capita ($)` <dbl> 796, 796, 796, 796, 796, 796, 796, 796, 796, 796,…
## $ generation           <chr> "Generation X", "Silent", "Generation X", "G.I. G…
```

```
str(master)
```

```
## spec_tbl_df [27,820 × 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ country           : chr [1:27820] "Albania" "Albania" "Albania" "Albania" ...
##  $ year              : num [1:27820] 1987 1987 1987 1987 1987 ...
##  $ sex               : chr [1:27820] "male" "male" "female" "male" ...
##  $ age               : chr [1:27820] "15-24 years" "35-54 years" "15-24 years" "75+ years" ...
##  $ suicides_no       : num [1:27820] 21 16 14 1 9 1 6 4 1 0 ...
##  $ population        : num [1:27820] 312900 308000 289700 21800 274300 ...
##  $ suicides/100k pop : num [1:27820] 6.71 5.19 4.83 4.59 3.28 2.81 2.15 1.56 0.73 0 ...
##  $ country-year      : chr [1:27820] "Albania1987" "Albania1987" "Albania1987" "Albania1987" ...
##  $ HDI for year      : num [1:27820] NA NA NA NA NA NA NA NA NA NA ...
##  $ gdp_for_year ($)  : num [1:27820] 2.16e+09 2.16e+09 2.16e+09 2.16e+09 2.16e+09 ...
##  $ gdp_per_capita ($): num [1:27820] 796 796 796 796 796 796 796 796 796 796 ...
##  $ generation        : chr [1:27820] "Generation X" "Silent" "Generation X" "G.I. Generation" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   country = col_character(),
##   ..   year = col_double(),
##   ..   sex = col_character(),
##   ..   age = col_character(),
##   ..   suicides_no = col_double(),
##   ..   population = col_double(),
##   ..   `suicides/100k pop` = col_double(),
##   ..   `country-year` = col_character(),
##   ..   `HDI for year` = col_double(),
##   ..   `gdp_for_year ($)` = col_number(),
##   ..   `gdp_per_capita ($)` = col_double(),
##   ..   generation = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

There are two continent datasets that are going to be imported in order to match as many countries as possible with the region and subregion

```
library(readr)
continents <- read_csv("all.csv")
library(readr)
continents2 <- read_csv("continents2.csv")
str(continents)
```

```
## spec_tbl_df [249 × 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ name                    : chr [1:249] "Afghanistan" "Åland Islands" "Albania" "Algeria" ...
##  $ alpha-2                 : chr [1:249] "AF" "AX" "AL" "DZ" ...
##  $ alpha-3                 : chr [1:249] "AFG" "ALA" "ALB" "DZA" ...
##  $ country-code            : chr [1:249] "004" "248" "008" "012" ...
##  $ iso_3166-2              : chr [1:249] "ISO 3166-2:AF" "ISO 3166-2:AX" "ISO 3166-2:AL" "ISO 3166-2:DZ" ...
##  $ region                  : chr [1:249] "Asia" "Europe" "Europe" "Africa" ...
##  $ sub-region              : chr [1:249] "Southern Asia" "Northern Europe" "Southern Europe" "Northern Africa"
...
##  $ intermediate-region     : chr [1:249] NA NA NA NA ...
##  $ region-code             : chr [1:249] "142" "150" "150" "002" ...
##  $ sub-region-code         : chr [1:249] "034" "154" "039" "015" ...
##  $ intermediate-region-code: chr [1:249] NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   name = col_character(),
##   ..   `alpha-2` = col_character(),
##   ..   `alpha-3` = col_character(),
##   ..   `country-code` = col_character(),
##   ..   `iso_3166-2` = col_character(),
##   ..   region = col_character(),
##   ..   `sub-region` = col_character(),
##   ..   `intermediate-region` = col_character(),
##   ..   `region-code` = col_character(),
##   ..   `sub-region-code` = col_character(),
##   ..   `intermediate-region-code` = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(continents2)
```

```
## spec_tbl_df [249 × 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ name                    : chr [1:249] "Afghanistan" "Åland Islands" "Albania" "Algeria" ...
##  $ alpha-2                 : chr [1:249] "AF" "AX" "AL" "DZ" ...
##  $ alpha-3                 : chr [1:249] "AFG" "ALA" "ALB" "DZA" ...
##  $ country-code            : num [1:249] 4 248 8 12 16 20 24 660 10 28 ...
##  $ iso_3166-2              : chr [1:249] "ISO 3166-2:AF" "ISO 3166-2:AX" "ISO 3166-2:AL" "ISO 3166-2:DZ" ...
##  $ region                  : chr [1:249] "Asia" "Europe" "Europe" "Africa" ...
##  $ sub-region              : chr [1:249] "Southern Asia" "Northern Europe" "Southern Europe" "Northern Africa"
...
##  $ intermediate-region     : chr [1:249] NA NA NA NA ...
##  $ region-code             : num [1:249] 142 150 150 2 9 150 2 19 NA 19 ...
##  $ sub-region-code         : num [1:249] 34 154 39 15 61 39 202 419 NA 419 ...
##  $ intermediate-region-code: num [1:249] NA NA NA NA NA NA 17 29 NA 29 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   name = col_character(),
##   ..   `alpha-2` = col_character(),
##   ..   `alpha-3` = col_character(),
##   ..   `country-code` = col_double(),
##   ..   `iso_3166-2` = col_character(),
##   ..   region = col_character(),
##   ..   `sub-region` = col_character(),
##   ..   `intermediate-region` = col_character(),
##   ..   `region-code` = col_double(),
##   ..   `sub-region-code` = col_double(),
##   ..   `intermediate-region-code` = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Combining datasets

I will select solely the columns I would like to add to the suicide rate dataset

```
continents <- data.frame(
  country_name = continents$name,
  region = continents$region,
  subregion = continents$`sub-region`
)

continents2 <- data.frame(
  country_name = continents2$name,
  region = continents2$region,
  subregion = continents2$`sub-region`
)
```

I will create a function that will allow me to merge the two dataset together based on the country names

```
mymerge <-
  function (x, y) {
    masterdata <- merge (
      x,
      y,
      by.x = c("country"),
      by.y = c("country_name"),
      all.x = TRUE
    )
    return(masterdata)
  }
```

I will combine the two continents dataset

```
continents <- rbind(continents, continents2)
```

I will now create the new dataset which contains the suicide rate information and the regions and subregions from the continents dataset

```
rate <-
  Reduce(mymerge,
         list(master,
              continents))
```

I will rename the columns necessary

```
rate = rename(
  rate,
  c(
    `suicides_100k` = `suicides/100k pop`,
    `HDI` = `HDI for year`,
    `GDP_year` = `gdp_for_year ($)`,
    `GDP_per_capita` = `gdp_per_capita ($)`,
    `country_year` = `country-year`
  )
)
```

Cleaning data

I will look for missing values and clean the duplicated rows

```
rate <- distinct(rate)
str(rate)
```

```
## 'data.frame':    27820 obs. of  14 variables:
##  $ country       : chr  "Albania" "Albania" "Albania" "Albania" ...
##  $ year          : num  1995 1995 1995 1999 1999 ...
##  $ sex           : chr  "male" "male" "male" "male" ...
##  $ age           : chr  "75+ years" "35-54 years" "15-24 years" "35-54 years" ...
##  $ suicides_no   : num  1 14 11 31 19 14 19 13 6 5 ...
##  $ population    : num  25100 375900 241200 391100 242300 ...
##  $ suicides_100k : num  3.98 3.72 4.56 7.93 7.84 7.56 6.4 4.7 3.18 1.34 ...
##  $ country_year  : chr  "Albania1995" "Albania1995" "Albania1995" "Albania1999" ...
##  $ HDI           : num  0.619 0.619 0.619 NA NA NA NA NA NA NA ...
##  $ GDP_year      : num  2.42e+09 2.42e+09 2.42e+09 3.41e+09 3.41e+09 ...
##  $ GDP_per_capita: num  835 835 835 1127 1127 ...
##  $ generation    : chr  "G.I. Generation" "Boomers" "Generation X" "Boomers" ...
##  $ region        : chr  "Europe" "Europe" "Europe" "Europe" ...
##  $ subregion     : chr  "Southern Europe" "Southern Europe" "Southern Europe" "Southern Europe" ...
```

```
glimpse(rate)
```

```
## Rows: 27,820
## Columns: 14
## $ country        <chr> "Albania", "Albania", "Albania", "Albania", "Albania", …
## $ year           <dbl> 1995, 1995, 1995, 1999, 1999, 1999, 1999, 1999, 1999, 1…
## $ sex            <chr> "male", "male", "male", "male", "male", "male", "female…
## $ age            <chr> "75+ years", "35-54 years", "15-24 years", "35-54 years…
## $ suicides_no    <dbl> 1, 14, 11, 31, 19, 14, 19, 13, 6, 5, 13, 10, 12, 9, 7, …
## $ population     <dbl> 25100, 375900, 241200, 391100, 242300, 185200, 296800, …
## $ suicides_100k  <dbl> 3.98, 3.72, 4.56, 7.93, 7.84, 7.56, 6.40, 4.70, 3.18, 1…
## $ country_year   <chr> "Albania1995", "Albania1995", "Albania1995", "Albania19…
## $ HDI            <dbl> 0.619, 0.619, 0.619, NA, NA, NA, NA, NA, NA, NA, 0.619,…
## $ GDP_year       <dbl> 2424499009, 2424499009, 2424499009, 3414760915, 3414760…
## $ GDP_per_capita <dbl> 835, 835, 835, 1127, 1127, 1127, 1127, 1127, 1127, 1127…
## $ generation     <chr> "G.I. Generation", "Boomers", "Generation X", "Boomers"…
## $ region         <chr> "Europe", "Europe", "Europe", "Europe", "Europe", "Euro…
## $ subregion      <chr> "Southern Europe", "Southern Europe", "Southern Europe"…
```

```
sum(is.na(rate))
```

```
## [1] 20824
```

```
glimpse(is.na(rate))
```

```
##  logi [1:27820, 1:14] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:14] "country" "year" "sex" "age" ...
```

```
sum(is.na(rate$region))
```

```
## [1] 684
```

```
sum(is.na(rate$subregion))
```

```
## [1] 684
```

```
sum(is.na(rate$HDI))
```

```
## [1] 19456
```

I will check if each country has 12 rows

```
rate %>% group_by(country_year) %>% count() %>% filter(n != 12)
```

country_year <chr>	**n** <int>
Armenia2016	10
Austria2016	10
Croatia2016	10
Cyprus2016	10
Czech Republic2016	10
Grenada2016	10
Hungary2016	10
Iceland2016	10
Lithuania2016	10
Mauritius2016	10

1-10 of 16 rows Previous **1** 2 Next

I will remove the year 2016 and countries with data for less than a three year period

```
rate <- rate %>% filter(year != 2016)
minimum_years <- rate %>%
  group_by(country) %>%
  summarize(rows = n(),
            years = rows / 12) %>%
  arrange(years)

rate <- rate %>%
  filter(!(country %in% head(minimum_years$country, 7)))
```

I will remove columns that are not needed and missing values

```
rate = subset(rate, select = -c(country_year))
rate = subset(rate, select = -c(HDI))
rate <- subset(rate, !is.na(region))
sum(is.na(rate))
```

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

I will now clean up the data types

```
rate$age <- gsub(" years", "", rate$age)
rate_nominal <- c('country', 'sex', 'continent')
rate[rate_nominal, ] <-
  lapply(rate[rate_nominal, ], function(x) {
    factor(x)
  })

rate$age <- factor(
  rate$age,
  ordered = T,
  levels = c("5-14",
             "15-24",
             "25-34",
             "35-54",
             "55-74",
             "75+")
)

rate$generation <- factor(
  rate$generation,
  ordered = T,
  levels = c(
    "G.I. Generation",
    "Silent",
    "Boomers",
    "Generation X",
    "Millenials",
    "Generation Z"
  )
)
```

I will convert the data to a tibble so it will recognise issues earlier

```
rate <- as_tibble(rate)
```

I will calculate the global suicide rate average

```
global_average <-
  (sum(as.numeric(rate$suicides_no)) / sum(as.numeric(rate$population))) * 100000
```

I will once again check for duplicates and missing data

```
rate <- distinct(rate)
summary(rate)
```

```
##    country               year           sex                 age
##  Length:26821       Min.   :1985   Length:26821       5-14 :4470
##  Class :character   1st Qu.:1995   Class :character   15-24:4470
##  Mode  :character   Median :2002   Mode  :character   25-34:4470
##                     Mean   :2001                      35-54:4470
##                     3rd Qu.:2008                      55-74:4470
##                     Max.   :2015                      75+  :4470
##                     NA's   :1                         NA's :   1
##   suicides_no        population       suicides_100k       GDP_year
##  Min.   :    0.0   Min.   :     278   Min.   :  0.00   Min.   :4.692e+07
##  1st Qu.:    3.0   1st Qu.:  102588   1st Qu.:  0.99   1st Qu.:9.399e+09
##  Median :   25.0   Median :  435925   Median :  6.01   Median :4.811e+10
##  Mean   :  241.3   Mean   : 1857394   Mean   : 12.78   Mean   :4.517e+11
##  3rd Qu.:  129.0   3rd Qu.: 1456899   3rd Qu.: 16.60   3rd Qu.:2.579e+11
##  Max.   :22338.0   Max.   :43805214   Max.   :224.97   Max.   :1.812e+13
##  NA's   :1         NA's   :1          NA's   :1        NA's   :1
##  GDP_per_capita              generation      region            subregion
##  Min.   :   251   G.I. Generation:2654   Length:26821       Length:26821
##  1st Qu.:  3397   Silent         :6146   Class :character   Class :character
##  Median :  9387   Boomers        :4810   Mode  :character   Mode  :character
##  Mean   : 16959   Generation X   :6178
##  3rd Qu.: 25191   Millenials     :5610
##  Max.   :126352   Generation Z   :1422
##  NA's   :1        NA's           :   1
```

```
rate <- subset(rate, !is.na(generation))
```

Analysing the variables

I will now analyse the numerical variables

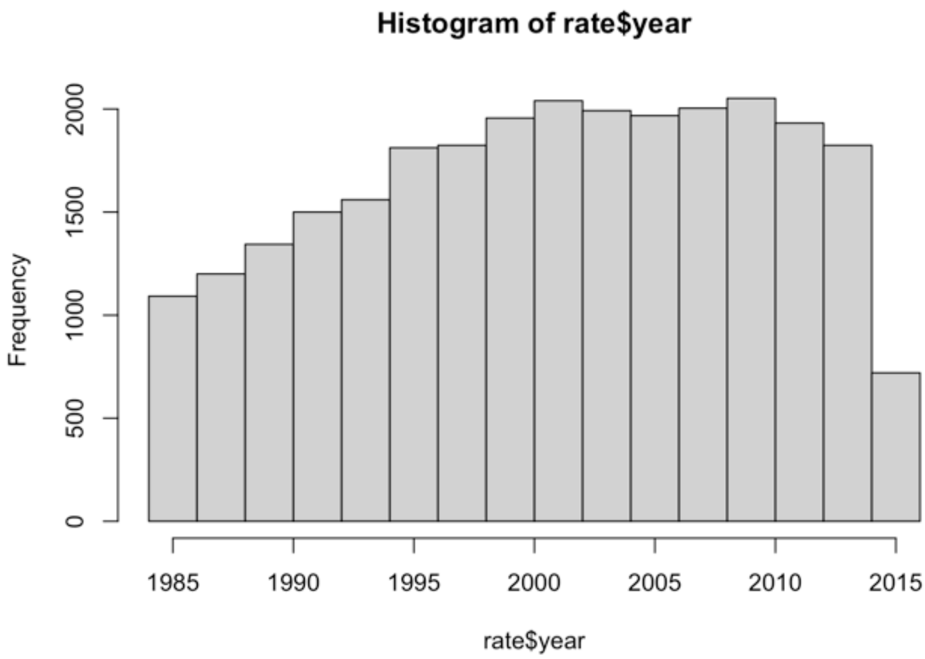
```
glimpse(rate)
```

```
## Rows: 26,820
## Columns: 12
## $ country        <chr> "Albania", "Albania", "Albania", "Albania", "Albania", …
## $ year           <dbl> 1995, 1995, 1995, 1999, 1999, 1999, 1999, 1999, 1999, 1…
## $ sex            <chr> "male", "male", "male", "male", "male", "male", "female…
## $ age            <ord> 75+, 35-54, 15-24, 35-54, 25-34, 55-74, 15-24, 25-34, 5…
## $ suicides_no    <dbl> 1, 14, 11, 31, 19, 14, 19, 13, 6, 5, 13, 10, 12, 9, 7, …
## $ population     <dbl> 25100, 375900, 241200, 391100, 242300, 185200, 296800, …
## $ suicides_100k  <dbl> 3.98, 3.72, 4.56, 7.93, 7.84, 7.56, 6.40, 4.70, 3.18, 1…
## $ GDP_year       <dbl> 2424499009, 2424499009, 2424499009, 3414760915, 3414760…
## $ GDP_per_capita <dbl> 835, 835, 835, 1127, 1127, 1127, 1127, 1127, 1127, 1127…
## $ generation     <ord> G.I. Generation, Boomers, Generation X, Boomers, Genera…
## $ region         <chr> "Europe", "Europe", "Europe", "Europe", "Europe", "Euro…
## $ subregion      <chr> "Southern Europe", "Southern Europe", "Southern Europe"…
```

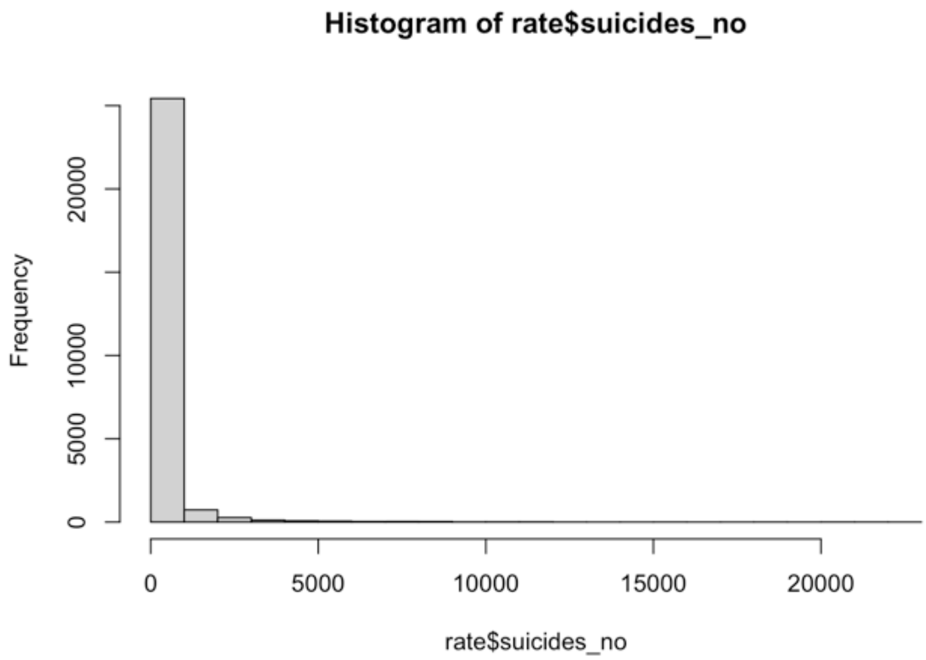
```
str(rate)
```

```
## tibble [26,820 × 12] (S3: tbl_df/tbl/data.frame)
##  $ country       : chr [1:26820] "Albania" "Albania" "Albania" "Albania" ...
##  $ year          : num [1:26820] 1995 1995 1995 1999 1999 ...
##  $ sex           : chr [1:26820] "male" "male" "male" "male" ...
##  $ age           : Ord.factor w/ 6 levels "5-14"<"15-24"<..: 6 4 2 4 3 5 2 3 5 4 ...
##  $ suicides_no   : num [1:26820] 1 14 11 31 19 14 19 13 6 5 ...
##  $ population    : num [1:26820] 25100 375900 241200 391100 242300 ...
##  $ suicides_100k : num [1:26820] 3.98 3.72 4.56 7.93 7.84 7.56 6.4 4.7 3.18 1.34 ...
##  $ GDP_year      : num [1:26820] 2.42e+09 2.42e+09 2.42e+09 3.41e+09 3.41e+09 ...
##  $ GDP_per_capita: num [1:26820] 835 835 835 1127 1127 ...
##  $ generation    : Ord.factor w/ 6 levels "G.I. Generation"<..: 1 3 4 3 4 2 4 4 2 3 ...
##  $ region        : chr [1:26820] "Europe" "Europe" "Europe" "Europe" ...
##  $ subregion     : chr [1:26820] "Southern Europe" "Southern Europe" "Southern Europe" "Southern Europe" ...
```

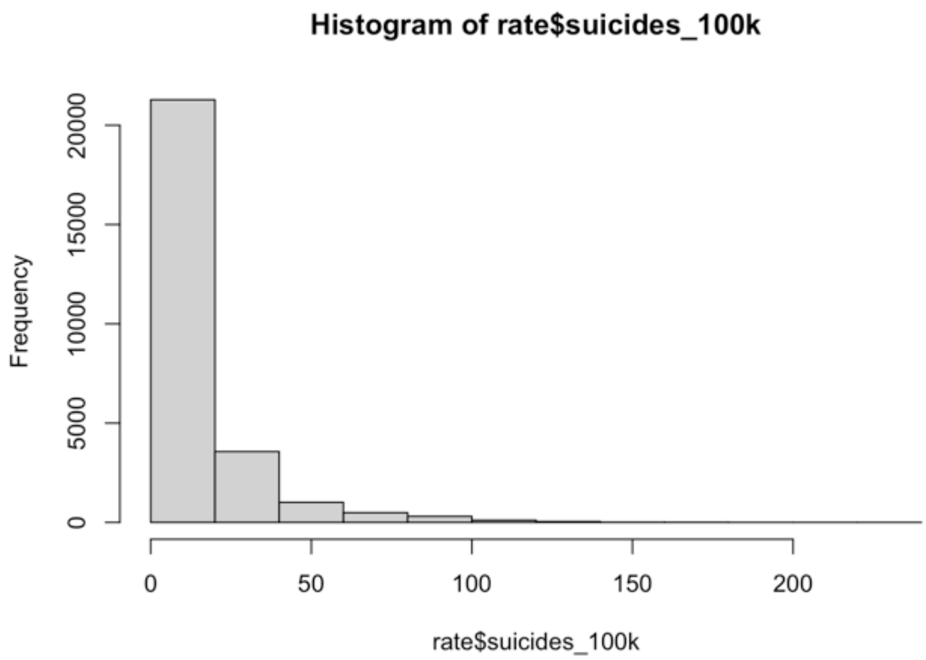
```
hist(rate$year)
```



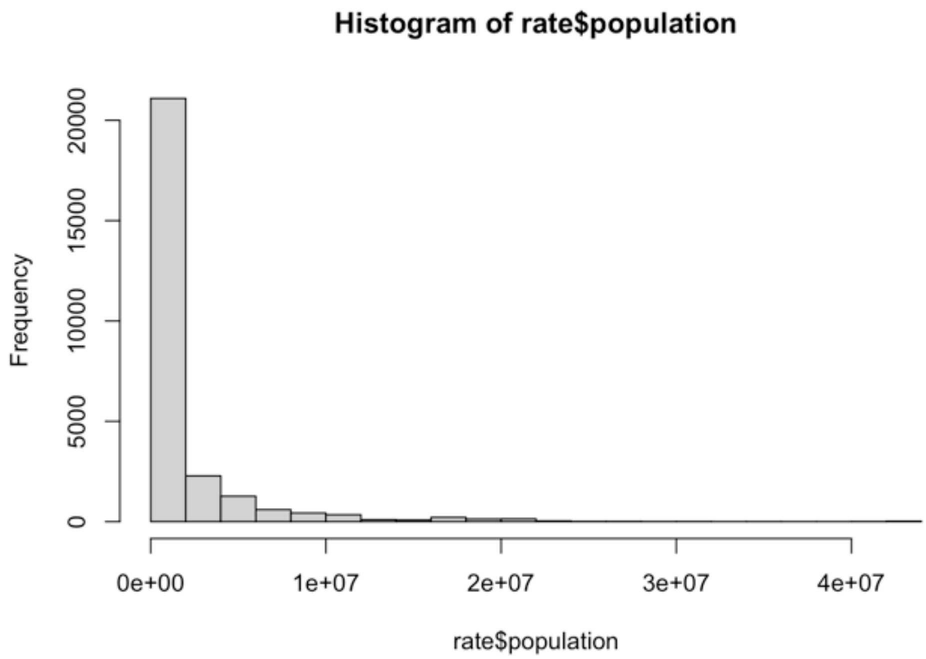

```
hist(rate$suicides_no)
```



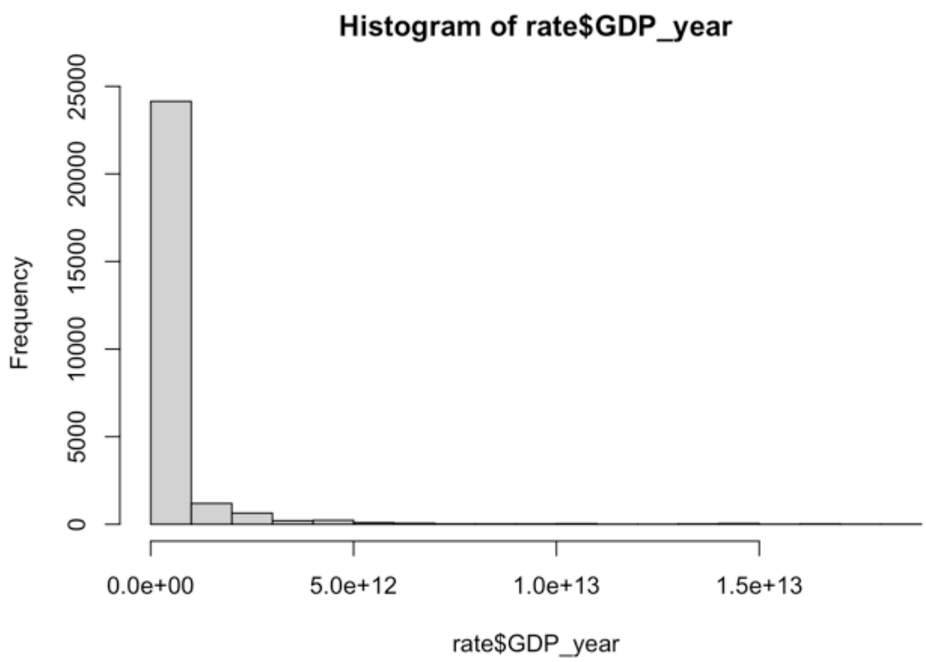

```
hist(rate$suicides_100k)
```



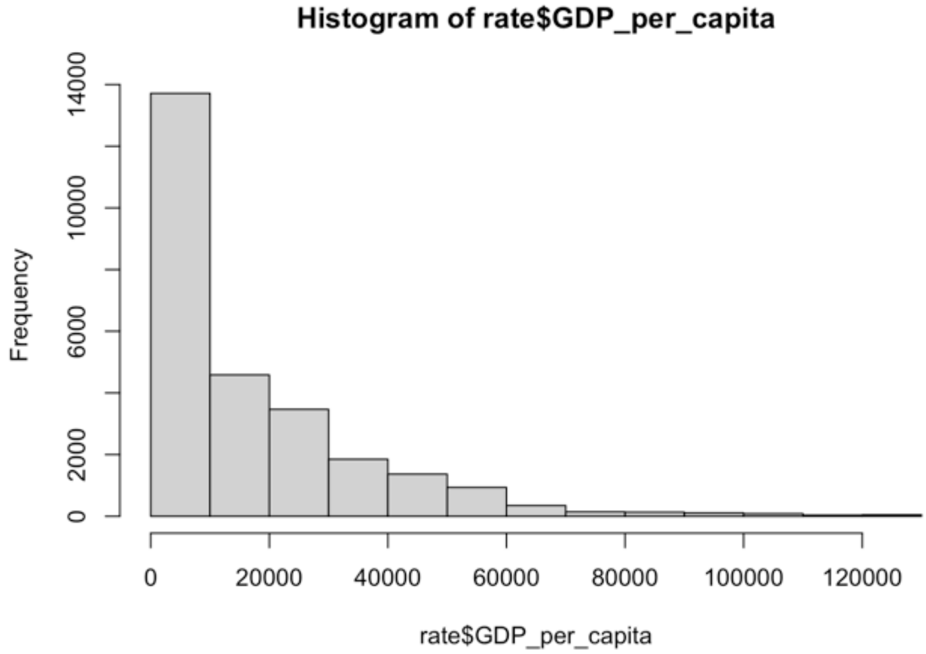

```
hist(rate$population)
```




```
hist(rate$GDP_year)
```




```
hist(rate$GDP_per_capita)
```



I will analyse the ordered factor variables

```
table(rate$age)
```

```
##
##  5-14 15-24 25-34 35-54 55-74   75+
##  4470  4470  4470  4470  4470  4470
```

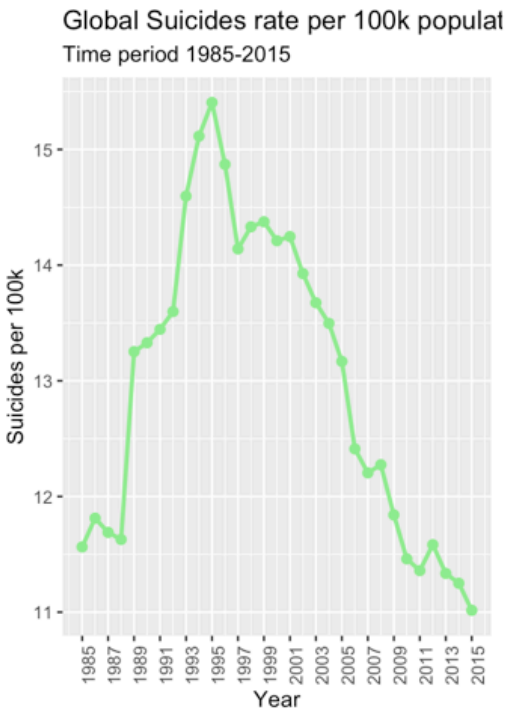
```
table(rate$generation)
```

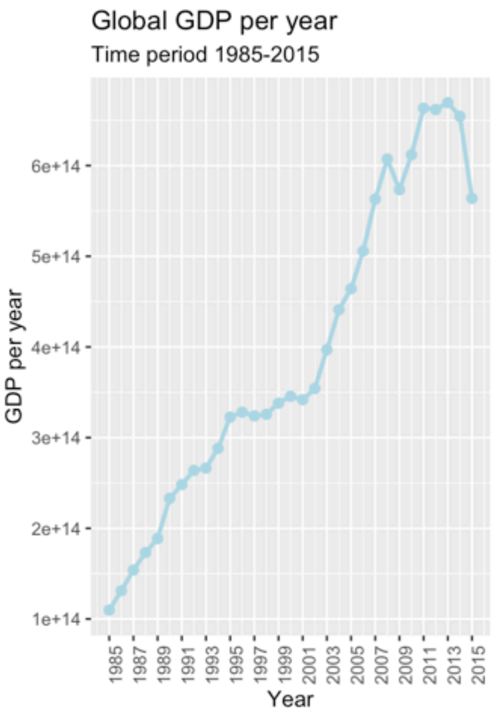
```
##
## G.I. Generation          Silent         Boomers    Generation X      Millenials
##            2654            6146            4810            6178            5610
##    Generation Z
##            1422
```

Visualising the data

The first diagram is the global rate of suicide over the years and the countries´ GDP over the years.

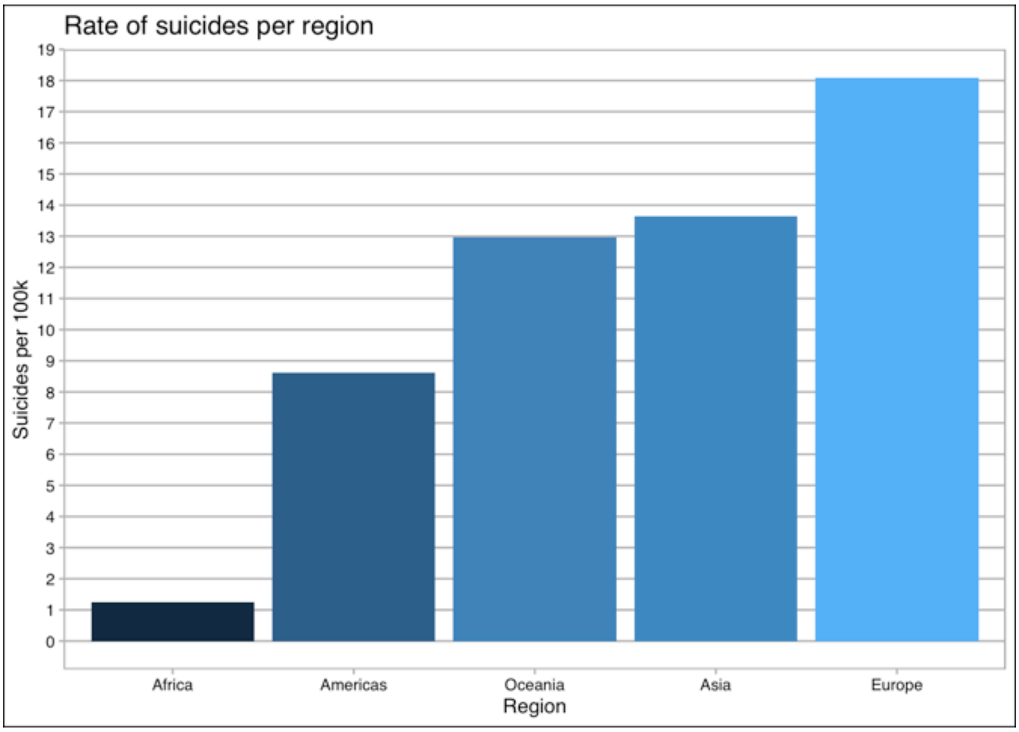
```
global_rate <- rate %>%
  group_by(year) %>%
  summarize(
    population = sum(population),
    suicides = sum(suicides_no),
    suicides_per_100k = (suicides / population) * 100000
  ) %>%
  ggplot(aes(x = year, y = suicides_per_100k)) +
  geom_line(col = "lightgreen", size = 1) +
  geom_point(col = "lightgreen", size = 2) +
  geom_hline(
    yintercept = global_average,
    linetype = 2,
    color = "darkgreen",
    size = 1
  ) +
  labs(
    title = "Global Suicides rate per 100k population",
    subtitle = "Time period 1985-2015",
    x = "Year",
    y = "Suicides per 100k"
  ) +
  scale_x_continuous(breaks = seq(1985, 2015, 2)) +
  scale_y_continuous(breaks = seq(10, 20)) + theme(axis.text.x = element_text(angle = 90))

global_gdp <- rate %>%
  group_by(year) %>%
  summarize(GDP_year = sum(GDP_year)) %>%
  ggplot(aes(x = year, y = GDP_year)) +
  geom_line(col = "lightblue", size = 1) +
  geom_point(col = "lightblue", size = 2) +
  geom_hline(
    yintercept = global_average,
    linetype = 2,
    color = "darkgreen",
    size = 1
  ) +
  labs(
    title = "Global GDP per year",
    subtitle = "Time period 1985-2015",
    x = "Year",
    y = "GDP per year"
  ) +
  scale_x_continuous(breaks = seq(1985, 2015, 2)) + theme(axis.text.x = element_text(angle = 90))

grid.arrange(global_rate, global_gdp, ncol = 2)
```





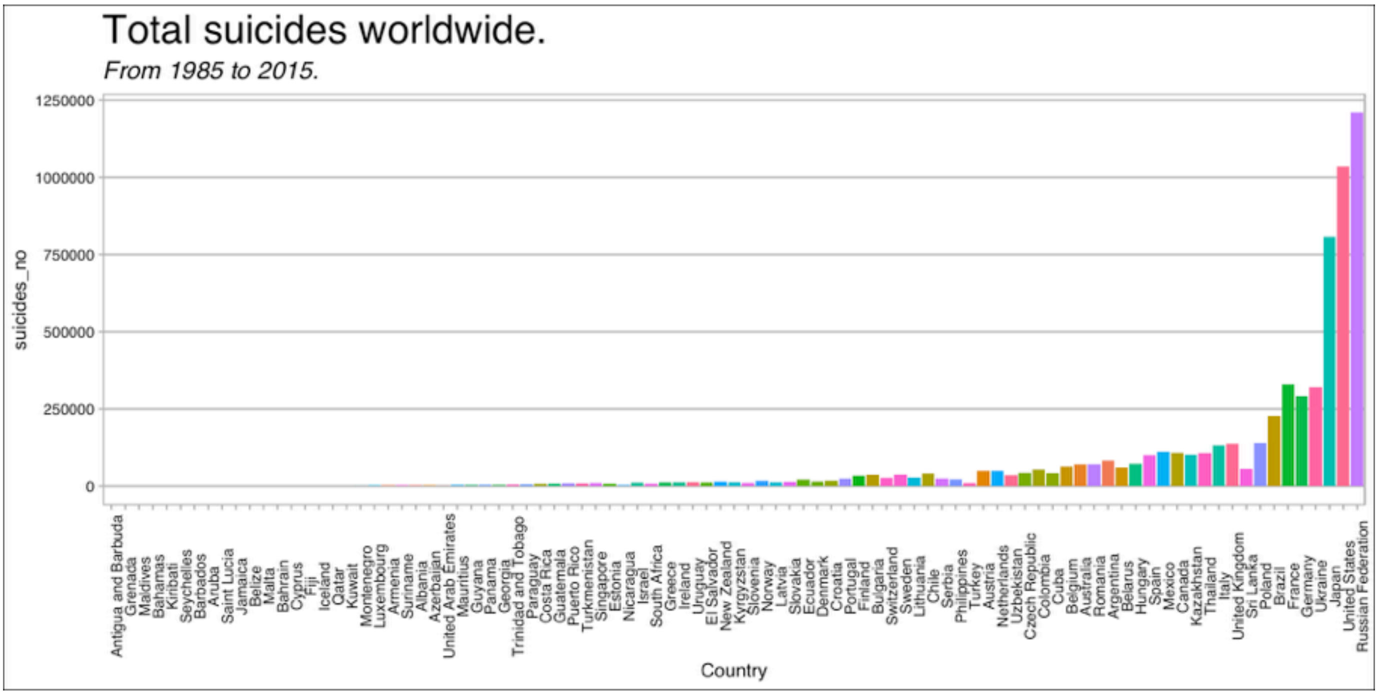
I will look at the difference of suicide numbers between the continents

```
continent <- rate %>%
  group_by(region) %>%
  summarize(suicide_per_100k = (sum(as.numeric(suicides_no)) / sum(as.numeric(population))) * 100000) %>%
  arrange(suicide_per_100k)

continent$region <-
  factor(continent$region,
         ordered = T,
         levels = continent$region)

ggplot(continent) +
  aes(
    x = region,
    y = suicide_per_100k,
    fill = suicide_per_100k,
    weight = region
  ) +
  geom_bar(aes(reorder(region, suicide_per_100k)), position = "dodge", stat = "identity") +
  labs(x = "Region",
       y = "Suicides per 100k",
       title = "Rate of suicides per region",
       fill = "Rate of suicides") + theme_calc() +
  theme(legend.position = "none", title = element_text(size = 10)) +
  scale_y_continuous(breaks = seq(0, 20, 1), minor_breaks = F)
```



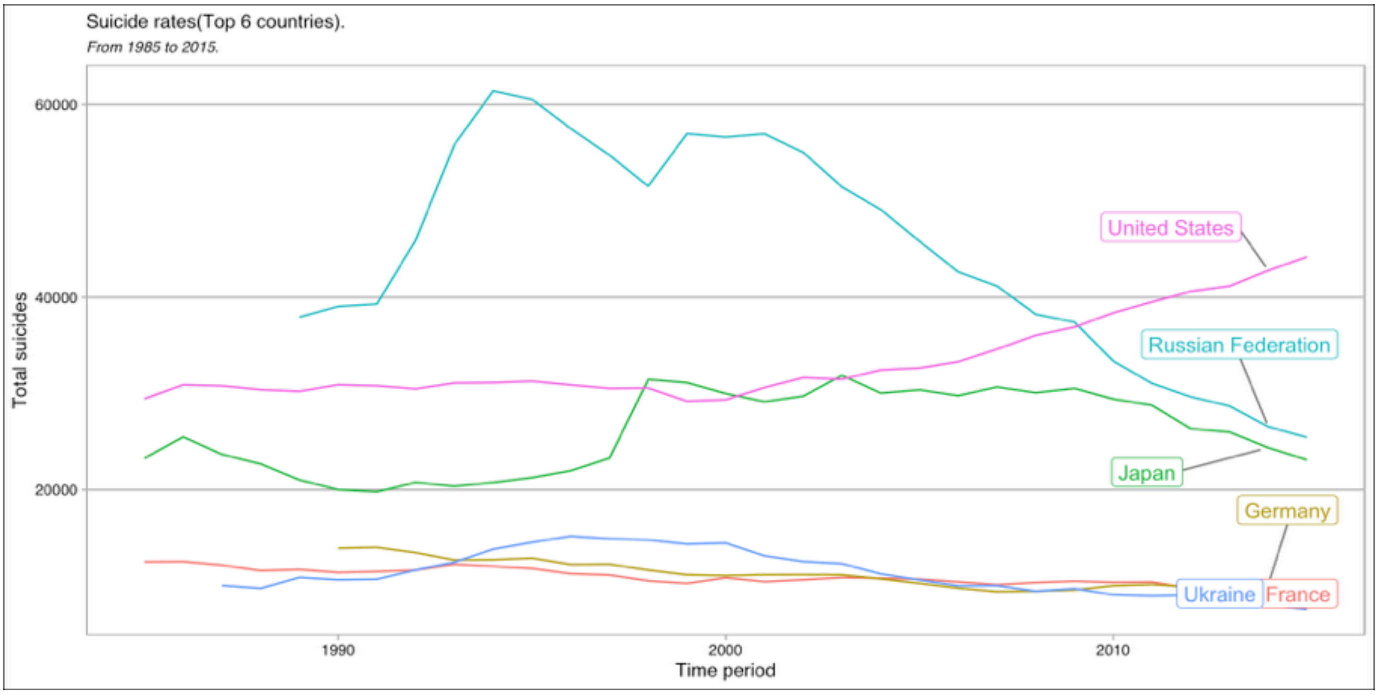
I will now look at the countries´ suicide numbers alone

```
ggplot(data = rate, aes(
  x = reorder(`country`, `suicides_no`),
  y = `suicides_no`,
  fill = `country`
)) + geom_col() + theme_calc() +
  theme(
    legend.position = "none",
    plot.subtitle = element_text(face = "italic", size = 13) ,
    plot.title = element_text(size = 21)
  ) +
  labs(x = "Country",
       title = "Total suicides worldwide.",
       subtitle = "From 1985 to 2015.") + theme(axis.text.x = element_text(angle = 90))
```



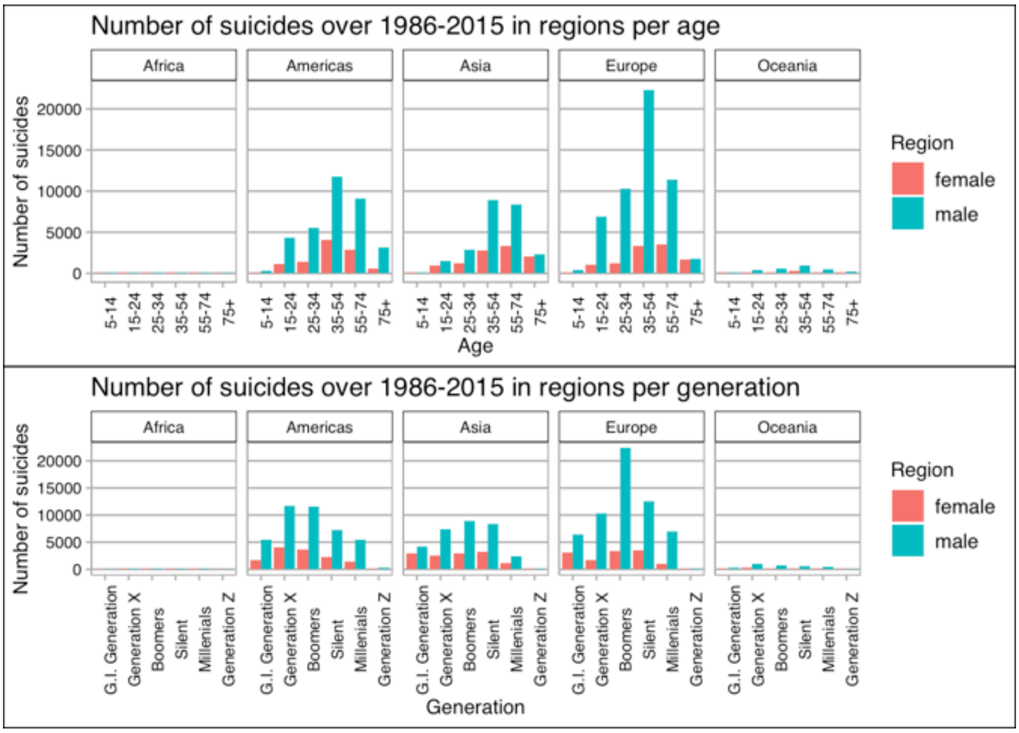
I will now analyse the six countries that have the highest suicide numbers

```
rate_top_6 <-
  rate %>% group_by(`country`, `year`) %>% summarize("Total Suicides per year" =sum(`suicides_no`)) %>% filter(`c
ountry` == "Russian Federation" ||`country` == "United States" ||`country` == "Japan" ||`country` == "France"
||`country` == "Ukraine" ||`country` == "Germany")

ggplot(data = rate_top_6,
       aes(x = `year`, y = `Total Suicides per year`, color = `country`)) + geom_line() +
  theme_calc() +
  geom_label_repel(
    aes(
      `year`,
      `Total Suicides per year`,
      label = ifelse(`year` == 2014, as.character(`country`), '')
    ),
    box.padding = 1,
    point.padding = 0.5,
    segment.color = 'gray50',
    max.overlaps = 30
  ) +
  labs(
    title = "Suicide rates(Top 6 countries).",
    subtitle = "From 1985 to 2015.",
    y = "Total suicides",
    x = "Time period"
  ) +
  theme(
    legend.position = "none",
    plot.subtitle = element_text(face = "italic", size = 8) ,
    plot.title = element_text(size = 10)
  )
```



I will analyse the differences in suicide numbers between gender, age and generation over the different regions

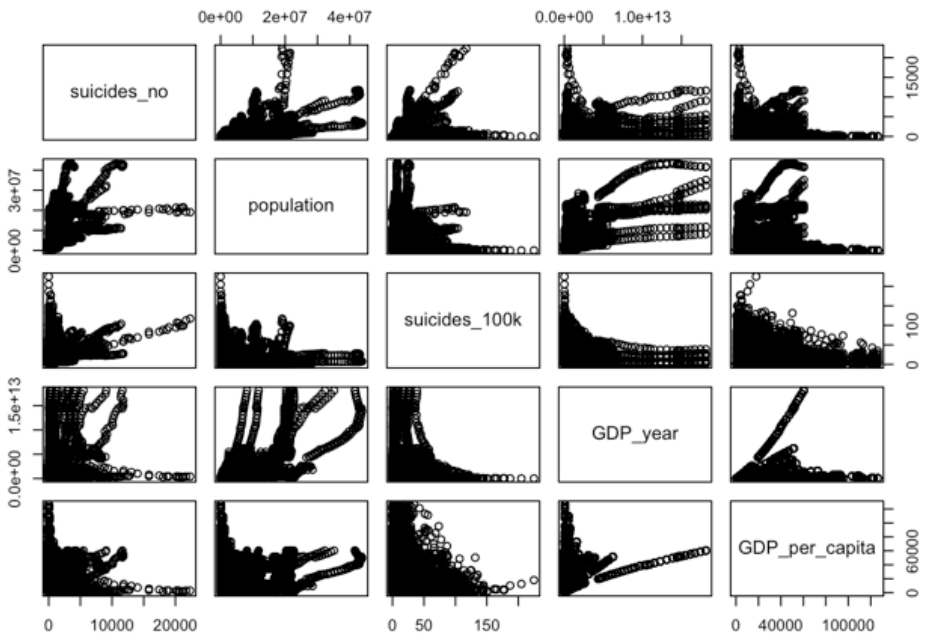
```
suicide_generation <- ggplot(rate) +
  aes(x = generation,
      y = suicides_no,
      fill = sex) +
  geom_bar(aes(reorder(generation,+year)), position = "dodge", stat = "identity") +
  labs(x = "Generation",
       y = "Number of suicides",
       title = "Number of suicides over 1986-2015 in regions per generation",
       fill = "Region") +
  facet_grid(vars(), vars(region)) + theme_calc() + theme(axis.text.x = element_text(angle = 90))

suicide_age <- ggplot(rate) +
  aes(x = age,
      y = suicides_no,
      fill = sex) +
  geom_bar(aes(reorder(age,+year)), position = "dodge", stat = "identity") +
  labs(x = "Age",
       y = "Number of suicides",
       title = "Number of suicides over 1986-2015 in regions per age",
       fill = "Region") +
  facet_grid(vars(), vars(region)) + theme_calc() + theme(axis.text.x = element_text(angle = 90))

grid.arrange(suicide_age, suicide_generation, nrow = 2)
```



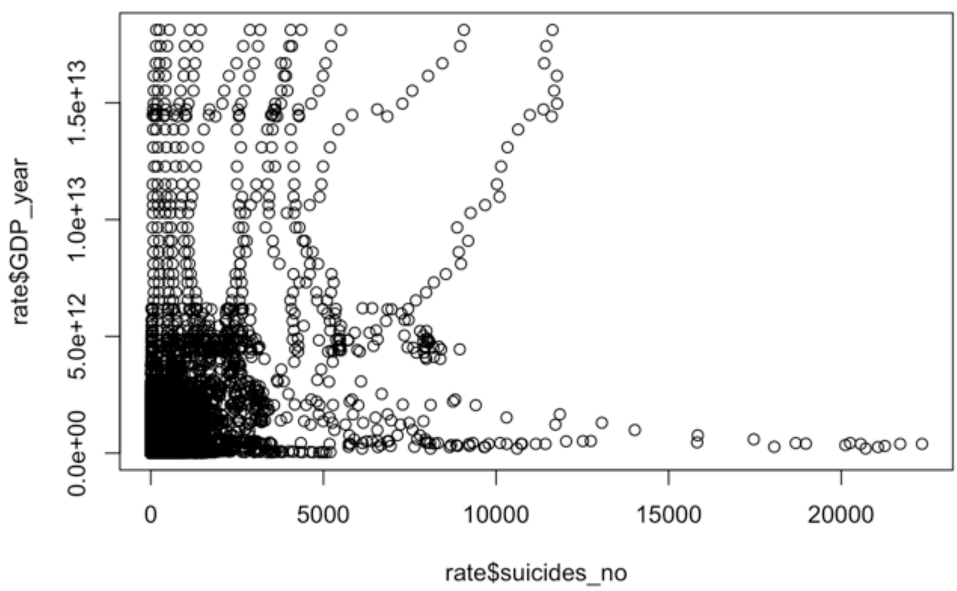
Cluster Analysis

In this part I have undertaken a hierarchical cluster analysis. First I looked at the different variables and their connections

```
pairs(rate[5:9])
```




```
plot(rate$GDP_year ~
       rate$suicides_no, data = rate)
```



I will use solely numerical values and check for missing values. I will also use the `scale()` function in order to normalise the columns I am utilizing.

```
rate_no_na <- rate %>% na.omit()
rate_sg = na.omit(subset(
  rate,
  select = c(suicides_no, GDP_per_capita, suicides_100k, GDP_year)
))
```

```
rate_sg_scale <- scale(rate_sg)
glimpse(rate_sg_scale)
```

```
##  num [1:26820, 1:4] -0.264 -0.249 -0.253 -0.231 -0.244 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:4] "suicides_no" "GDP_per_capita" "suicides_100k" "GDP_year"
##  - attr(*, "scaled:center")= Named num [1:4] 2.41e+02 1.70e+04 1.28e+01 4.52e+11
##   ..- attr(*, "names")= chr [1:4] "suicides_no" "GDP_per_capita" "suicides_100k" "GDP_year"
##  - attr(*, "scaled:scale")= Named num [1:4] 9.11e+02 1.90e+04 1.87e+01 1.48e+12
##   ..- attr(*, "names")= chr [1:4] "suicides_no" "GDP_per_capita" "suicides_100k" "GDP_year"
```

I will measure the distance

```
rate_dist <- dist(rate_sg_scale)
```

I will create a hierarchical clustering algorithm

```
rate_hg <- hclust(rate_dist, method = "complete")
rate_hg
```

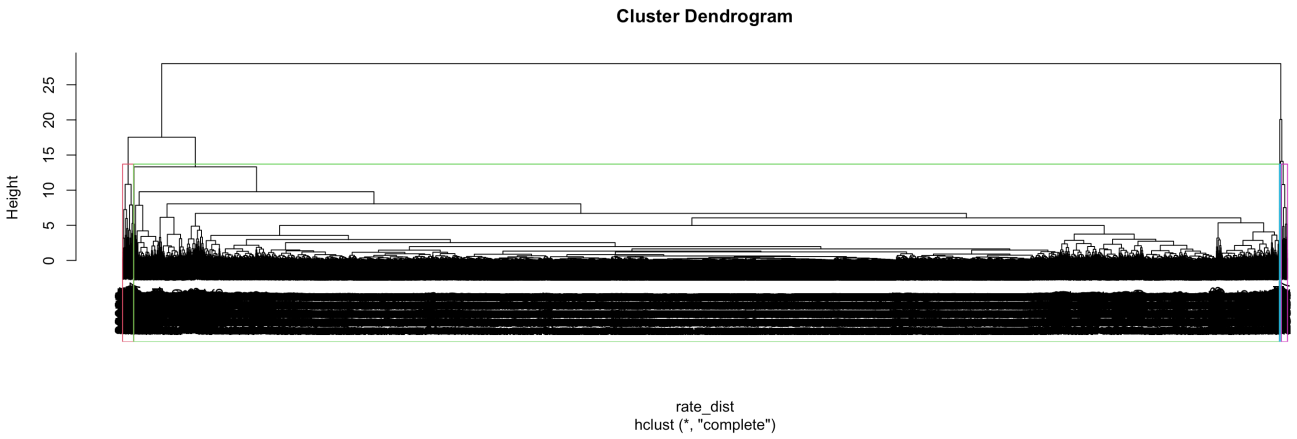
```
##
## Call:
## hclust(d = rate_dist, method = "complete")
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 26820
```

I will use the Dendrogram to identify clusters

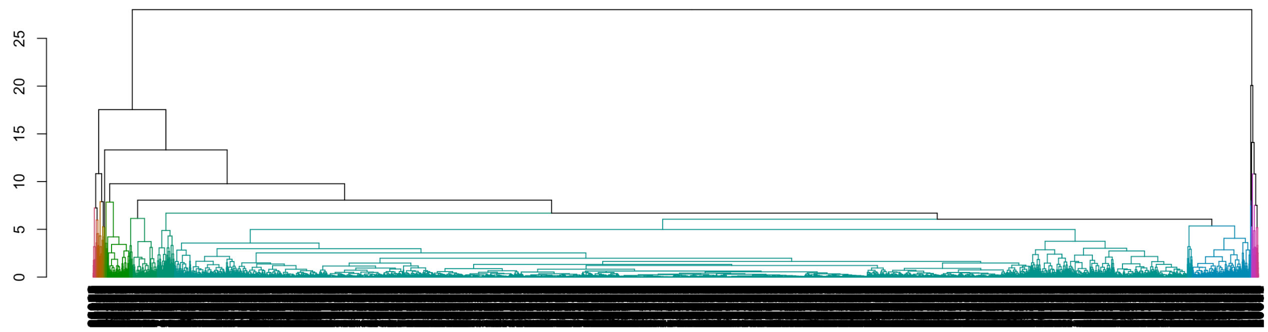
```
plot(rate_hg)
unique(rate$region)
```

```
## [1] "Europe"   "Americas" "Asia"     "Oceania"  "Africa"
```

```
rect.hclust(rate_hg, k = 5, border = 2:6)
```




```
dendrogram <- as.dendrogram(rate_hg)
dendrogram_colour<-color_branches(dendrogram, h=5)
plot(dendrogram_colour)
```



I will look at the clusters

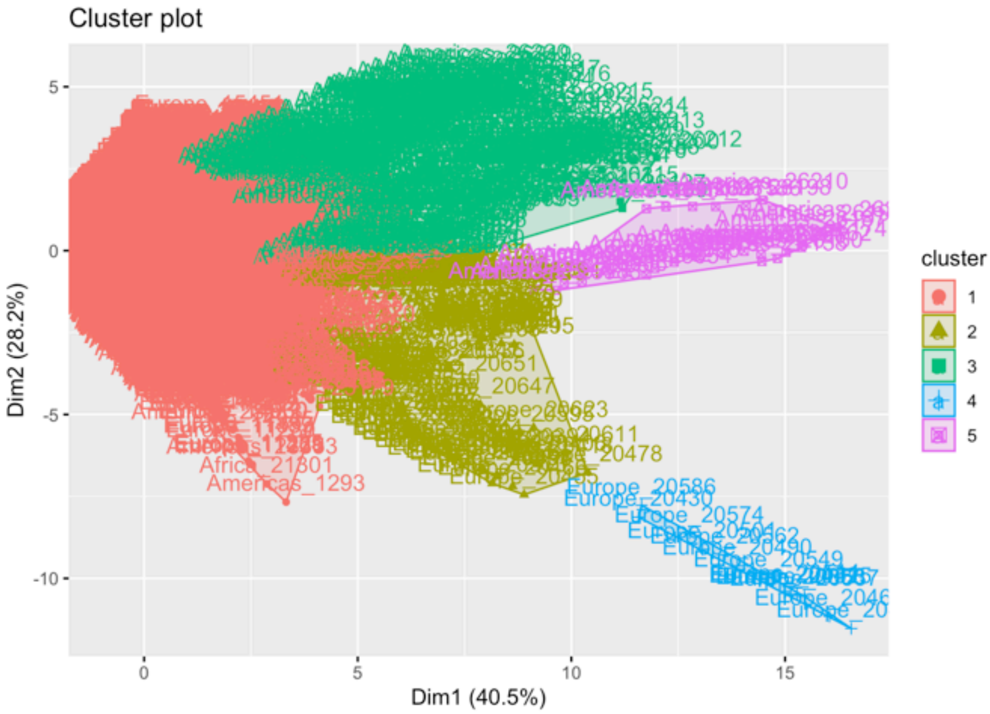
```
rate_clusters <- cutree(rate_hg, k = 5)
glimpse(rate_clusters)
```

```
##  int [1:26820] 1 1 1 1 1 1 1 1 1 1 ...
```

```
rate_no_na <- rate %>% na.omit()
rate_sr <- rate_no_na %>% mutate(cluster = rate_clusters)
```

I will visualise the clusters

```
rownames(rate_sg_scale) <-
  paste(rate$region, 1:dim(rate)[1], sep = "_")

fviz_cluster(list(data = rate_sg_scale, cluster = rate_clusters))
```




```
table(rate_clusters, rate$region)
```

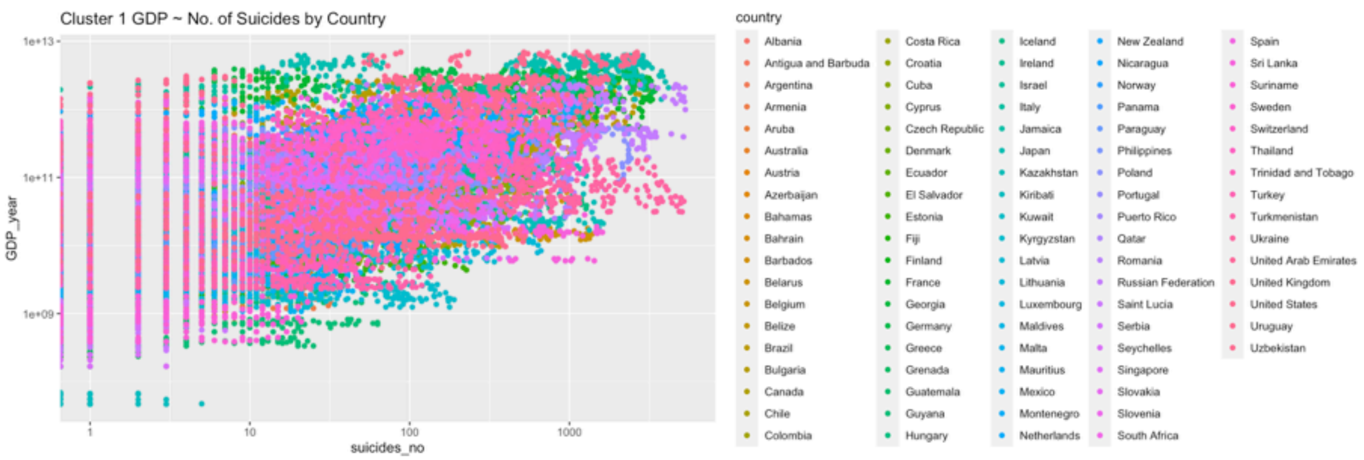
```
##
## rate_clusters Africa Americas  Asia Europe Oceania
##             1    828     8554  4841  11184     972
##             2      0       24    50     70       0
##             3      0      251     5      0       0
##             4      0        0     0     14       0
##             5      0       27     0      0       0
```

```
sum = subset(rate, select = -c(country, age, sex, generation, region, subregion)) %>% summary()
sum
```

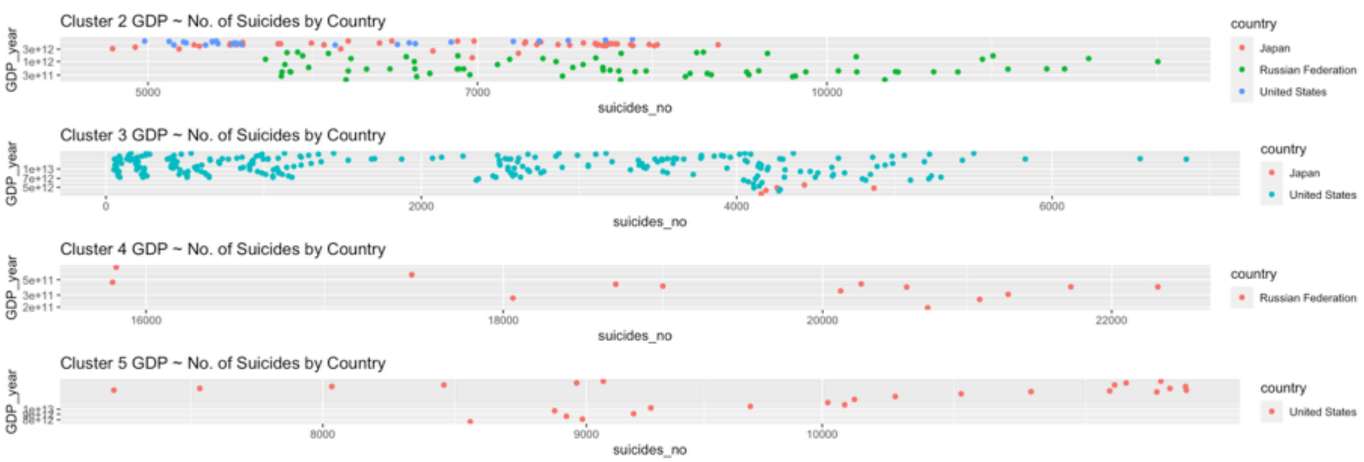
```
##       year       suicides_no        population       suicides_100k
##  Min.   :1985   Min.   :    0.0   Min.   :     278   Min.   :  0.00
##  1st Qu.:1995   1st Qu.:    3.0   1st Qu.:  102588   1st Qu.:  0.99
##  Median :2002   Median :   25.0   Median :  435925   Median :  6.01
##  Mean   :2001   Mean   :  241.3   Mean   : 1857394   Mean   : 12.78
##  3rd Qu.:2008   3rd Qu.:  129.0   3rd Qu.: 1456899   3rd Qu.: 16.60
##  Max.   :2015   Max.   :22338.0   Max.   :43805214   Max.   :224.97
##     GDP_year          GDP_per_capita
##  Min.   :4.692e+07   Min.   :   251
##  1st Qu.:9.399e+09   1st Qu.:  3397
##  Median :4.811e+10   Median :  9387
##  Mean   :4.517e+11   Mean   : 16959
##  3rd Qu.:2.579e+11   3rd Qu.: 25191
##  Max.   :1.812e+13   Max.   :126352
```

I will now look at each of the clusters and see their characteristics, this will be done on country, age, generation, gender, population and GDP per capita level

```
cluster1 <- rate_sr %>%
  filter(rate_clusters ==  1) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = country)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 1 GDP ~ No. of Suicides by Country")

cluster2 <- rate_sr %>%
  filter(rate_clusters ==  2) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = country)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 2 GDP ~ No. of Suicides by Country")

cluster3 <- rate_sr %>%
  filter(rate_clusters ==  3) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = country)) +
  geom_point() +
  scale_y_log10() + ggtitle("Cluster 3 GDP ~ No. of Suicides by Country")

cluster4 <- rate_sr %>%
  filter(rate_clusters ==  4) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = country)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 4 GDP ~ No. of Suicides by Country")

cluster5 <- rate_sr %>%
  filter(rate_clusters ==  5) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = country)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 5 GDP ~ No. of Suicides by Country")

cluster1
```

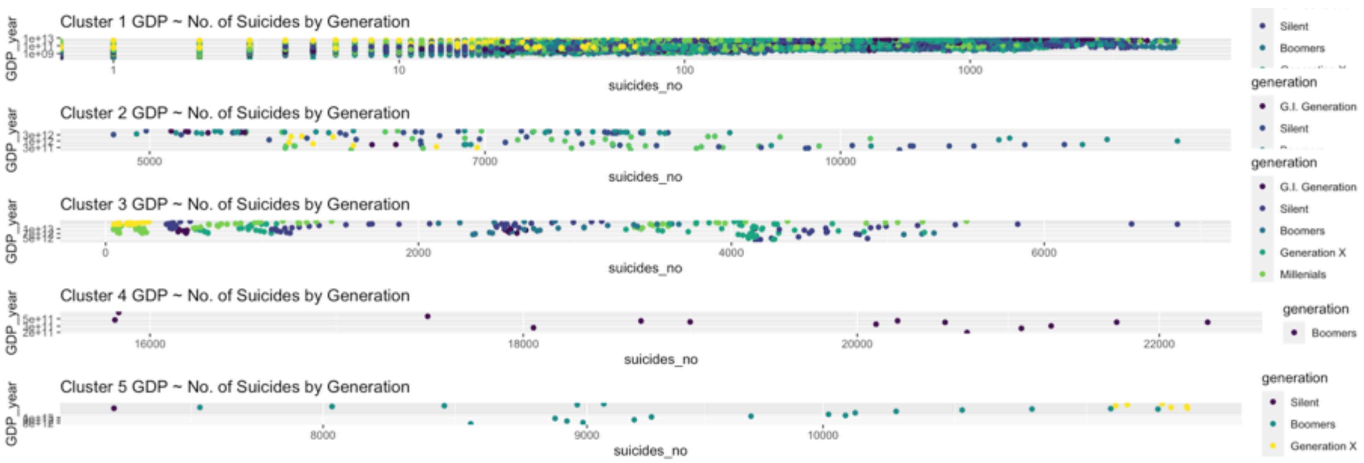



```
grid.arrange(cluster2, cluster3, cluster4, cluster5, nrow=4)
```

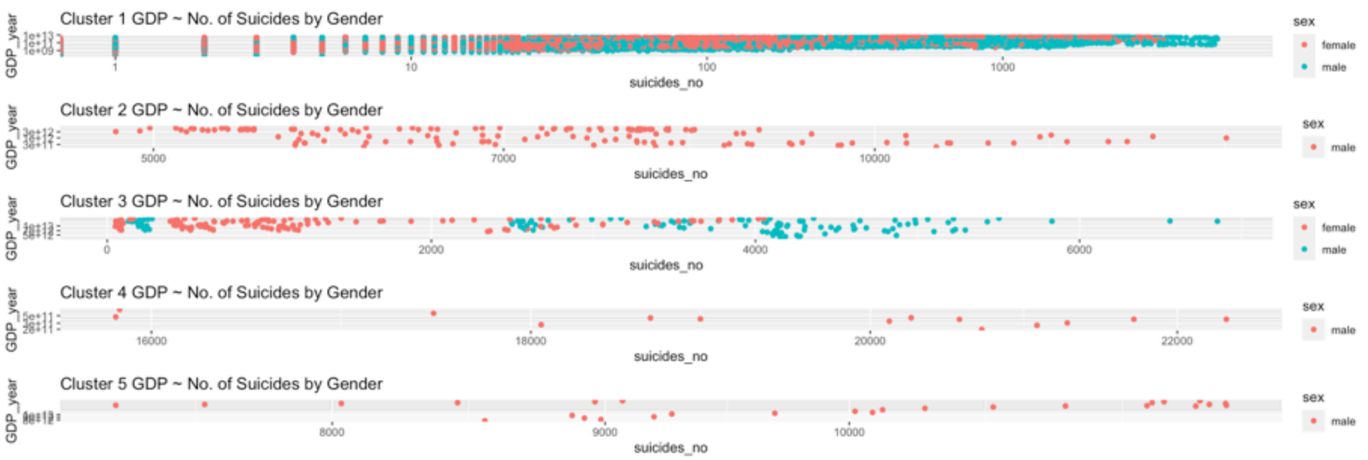


```r
cluster1 <- rate_sr %>%
  filter(rate_clusters ==  1) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = age)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 1 GDP ~ No. of Suicides by Age")

cluster2 <- rate_sr %>%
  filter(rate_clusters ==  2) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = age)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 2 GDP ~ No. of Suicides by Age")

cluster3 <- rate_sr %>%
  filter(rate_clusters ==  3) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = age)) +
  geom_point() +
  scale_y_log10() + ggtitle("Cluster 3 GDP ~ No. of Suicides by Age")

cluster4 <- rate_sr %>%
  filter(rate_clusters ==  4) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = age)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 4 GDP ~ No. of Suicides by Age")

cluster5 <- rate_sr %>%
  filter(rate_clusters ==  5) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = age)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 5 GDP ~ No. of Suicides by Age")

grid.arrange(cluster1, cluster2, cluster3, cluster4, cluster5, nrow=5)
```

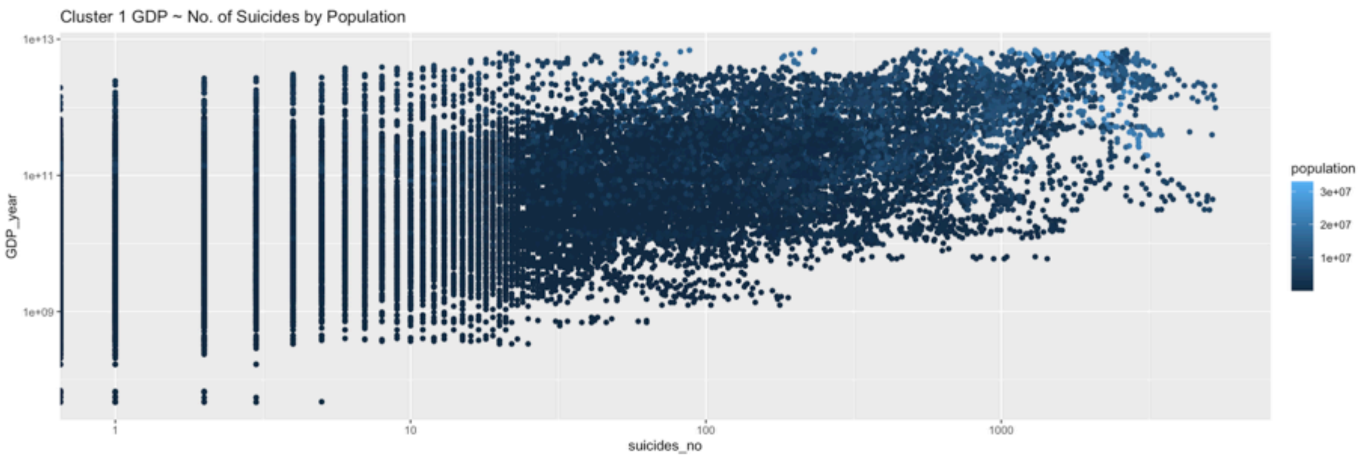
```
cluster1 <- rate_sr %>%
  filter(rate_clusters ==  1) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = generation)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 1 GDP ~ No. of Suicides by Generation")

cluster2 <- rate_sr %>%
  filter(rate_clusters ==  2) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = generation)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 2 GDP ~ No. of Suicides by Generation")

cluster3 <- rate_sr %>%
  filter(rate_clusters ==  3) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = generation)) +
  geom_point() +
  scale_y_log10() + ggtitle("Cluster 3 GDP ~ No. of Suicides by Generation")

cluster4 <- rate_sr %>%
  filter(rate_clusters ==  4) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = generation)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 4 GDP ~ No. of Suicides by Generation")

cluster5 <- rate_sr %>%
  filter(rate_clusters ==  5) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = generation)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 5 GDP ~ No. of Suicides by Generation")

grid.arrange(cluster1, cluster2, cluster3, cluster4, cluster5, nrow=5)
```



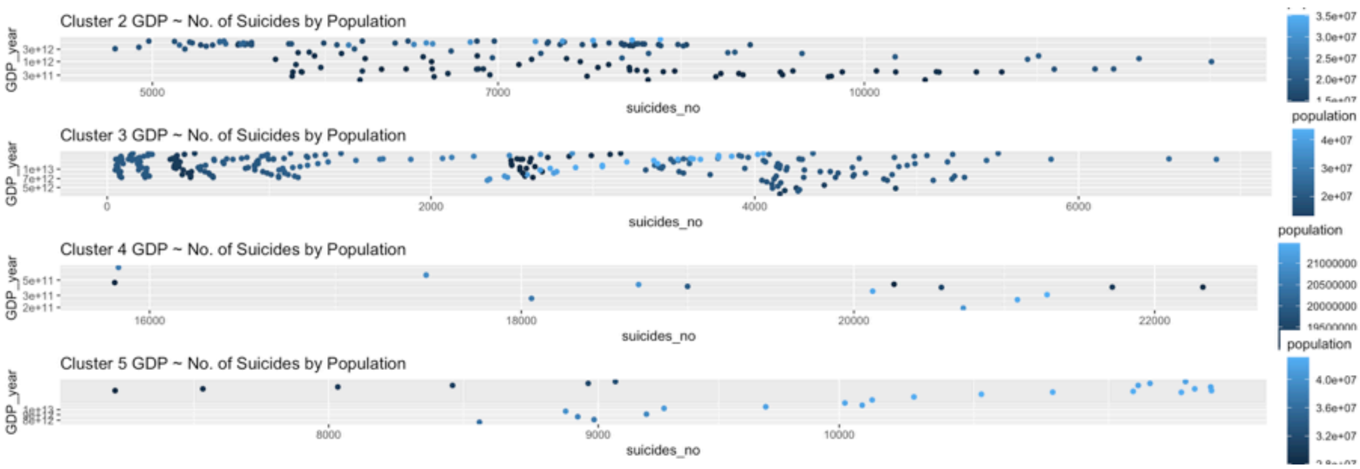
```
cluster1 <- rate_sr %>%
  filter(rate_clusters ==  1) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = sex)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 1 GDP ~ No. of Suicides by Gender")

cluster2 <- rate_sr %>%
  filter(rate_clusters ==  2) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = sex)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 2 GDP ~ No. of Suicides by Gender")

cluster3 <- rate_sr %>%
  filter(rate_clusters ==  3) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = sex)) +
  geom_point() +
  scale_y_log10() + ggtitle("Cluster 3 GDP ~ No. of Suicides by Gender")

cluster4 <- rate_sr %>%
  filter(rate_clusters ==  4) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = sex)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 4 GDP ~ No. of Suicides by Gender")

cluster5 <- rate_sr %>%
  filter(rate_clusters ==  5) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = sex)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 5 GDP ~ No. of Suicides by Gender")

grid.arrange(cluster1, cluster2, cluster3, cluster4, cluster5, nrow=5)
```



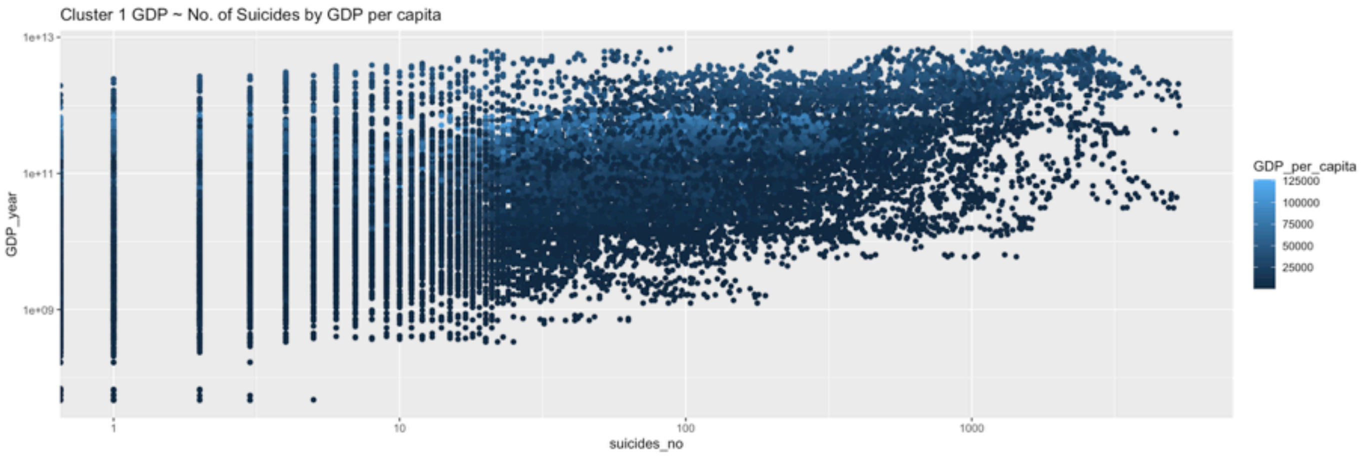
```
cluster1 <- rate_sr %>%
  filter(rate_clusters ==  1) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = population)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 1 GDP ~ No. of Suicides by Population")

cluster2 <- rate_sr %>%
  filter(rate_clusters ==  2) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = population)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 2 GDP ~ No. of Suicides by Population")

cluster3 <- rate_sr %>%
  filter(rate_clusters ==  3) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = population)) +
  geom_point() +
  scale_y_log10() + ggtitle("Cluster 3 GDP ~ No. of Suicides by Population")

cluster4 <- rate_sr %>%
  filter(rate_clusters ==  4) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = population)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 4 GDP ~ No. of Suicides by Population")

cluster5 <- rate_sr %>%
  filter(rate_clusters ==  5) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = population)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 5 GDP ~ No. of Suicides by Population")

cluster1
```



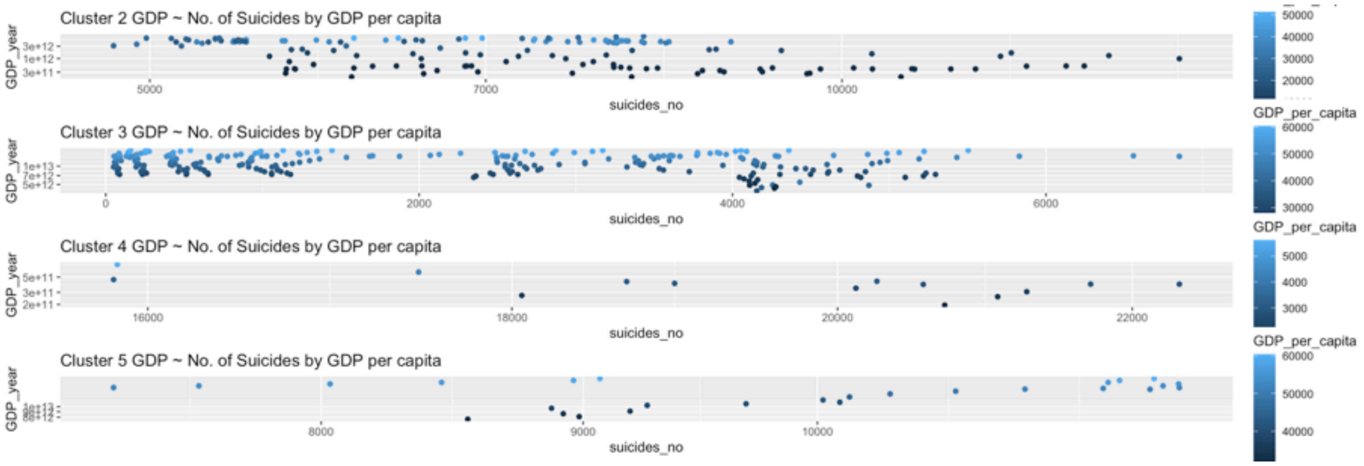

```
grid.arrange(cluster2, cluster3, cluster4, cluster5, nrow=4)
```



```
cluster1 <- rate_sr %>%
  filter(rate_clusters ==  1) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = GDP_per_capita)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 1 GDP ~ No. of Suicides by GDP per capita")

cluster2 <- rate_sr %>%
  filter(rate_clusters ==  2) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = GDP_per_capita)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 2 GDP ~ No. of Suicides by GDP per capita")

cluster3 <- rate_sr %>%
  filter(rate_clusters ==  3) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = GDP_per_capita)) +
  geom_point() +
  scale_y_log10() + ggtitle("Cluster 3 GDP ~ No. of Suicides by GDP per capita")

cluster4 <- rate_sr %>%
  filter(rate_clusters ==  4) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = GDP_per_capita)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 4 GDP ~ No. of Suicides by GDP per capita")

cluster5 <- rate_sr %>%
  filter(rate_clusters ==  5) %>%
  ggplot(aes(x = suicides_no, y = `GDP_year`, color = GDP_per_capita)) +
  geom_point() +
  scale_y_log10() +
  scale_x_log10() + ggtitle("Cluster 5 GDP ~ No. of Suicides by GDP per capita")

cluster1
```




```
grid.arrange(cluster2, cluster3, cluster4, cluster5, nrow=4)
```



Summarising

I will put the mean of clusters in a table and introduce another variable which will analyse the population

```
population <- rate_sr %>% summarize(
  suicides_no = mean(suicides_no),
  suicides_100k = mean(suicides_100k),
  GDP_year = mean(GDP_year),
  GDP_per_capita = mean(GDP_per_capita)
) %>% mutate(cluster = "population")

summary <-
  rate_sr %>% group_by(cluster) %>% summarize(
    suicides_no = mean(suicides_no),
    suicides_100k = mean(suicides_100k),
    GDP_year = mean(GDP_year),
    GDP_per_capita = mean(GDP_per_capita)
  )


summary <- add_row(
  summary,
  cluster = 6,
  suicides_no = 206.1243,
  suicides_100k = 11.99194,
  GDP_year = 547663851141,
  GDP_per_capita = 21074.37,
  .before = 1
)

summary <- as.data.frame(t(summary))
summary$profiling_var <- rownames(summary)
summary <- summary[-c(1),]
colnames(summary) <-
  c("Population",
    "Cluster 1",
    "Cluster 2",
    "Cluster 3",
    "Cluster 4",
    "Cluster 5")
colnames(summary)[7] <- "profiling_var"

summary <- tbl_df(summary)
summary <- summary %>% mutate_at(vars(1:6), as.character)
summary <- summary %>% mutate_at(vars(1:6), as.numeric)
summary %>% datatable()
```

Show 10 **entries**

Search:

	Population	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	profiling_va
1	206.1243	162.628113271921	7451.61805555556	2214.671875	19489.6428571429	9915.85185185185	suicides_no
2	11.99194	12.507055612419	52.1757638888889	12.7042578125	96.4157142857143	25.6225925925926	suicides_100k
3	547663851141	313923362560.361	2870527366221.72	11899185810828.4	403306516391.786	13596727370370.4	GDP_year
4	21074.37	16656.6620038667	19967.2222222222	43890.71484375	2926.35714285714	48303.8888888889	GDP_per_cap

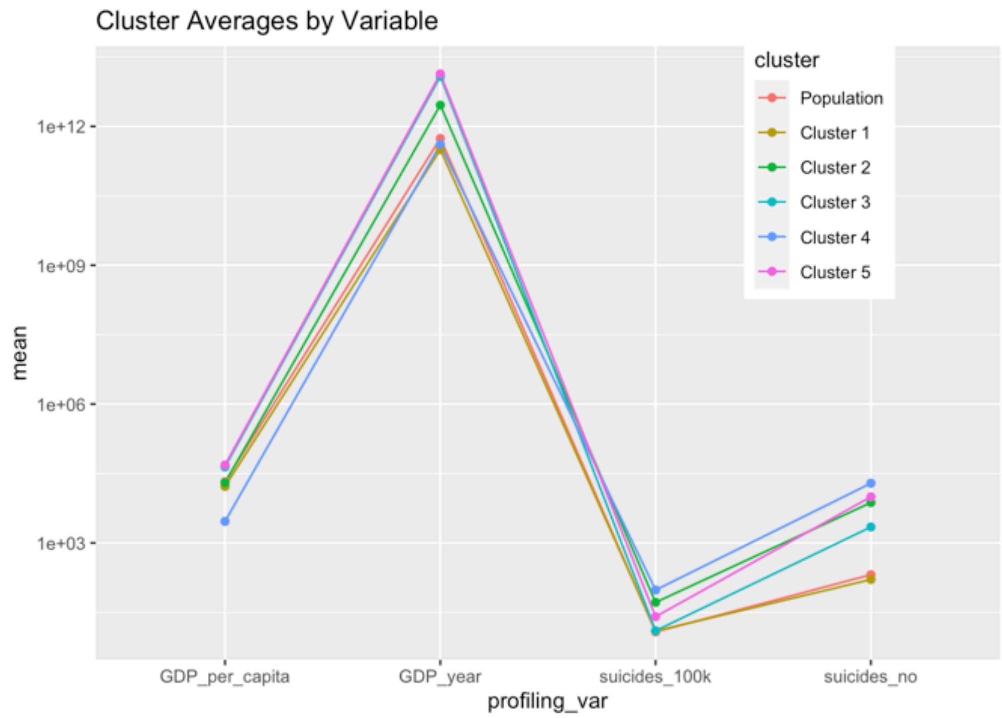
Showing 1 to 4 of 4 entries

Previous 1 Next

Snake Plot

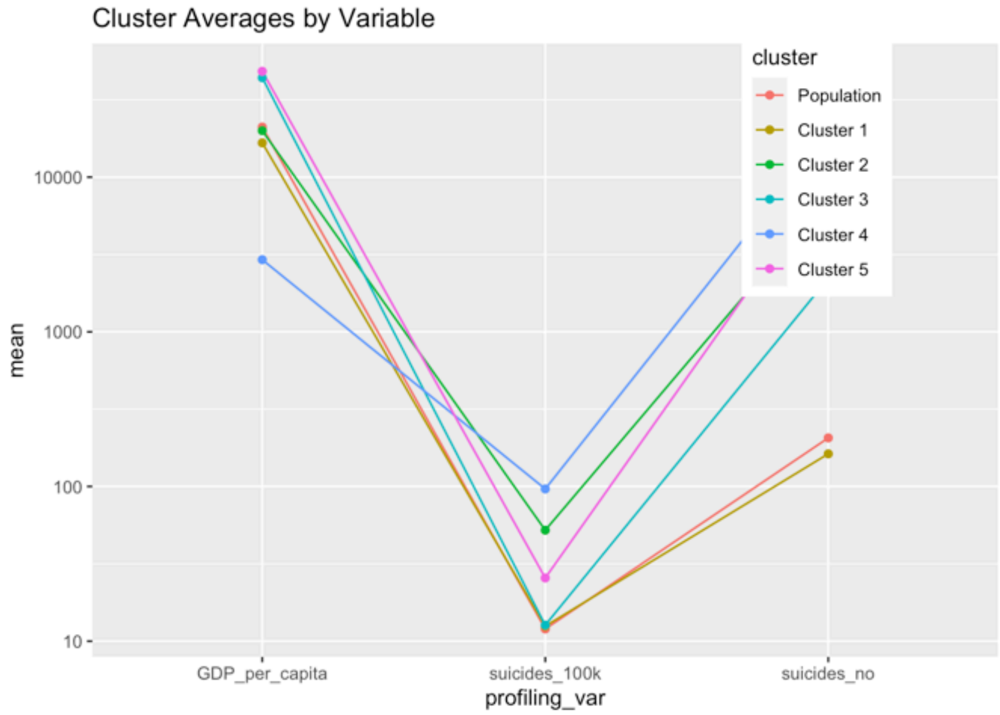
I will create a snake plot for visualisation of the different clusters

```
rate_sum_melt <- melt(summary, id = "profiling_var")
colnames(rate_sum_melt)[3] <- "mean"
colnames(rate_sum_melt)[2] <- "cluster"
rate_sum_melt <- tbl_df(rate_sum_melt)

snake_plot_rate <-
  rate_sum_melt %>% ggplot(aes(
    x = profiling_var,
    y = mean,
    group = cluster,
    color = cluster
  )) + geom_line() + geom_point() + scale_y_log10() + ggtitle("Cluster Averages by Variable") +
  theme(legend.position = c(0.8, 0.8))
snake_plot_rate
```



Omiting GDP per year to create a better visualisation

```
rate_sum_melt2 <- rate_sum_melt %>% filter(profiling_var !="GDP_year")
snake_plot_rate2 <- rate_sum_melt2 %>% ggplot(aes(
  x = profiling_var,
  y = mean,
  group = cluster,
  color = cluster
)) + geom_line() + geom_point() + scale_y_log10() + ggtitle("Cluster Averages by Variable") +
  theme(legend.position = c(0.8, 0.8))
snake_plot_rate2
```


Cluster Averages by Variable
cluster
Population
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
10000
1000
100
10
mean
GDP_per_capita
suicides_100k
suicides_no
profiling_var