

Team 5 Assignment 1

2022-08-06

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
SuicideRate <- read_excel("C:/Users/varni/Desktop/Course Documents/GBC Jan 2022/Semester 2/BUS Intro to
View(SuicideRate)
```

```
# 1. Print the structure of your dataset -
str(SuicideRate)
```

```
## tibble [27,820 x 12] (S3: 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" ...
```

```
# 2. List the variables in your dataset
names(SuicideRate)
```

```
##   [1] "country"      "year"          "sex"
##   [4] "age"          "suicides_no"   "population"
##   [7] "suicides/100k pop" "country-year"   "HDI for year"
##  [10] "gdp_for_year ($)" "gdp_per_capita ($)" "generation"
```

```
# 3. Print the top 15 rows of your dataset
head(SuicideRate, 15)
```

```
## # A tibble: 15 x 12
##   country year sex   age   suici~1 popul~2 suici~3 count~4 HDI f~5 gdp_f~6
##   <chr>   <dbl> <chr> <chr>   <dbl>   <dbl>   <dbl> <chr>   <dbl>   <dbl>
## 1 Albania 1987 male 15-24 y~    21 312900    6.71 Albani~    NA 2.16e9
```

```
## 2 Albania 1987 male 35-54 y~ 16 308000 5.19 Albani~ NA 2.16e9
## 3 Albania 1987 female 15-24 y~ 14 289700 4.83 Albani~ NA 2.16e9
## 4 Albania 1987 male 75+ yea~ 1 21800 4.59 Albani~ NA 2.16e9
## 5 Albania 1987 male 25-34 y~ 9 274300 3.28 Albani~ NA 2.16e9
## 6 Albania 1987 female 75+ yea~ 1 35600 2.81 Albani~ NA 2.16e9
## 7 Albania 1987 female 35-54 y~ 6 278800 2.15 Albani~ NA 2.16e9
## 8 Albania 1987 female 25-34 y~ 4 257200 1.56 Albani~ NA 2.16e9
## 9 Albania 1987 male 55-74 y~ 1 137500 0.73 Albani~ NA 2.16e9
## 10 Albania 1987 female 5-14 ye~ 0 311000 0 Albani~ NA 2.16e9
## 11 Albania 1987 female 55-74 y~ 0 144600 0 Albani~ NA 2.16e9
## 12 Albania 1987 male 5-14 ye~ 0 338200 0 Albani~ NA 2.16e9
## 13 Albania 1988 female 75+ yea~ 2 36400 5.49 Albani~ NA 2.13e9
## 14 Albania 1988 male 15-24 y~ 17 319200 5.33 Albani~ NA 2.13e9
## 15 Albania 1988 male 75+ yea~ 1 22300 4.48 Albani~ NA 2.13e9
## # ... with 2 more variables: `gdp_per_capita ($)` <dbl>, generation <chr>, and
## # abbreviated variable names 1: suicides_no, 2: population,
## # 3: `suicides/100k pop`, 4: `country-year`, 5: `HDI for year`,
## # 6: `gdp_for_year ($)`
## # i Use `colnames()` to see all variable names
```

4. Write a user defined function using any of the variables from the data set.

```
mean_row = function(newdata)
{
  result = mean(SuicideRate$suicides_no)
  return(result)
}
print(mean_row(SuicideRate))
```

```
## [1] 242.5744
```

5. Use data manipulation techniques and filter rows based on any logical criteria that exist in your
install.packages("dplyr")

```
## Warning: package 'dplyr' is in use and will not be installed
```

```
library(dplyr)
SuicideRatenew=select(SuicideRate,-"HDI for year")
SuicideRatenew = filter(SuicideRatenew, year >= 2000 & sex == "female")
head(SuicideRatenew, 15)
```

```
## # A tibble: 15 x 11
##   country year sex age suici~1 popul~2 suici~3 count~4 gdp_f~5 gdp_p~6
##   <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl> <dbl>
## 1 Albania 2000 female 75+ yea~ 2 37800 5.29 Albani~ 3.63e9 1299
## 2 Albania 2000 female 15-24 y~ 6 263900 2.27 Albani~ 3.63e9 1299
## 3 Albania 2000 female 35-54 y~ 5 332200 1.51 Albani~ 3.63e9 1299
## 4 Albania 2000 female 25-34 y~ 3 245800 1.22 Albani~ 3.63e9 1299
## 5 Albania 2000 female 5-14 ye~ 0 324700 0 Albani~ 3.63e9 1299
## 6 Albania 2000 female 55-74 y~ 0 168000 0 Albani~ 3.63e9 1299
## 7 Albania 2001 female 75+ yea~ 2 47254 4.23 Albani~ 4.06e9 1451
## 8 Albania 2001 female 15-24 y~ 9 271359 3.32 Albani~ 4.06e9 1451
## 9 Albania 2001 female 35-54 y~ 12 370191 3.24 Albani~ 4.06e9 1451
## 10 Albania 2001 female 55-74 y~ 6 189799 3.16 Albani~ 4.06e9 1451
## 11 Albania 2001 female 25-34 y~ 4 222771 1.8 Albani~ 4.06e9 1451
## 12 Albania 2001 female 5-14 ye~ 2 307356 0.65 Albani~ 4.06e9 1451
## 13 Albania 2002 female 15-24 y~ 14 275970 5.07 Albani~ 4.44e9 1573
```

```
## 14 Albania 2002 female 35-54 y~ 15 375113 4 Albani~ 4.44e9 1573
## 15 Albania 2002 female 25-34 y~ 7 223685 3.13 Albani~ 4.44e9 1573
## # ... with 1 more variable: generation <chr>, and abbreviated variable names
## # 1: suicides_no, 2: population, 3: `suicides/100k pop`, 4: `country-year`,
## # 5: `gdp_for_year ($)`, 6: `gdp_per_capita ($)`,
## # i Use `colnames()` to see all variable names
```

```
# 6. Identify the dependent & independent variables and use reshaping techniques and create a new data
colnames(SuicideRate) [9] = "gdp_year"
Dependent = select(SuicideRate, country, population, gdp_year)
head(Dependent, 15)
```

```
## # A tibble: 15 x 3
##   country population gdp_year
##   <chr>         <dbl>    <dbl>
## 1 Albania      312900      NA
## 2 Albania      308000      NA
## 3 Albania      289700      NA
## 4 Albania       21800      NA
## 5 Albania      274300      NA
## 6 Albania       35600      NA
## 7 Albania      278800      NA
## 8 Albania      257200      NA
## 9 Albania      137500      NA
## 10 Albania     311000      NA
## 11 Albania     144600      NA
## 12 Albania     338200      NA
## 13 Albania      36400      NA
## 14 Albania     319200      NA
## 15 Albania      22300      NA
```

```
# 7. Remove missing values in your dataset.
library(dplyr)
newSuicideRate = SuicideRate
newSuicideRate %>% filter(!is.na("HDI for year"))
```

```
## # A tibble: 27,820 x 12
##   country year sex age      suici~1 popul~2 suici~3 count~4 gdp_y~5 gdp_f~6
##   <chr>   <dbl> <chr> <chr>    <dbl>    <dbl>    <dbl> <chr>    <dbl>    <dbl>
## 1 Albania 1987 male 15-24 y~ 21 312900 6.71 Albani~ NA 2.16e9
## 2 Albania 1987 male 35-54 y~ 16 308000 5.19 Albani~ NA 2.16e9
## 3 Albania 1987 female 15-24 y~ 14 289700 4.83 Albani~ NA 2.16e9
## 4 Albania 1987 male 75+ yea~ 1 21800 4.59 Albani~ NA 2.16e9
## 5 Albania 1987 male 25-34 y~ 9 274300 3.28 Albani~ NA 2.16e9
## 6 Albania 1987 female 75+ yea~ 1 35600 2.81 Albani~ NA 2.16e9
## 7 Albania 1987 female 35-54 y~ 6 278800 2.15 Albani~ NA 2.16e9
## 8 Albania 1987 female 25-34 y~ 4 257200 1.56 Albani~ NA 2.16e9
## 9 Albania 1987 male 55-74 y~ 1 137500 0.73 Albani~ NA 2.16e9
## 10 Albania 1987 female 5-14 ye~ 0 311000 0 Albani~ NA 2.16e9
## # ... with 27,810 more rows, 2 more variables: `gdp_per_capita ($)`, <dbl>,
## # generation <chr>, and abbreviated variable names 1: suicides_no,
## # 2: population, 3: `suicides/100k pop`, 4: `country-year`, 5: gdp_year,
## # 6: `gdp_for_year ($)`,
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
print(newSuicideRate)
```

```
## # A tibble: 27,820 x 12
##   country year sex age      suici~1 popul~2 suici~3 count~4 gdp_y~5 gdp_f~6
##   <chr>   <dbl> <chr> <chr>      <dbl>   <dbl>   <dbl> <chr>      <dbl>   <dbl>
## 1 Albania 1987 male 15-24 y~      21  312900    6.71 Albani~      NA  2.16e9
## 2 Albania 1987 male 35-54 y~      16  308000    5.19 Albani~      NA  2.16e9
## 3 Albania 1987 female 15-24 y~      14  289700    4.83 Albani~      NA  2.16e9
## 4 Albania 1987 male 75+ yea~      1   21800    4.59 Albani~      NA  2.16e9
## 5 Albania 1987 male 25-34 y~      9  274300    3.28 Albani~      NA  2.16e9
## 6 Albania 1987 female 75+ yea~      1   35600    2.81 Albani~      NA  2.16e9
## 7 Albania 1987 female 35-54 y~      6  278800    2.15 Albani~      NA  2.16e9
## 8 Albania 1987 female 25-34 y~      4  257200    1.56 Albani~      NA  2.16e9
## 9 Albania 1987 male 55-74 y~      1  137500    0.73 Albani~      NA  2.16e9
## 10 Albania 1987 female 5-14 ye~      0  311000     0 Albani~      NA  2.16e9
## # ... with 27,810 more rows, 2 more variables: `gdp_per_capita ($)` <dbl>,
## #   generation <chr>, and abbreviated variable names 1: suicides_no,
## #   2: population, 3: `suicides/100k pop`, 4: `country-year`, 5: gdp_year,
## #   6: `gdp_for_year ($)`
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
# 8. Identify and remove duplicated data in your dataset
duplicated(SuicideRate)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## [27577] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27589] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27601] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27613] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27625] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27637] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27649] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27661] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27673] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27685] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27697] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27709] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27721] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27733] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27745] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27757] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27769] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27781] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27793] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27805] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [27817] FALSE FALSE FALSE FALSE
```

9. Reorder multiple rows in descending order

```
SuicideRate = SuicideRate %>% arrange(desc(country), desc(population))
head(SuicideRate, 15)
```

```
## # A tibble: 15 x 12
##   country    year sex    age    suici-1 popul-2 suici-3 count-4 gdp_y-5 gdp_f-6
##   <chr>      <dbl> <chr> <chr>    <dbl>    <dbl>    <dbl> <chr>    <dbl>    <dbl>
## 1 Uzbekistan 2014 female 35-5~    107 3620833    2.96 Uzbeki~ 0.675 6.31e10
## 2 Uzbekistan 2013 female 35-5~     99 3547895    2.79 Uzbeki~ 0.672 5.77e10
## 3 Uzbekistan 2012 female 35-5~     89 3474788    2.56 Uzbeki~ 0.668 5.18e10
## 4 Uzbekistan 2014 male   35-5~    519 3421300   15.2 Uzbeki~ 0.675 6.31e10
## 5 Uzbekistan 2011 female 35-5~     78 3404363    2.29 Uzbeki~ 0.661 4.59e10
## 6 Uzbekistan 2013 male   35-5~    481 3346411   14.4 Uzbeki~ 0.672 5.77e10
## 7 Uzbekistan 2010 female 35-5~    101 3339076    3.02 Uzbeki~ 0.655 3.93e10
## 8 Uzbekistan 2001 male   5-14~     64 3281449    1.95 Uzbeki~ NA      1.14e10
## 9 Uzbekistan 2000 male   5-14~     57 3279407    1.74 Uzbeki~ 0.594 1.38e10
## 10 Uzbekistan 2012 male   35-5~    404 3273281   12.3 Uzbeki~ 0.668 5.18e10
## 11 Uzbekistan 2009 female 35-5~     65 3263931    1.99 Uzbeki~ NA      3.37e10
## 12 Uzbekistan 2002 male   5-14~     49 3260824    1.5  Uzbeki~ NA      9.69e 9
## 13 Uzbekistan 1999 male   5-14~     59 3260451    1.81 Uzbeki~ NA      1.71e10
## 14 Uzbekistan 1998 male   5-14~     53 3222800    1.64 Uzbeki~ NA      1.50e10
## 15 Uzbekistan 2003 male   5-14~     45 3219722    1.4  Uzbeki~ NA      1.01e10
## # ... with 2 more variables: `gdp_per_capita ($)` <dbl>, generation <chr>, and
## # abbreviated variable names 1: suicides_no, 2: population,
## # 3: `suicides/100k pop`, 4: `country-year`, 5: gdp_year,
## # 6: `gdp_for_year ($)`
## # i Use `colnames()` to see all variable names
```

10. Rename some of the column names in your dataset

```
colnames(SuicideRate) [9] = "gdp_year"
names(SuicideRate)
```

```
## [1] "country"      "year"          "sex"
## [4] "age"          "suicides_no"  "population"
```

```
## [7] "suicides/100k pop" "country-year" "gdp_year"
## [10] "gdp_for_year ($)" "gdp_per_capita ($)" "generation"

# 11. Add new variables in your data frame by using a mathematical function (for e.g. - multiply an exi
SuicideRate$double_suicide <- SuicideRate$suicides_no*2
print(SuicideRate)
```

```
## # A tibble: 27,820 x 13
##   country    year sex    age  suici~1 popul~2 suici~3 count~4 gdp_y~5 gdp_f~6
##   <chr>      <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl> <dbl>
## 1 Uzbekistan 2014 female 35-5~ 107 3620833 2.96 Uzbeki~ 0.675 6.31e10
## 2 Uzbekistan 2013 female 35-5~ 99 3547895 2.79 Uzbeki~ 0.672 5.77e10
## 3 Uzbekistan 2012 female 35-5~ 89 3474788 2.56 Uzbeki~ 0.668 5.18e10
## 4 Uzbekistan 2014 male 35-5~ 519 3421300 15.2 Uzbeki~ 0.675 6.31e10
## 5 Uzbekistan 2011 female 35-5~ 78 3404363 2.29 Uzbeki~ 0.661 4.59e10
## 6 Uzbekistan 2013 male 35-5~ 481 3346411 14.4 Uzbeki~ 0.672 5.77e10
## 7 Uzbekistan 2010 female 35-5~ 101 3339076 3.02 Uzbeki~ 0.655 3.93e10
## 8 Uzbekistan 2001 male 5-14~ 64 3281449 1.95 Uzbeki~ NA 1.14e10
## 9 Uzbekistan 2000 male 5-14~ 57 3279407 1.74 Uzbeki~ 0.594 1.38e10
## 10 Uzbekistan 2012 male 35-5~ 404 3273281 12.3 Uzbeki~ 0.668 5.18e10
## # ... with 27,810 more rows, 3 more variables: `gdp_per_capita ($)` <dbl>,
## # generation <chr>, double_suicide <dbl>, and abbreviated variable names
## # 1: suicides_no, 2: population, 3: `suicides/100k pop`, 4: `country-year`,
## # 5: gdp_year, 6: `gdp_for_year ($)`
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
# 12. Create a training set using random number generator engine
head(SuicideRate)
```

```
## # A tibble: 6 x 13
##   country    year sex    age  suici~1 popul~2 suici~3 count~4 gdp_y~5 gdp_f~6
##   <chr>      <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl> <dbl>
## 1 Uzbekistan 2014 female 35-54~ 107 3620833 2.96 Uzbeki~ 0.675 6.31e10
## 2 Uzbekistan 2013 female 35-54~ 99 3547895 2.79 Uzbeki~ 0.672 5.77e10
## 3 Uzbekistan 2012 female 35-54~ 89 3474788 2.56 Uzbeki~ 0.668 5.18e10
## 4 Uzbekistan 2014 male 35-54~ 519 3421300 15.2 Uzbeki~ 0.675 6.31e10
## 5 Uzbekistan 2011 female 35-54~ 78 3404363 2.29 Uzbeki~ 0.661 4.59e10
## 6 Uzbekistan 2013 male 35-54~ 481 3346411 14.4 Uzbeki~ 0.672 5.77e10
## # ... with 3 more variables: `gdp_per_capita ($)` <dbl>, generation <chr>,
## # double_suicide <dbl>, and abbreviated variable names 1: suicides_no,
## # 2: population, 3: `suicides/100k pop`, 4: `country-year`, 5: gdp_year,
## # 6: `gdp_for_year ($)`
## # i Use `colnames()` to see all variable names
```

```
h <-runif(nrow(SuicideRate))
rate<-SuicideRate[order(h), ]
str(rate)
```

```
## tibble [27,820 x 13] (S3: tbl_df/tbl/data.frame)
## $ country      : chr [1:27820] "Poland" "Hungary" "Colombia" "Brazil" ...
## $ year         : num [1:27820] 2009 2010 1987 1985 2003 ...
## $ sex          : chr [1:27820] "female" "female" "female" "male" ...
## $ age          : chr [1:27820] "15-24 years" "25-34 years" "5-14 years" "35-54 years" ...
## $ suicides_no  : num [1:27820] 75 37 9 1011 191 ...
## $ population   : num [1:27820] 2740576 748440 3796166 11986300 2795116 ...
## $ suicides/100k pop : num [1:27820] 2.74 4.94 0.24 8.43 6.83 ...
```



```
## $ country-year      : chr [1:27820] "Poland2009" "Hungary2010" "Colombia1987" "Brazil1985" ...
## $ gdp_year          : num [1:27820] NA 0.821 NA 0.576 NA NA NA 0.874 NA NA ...
## $ gdp_for_year ($)  : num [1:27820] 4.40e+11 1.31e+11 3.64e+10 2.23e+11 9.07e+11 ...
## $ gdp_per_capita ($) : num [1:27820] 12141 13761 1299 1898 22701 ...
## $ generation        : chr [1:27820] "Millenials" "Generation X" "Generation X" "Silent" ...
## $ double_suicide    : num [1:27820] 150 74 18 2022 382 ...
```

```
train <- rate[1:36, ]
SuicideRateclean = na.omit(SuicideRate)
set.seed(1234)
SuicideRateTraining = as.data.frame(SuicideRateclean %>% sample_frac(0.75,replace = FALSE))
```

```
# 13. Print the summary statistics of your dataset
summary(SuicideRate)
```

```
##      country          year          sex          age
## Length:27820      Min.   :1985  Length:27820      Length:27820
## Class :character  1st Qu.:1995  Class :character  Class :character
## Mode  :character  Median :2002  Mode  :character  Mode  :character
##                               Mean  :2001
##                               3rd Qu.:2008
##                               Max.   :2016
##
## suicides_no      population      suicides/100k pop country-year
## Min.   :    0.0  Min.   :    278  Min.   :    0.00  Length:27820
## 1st Qu.:    3.0  1st Qu.:   97498  1st Qu.:    0.92  Class :character
## Median :   25.0  Median :  430150  Median :    5.99  Mode  :character
## Mean   :  242.6  Mean   :1844794  Mean   :   12.82
## 3rd Qu.:  131.0  3rd Qu.:1486143  3rd Qu.:   16.62
## Max.   :22338.0  Max.   :43805214  Max.   :   224.97
##
##      gdp_year      gdp_for_year ($)      gdp_per_capita ($)      generation
## Min.   :0.483  Min.   :4.692e+07  Min.   :    251  Length:27820
## 1st Qu.:0.713  1st Qu.:8.985e+09  1st Qu.:   3447  Class :character
## Median :0.779  Median :4.811e+10  Median :   9372  Mode  :character
## Mean   :0.777  Mean   :4.456e+11  Mean   :  16866
## 3rd Qu.:0.855  3rd Qu.:2.602e+11  3rd Qu.: 24874
## Max.   :0.944  Max.   :1.812e+13  Max.   :126352
## NA's   :19456
## double_suicide
## Min.   :    0.0
## 1st Qu.:    6.0
## Median :   50.0
## Mean   :  485.1
## 3rd Qu.:  262.0
## Max.   :44676.0
##
```

```
# 14. Use any of the numerical variables from the dataset and perform the following statistical functions
# 14.1 Mean
```

```
summary(SuicideRate$population)
```

```
##      Min.  1st Qu.  Median    Mean  3rd Qu.    Max.
##      278    97498   430150 1844794 1486143 43805214
```

14.2 Median

```
summary(SuicideRate$population)
```

```
##      Min.   1st Qu.   Median     Mean  3rd Qu.     Max.
##      278     97498    430150   1844794  1486143  43805214
```

14.3 Mode

```
vec1 <- SuicideRate$population
getmode <- function(v)
{
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
result <- getmode(vec1)
print(result)
```

```
## [1] 24000
```

14.4 Range

```
range(SuicideRate$population)
```

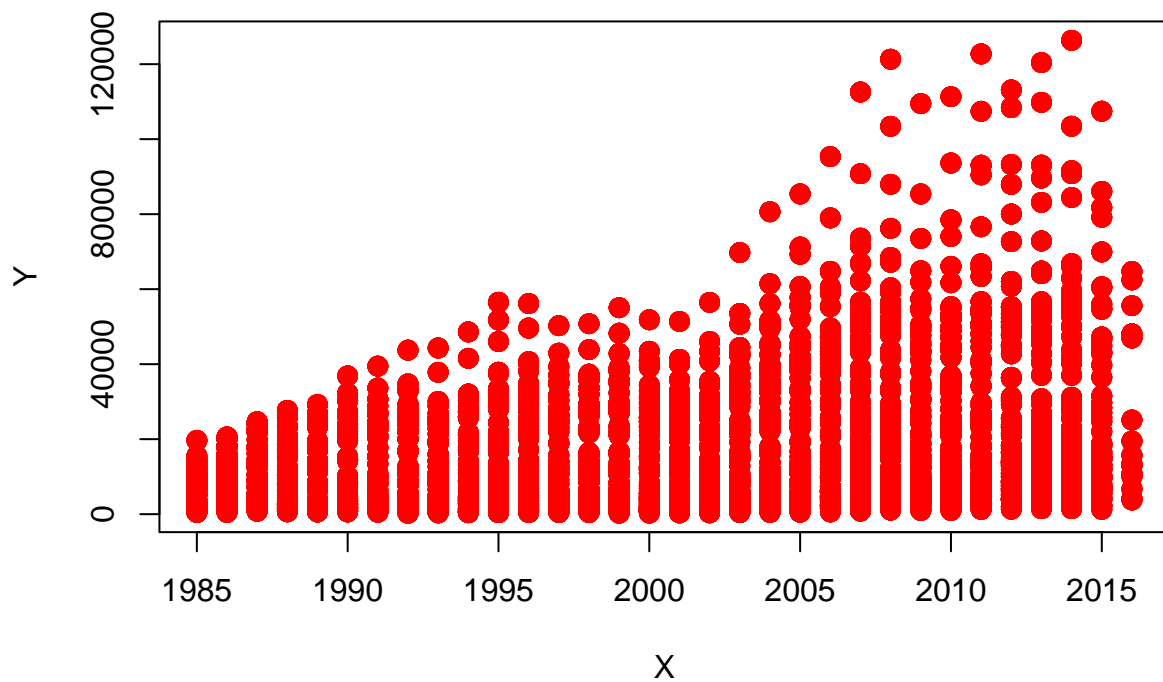
```
## [1]      278 43805214
```

15. Plot a scatter plot for any 2 variables in your dataset

```
library(ggplot2)
```

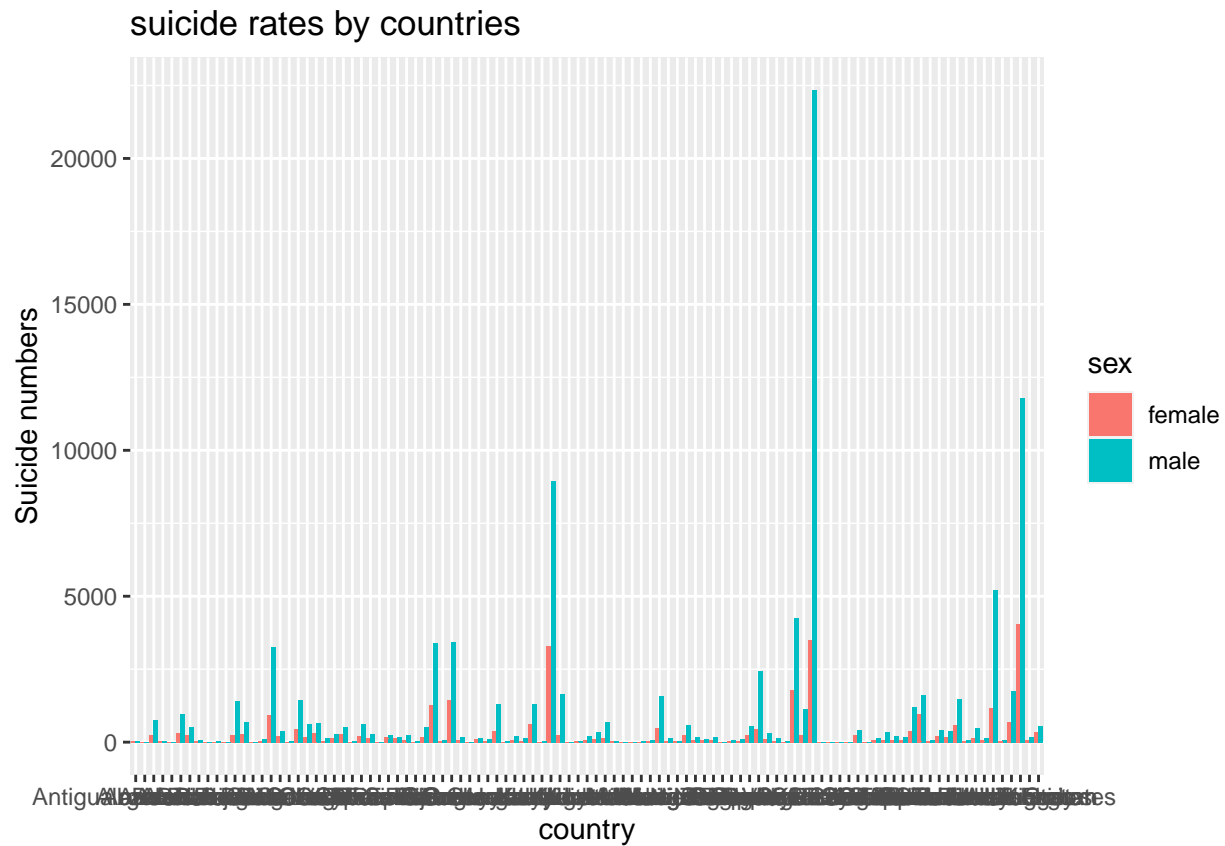
```
plot(SuicideRate$year, SuicideRate$`gdp_per_capita ($)` , col='red', pch=19, cex=1.3, xlab='X', ylab='Y')
```

Scatterplot of Two Variables



16. Plot a bar plot for any 2 variables in your dataset

```
ggplot(SuicideRate, aes(fill=sex, y=suicides_no, x=country)) + geom_bar(position='dodge', stat='identity')
```



17. Find the correlation between any 2 variables by applying least square linear regression model

```
Y<- SuicideRate[, "population"]
X<- SuicideRate[, "suicides_no"]
head(X)
```

```
## # A tibble: 6 x 1
##   suicides_no
##   <dbl>
## 1      107
## 2       99
## 3       89
## 4      519
## 5       78
## 6      481
```

```
head(Y)
```

```
## # A tibble: 6 x 1
##   population
##   <dbl>
## 1  3620833
## 2  3547895
## 3  3474788
## 4  3421300
```

```
## 5    3404363
## 6    3346411
xycorr <- cor(Y,X, method="pearson")
```