Introduction

The World Happiness Report is a yearly survey of the state of global happiness. The report are used global from governments and organization to use happiness indicators to help with policy-making decisions.

```
 # installs the necessary libraries
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
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v readr   1.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(readr)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following object is masked from 'package:purrr':
##
##     transpose

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(corrplot)
```

```
## corrplot 0.89 loaded
```

```
library(corrgram)
library(forcats)
```

Overview of the Data

The data that used here is the survery from 2019. They are 9 variables in this survery. 9 Varables are as follows

Overall rank: The over all rank in which country has been ranked.

Country or region: The name of the country in the survery.

Score: The score that each country has been given in the survery.

GDP per capita: Per capita gross domestic product (GDP) is the country's economic output per person.

Social Support: Social support is rated by how munch a citizen of the country gets support from family and friends.

Healthy life expectancy: The rating of health and life expectancy of the country.

Freedom to make life choices: The rating of the country's citizen to make their own choices in life.

Generosity: The rating of the country's citizen of givng their time to others.

Perceptions of corruption: The country's perceived levels of public sector corruption.

```
# reads the 2019 csv file and shows the first 5 rows of the data set
data <- read.csv("2019.csv")
head(data)
```

```
##   Overall.rank Country.or.region Score GDP.per.capita Social.support
## 1            1           Finland 7.769          1.340          1.587
## 2            2           Denmark 7.600          1.383          1.573
## 3            3            Norway 7.554          1.488          1.582
## 4            4           Iceland 7.494          1.380          1.624
## 5            5       Netherlands 7.488          1.396          1.522
## 6            6       Switzerland 7.480          1.452          1.526
##   Healthy.life.expectancy Freedom.to.make.life.choices Generosity
## 1                   0.986                        0.596      0.153
## 2                   0.996                        0.592      0.252
## 3                   1.028                        0.603      0.271
## 4                   1.026                        0.591      0.354
## 5                   0.999                        0.557      0.322
## 6                   1.052                        0.572      0.263
##   Perceptions.of.corruption
## 1                     0.393
## 2                     0.410
## 3                     0.341
## 4                     0.118
## 5                     0.298
## 6                     0.343
```

Structure of the Data

Below will show the structure of the data set. The output shows there are 156 columns and 9 rows.

```
# shows the structure of the data set
str(data)
```

```
## 'data.frame':    156 obs. of  9 variables:
##  $ Overall.rank                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Country.or.region           : chr  "Finland" "Denmark" "Norway" "Iceland" ...
##  $ Score                       : num  7.77 7.6 7.55 7.49 7.49 ...
##  $ GDP.per.capita              : num  1.34 1.38 1.49 1.38 1.4 ...
##  $ Social.support              : num  1.59 1.57 1.58 1.62 1.52 ...
##  $ Healthy.life.expectancy     : num  0.986 0.996 1.028 1.026 0.999 ...
##  $ Freedom.to.make.life.choices: num  0.596 0.592 0.603 0.591 0.557 0.572 0.574 0.585 0.584 0.532 ..
##  $ Generosity                  : num  0.153 0.252 0.271 0.354 0.322 0.263 0.267 0.33 0.285 0.244 ...
##  $ Perceptions.of.corruption   : num  0.393 0.41 0.341 0.118 0.298 0.343 0.373 0.38 0.308 0.226 ...
```

Summary of the Data set

Below will shows the summary of the data. For the numeric variables it show Min, 1st Quarter, Median, Mean, 3rd Quarter, and Max. For Country.or.region it shows that it is a character class.

```
# shows the summary of the data set
summary(data)
```

```
##   Overall.rank    Country.or.region      Score       GDP.per.capita
##  Min.   :  1.00   Length:156         Min.   :2.853   Min.   :0.0000
##  1st Qu.: 39.75   Class :character   1st Qu.:4.545   1st Qu.:0.6028
##  Median : 78.50   Mode  :character   Median :5.380   Median :0.9600
##  Mean   : 78.50                      Mean   :5.407   Mean   :0.9051
##  3rd Qu.:117.25                      3rd Qu.:6.184   3rd Qu.:1.2325
##  Max.   :156.00                      Max.   :7.769   Max.   :1.6840
##  Social.support  Healthy.life.expectancy Freedom.to.make.life.choices
##  Min.   :0.000   Min.   :0.0000          Min.   :0.0000
##  1st Qu.:1.056   1st Qu.:0.5477          1st Qu.:0.3080
##  Median :1.272   Median :0.7890          Median :0.4170
##  Mean   :1.209   Mean   :0.7252          Mean   :0.3926
##  3rd Qu.:1.452   3rd Qu.:0.8818          3rd Qu.:0.5072
##  Max.   :1.624   Max.   :1.1410          Max.   :0.6310
##    Generosity     Perceptions.of.corruption
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.1087   1st Qu.:0.0470
##  Median :0.1775   Median :0.0855
##  Mean   :0.1848   Mean   :0.1106
##  3rd Qu.:0.2482   3rd Qu.:0.1412
##  Max.   :0.5660   Max.   :0.4530
```

****Renaming the columns.**

Here I renamed the columns for efficient use through out the report when calling them.

```
# renames the columns in the data set and shows the first 5 rows with the change of column names
data <- data %>% rename(country = Country.or.region, gdp = GDP.per.capita, support=Social.support,life_

head(data)
```

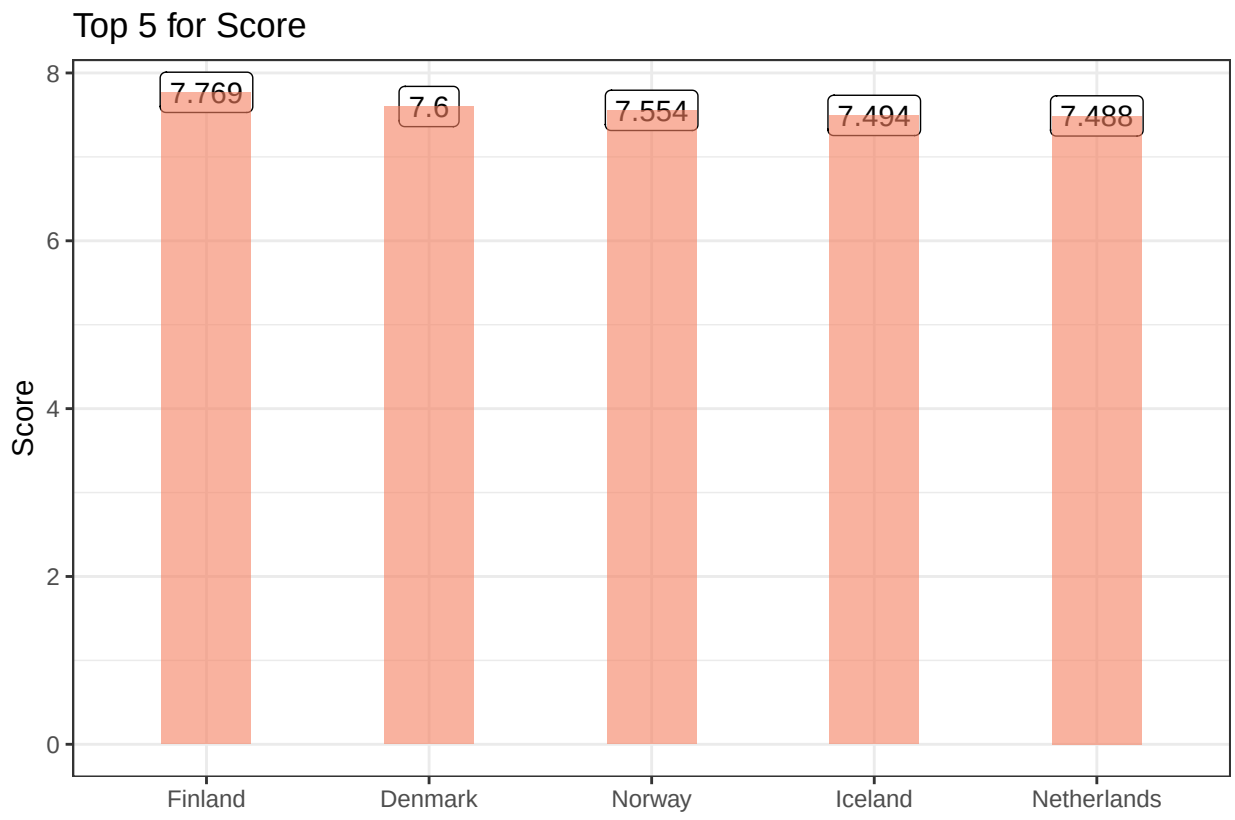
```
##   rank     country Score   gdp support life_expectancy make_choices Generosity
## 1    1     Finland 7.769 1.340   1.587           0.986        0.596      0.153
## 2    2     Denmark 7.600 1.383   1.573           0.996        0.592      0.252
## 3    3      Norway 7.554 1.488   1.582           1.028        0.603      0.271
## 4    4     Iceland 7.494 1.380   1.624           1.026        0.591      0.354
## 5    5 Netherlands 7.488 1.396   1.522           0.999        0.557      0.322
## 6    6 Switzerland 7.480 1.452   1.526           1.052        0.572      0.263
##   corruption
## 1      0.393
## 2      0.410
## 3      0.341
## 4      0.118
## 5      0.298
## 6      0.343
```

Graph for Top 5 countries with the highest Scores

Below will output a bar chart of the top 5 countries with the highest score

```
# outputs bar chart of the top 5 countries with the highest scores
top_n(data,n=5,Score)  %>% mutate(country = fct_reorder(country, desc(Score))) %>%
  ggplot( aes(x=country, y= Score)) + geom_label(aes(label = Score))+
    geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +

    xlab("") +
    theme_bw()+labs(title="Top 5 for Score",y="Score")
```

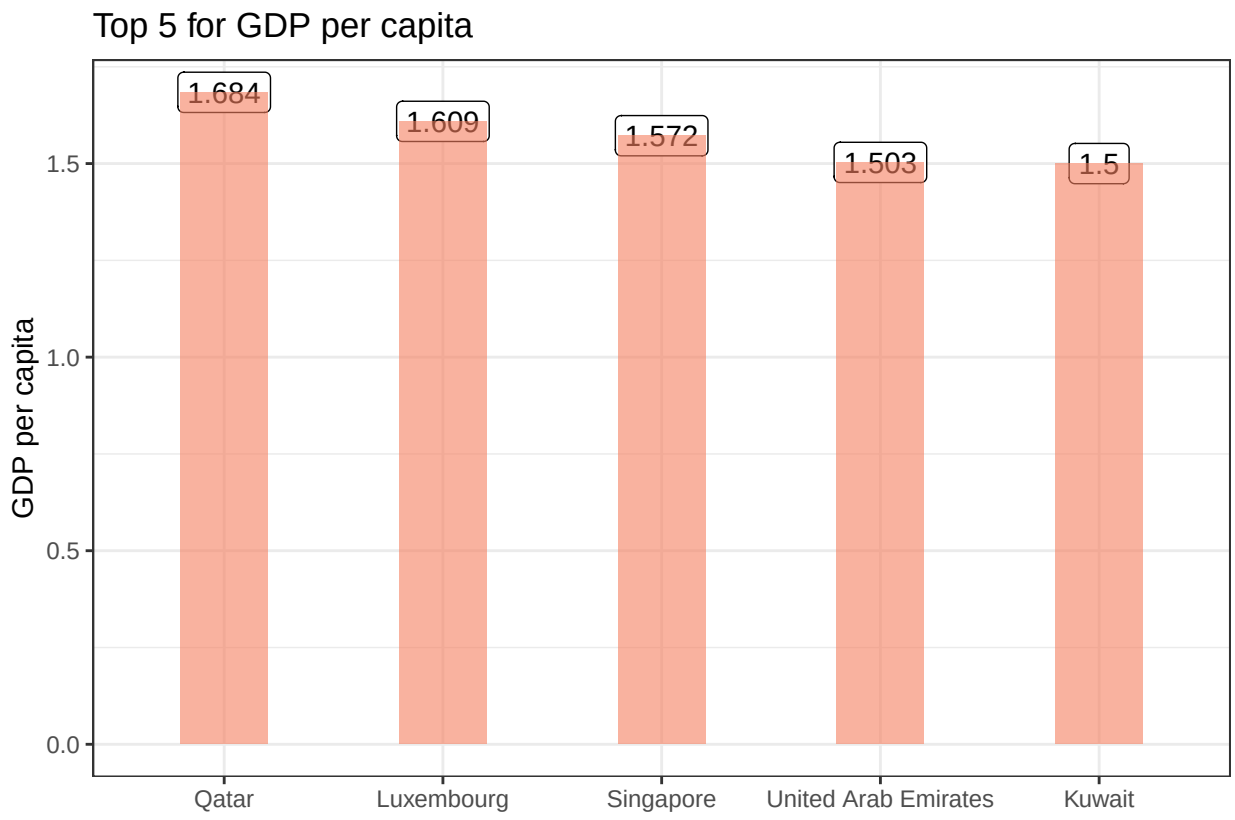


**** Graph for Top 5 countires with the highest GDP per Capita****

Below will output a bar chart of the top 5 countries with the GDP per Capita

```
# outputs the top 5 countries with highest GDP per capita
top_n(data,n=5, gdp) %>%mutate(country = fct_reorder(country, desc(gdp))) %>%
  ggplot( aes(x=country, y=gdp)) + geom_label(aes(label = gdp))+
    geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +

    xlab("") +
    theme_bw()+labs(title="Top 5 for GDP per capita ", y="GDP per capita")
```

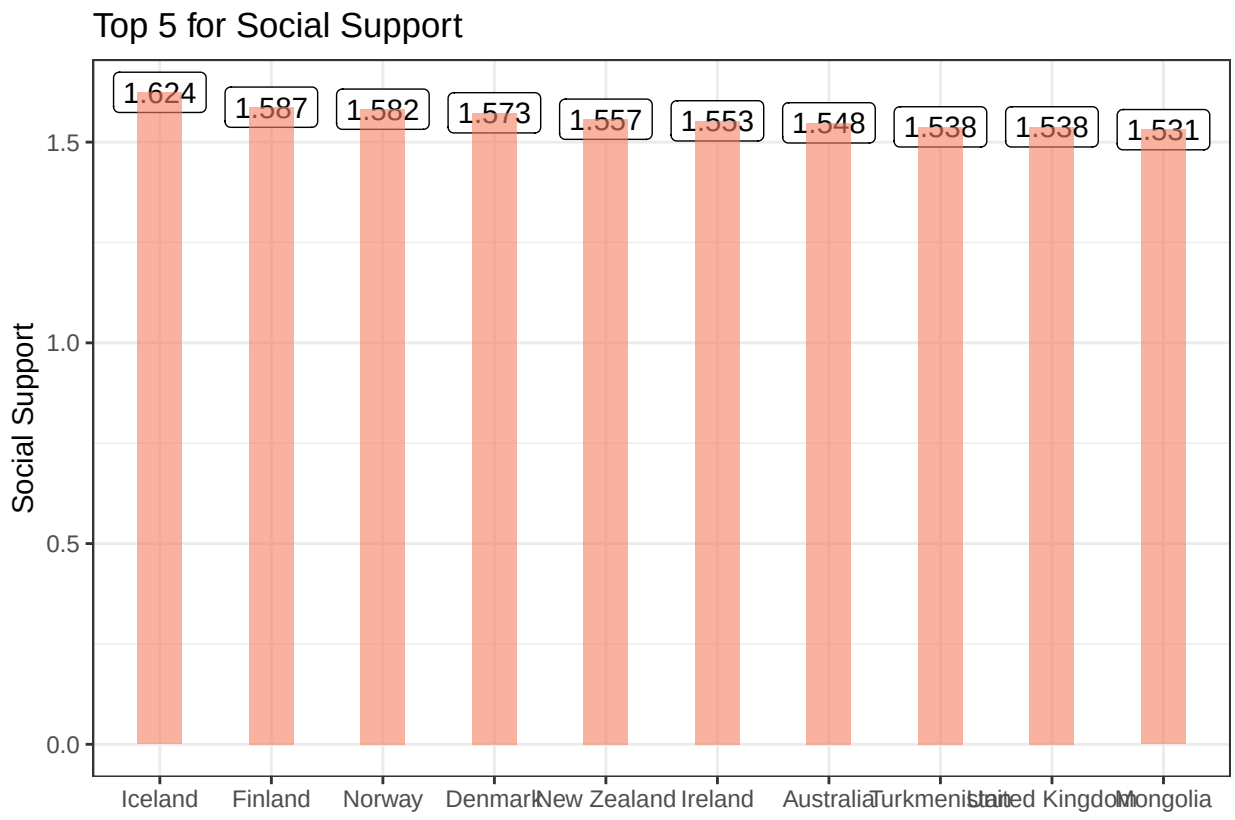


**** Graph for Top 5 countires with the highest Social Support ****

Below will output a bar chart of the top 5 countries with the Social Support

```
# outputs the top 5 countries with the highest Social Support
top_n(data,n=10, support) %>%mutate(country = fct_reorder(country, desc(support))) %>%
  ggplot( aes(x=country, y=support)) + geom_label(aes(label = support))+ geom_bar(stat="identity", fill=

    xlab("") +
    theme_bw() + labs(title="Top 5 for Social Support",y="Social Support")
```

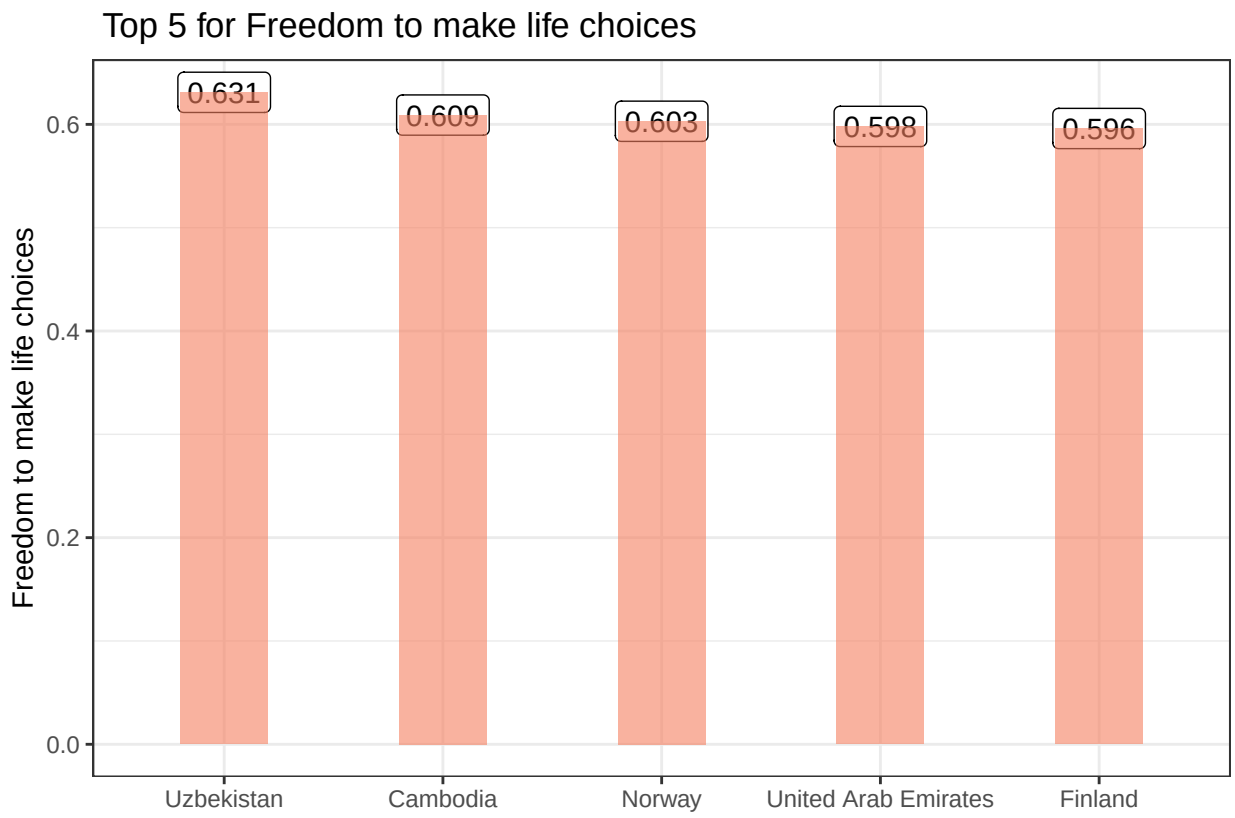


**** Graph for Top 5 countires with the highest Freedom to make life choices ****

Below will output a bar chart of the top 5 countries with the Freedom to make life choices

```
# outputs the top 5 countries with the highest Freedom to make life choices
top_n(data,n=5,make_choices)  %>% mutate(country = fct_reorder(country, desc(make_choices))) %>%
  ggplot( aes(x=country, y = make_choices)) + geom_label(aes(label = make_choices))+
    geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +

    xlab("") +
    theme_bw() + labs(title=" Top 5 for Freedom to make life choices ",y="Freedom to make life choices")
```

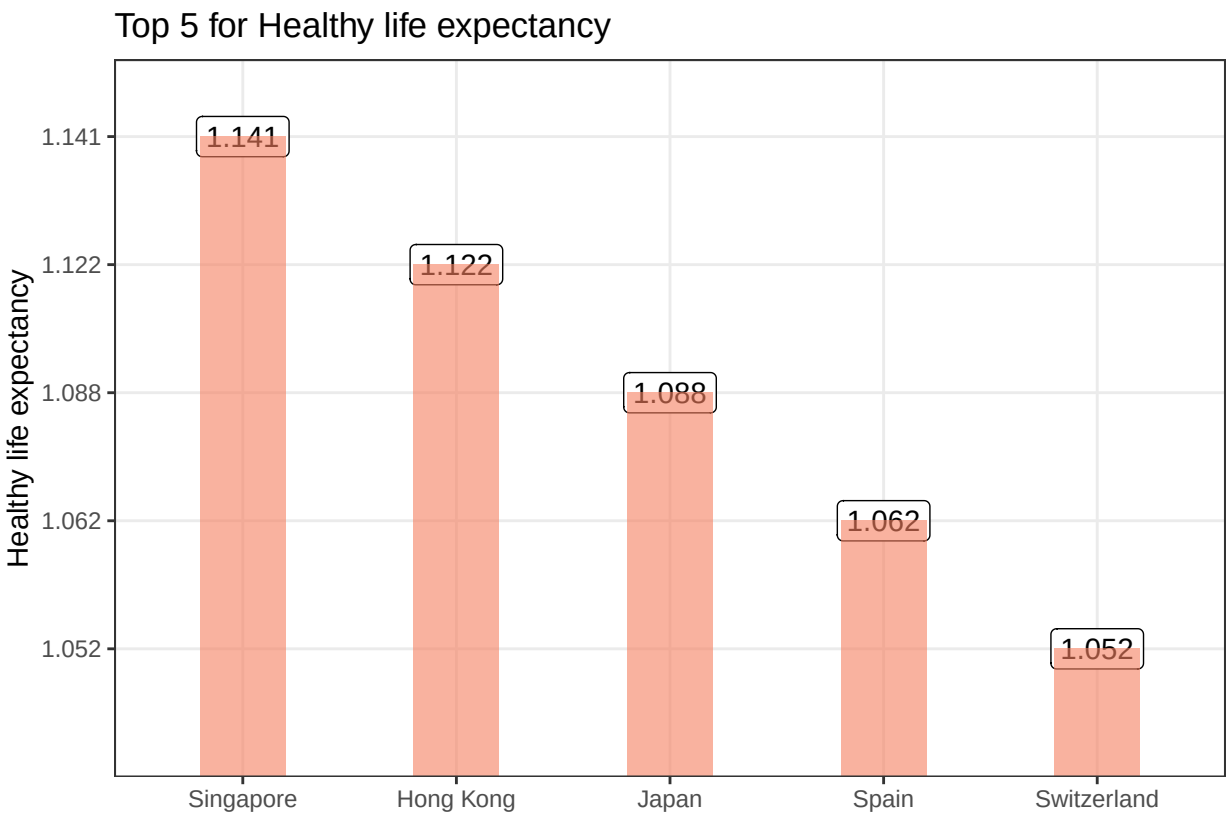


**** Graph for Top 5 countires with the highest Healthy life expectancy ****

Below will output a bar chart of the top 5 countries with the Healthy life expectancy

```
# outputs the top 5 countries with the highest Healthy life expectancy
top_n(data,n=5,life_expectancy) %>% mutate(country = fct_reorder(country, desc(life_expectancy))) %>%
  ggplot( aes(x= country, y= as.character(life_expectancy))) + geom_label(aes(label = life_expectancy))+
    geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +

    xlab("") +
    theme_bw() + labs(title="Top 5 for Healthy life expectancy", y="Healthy life expectancy")
```

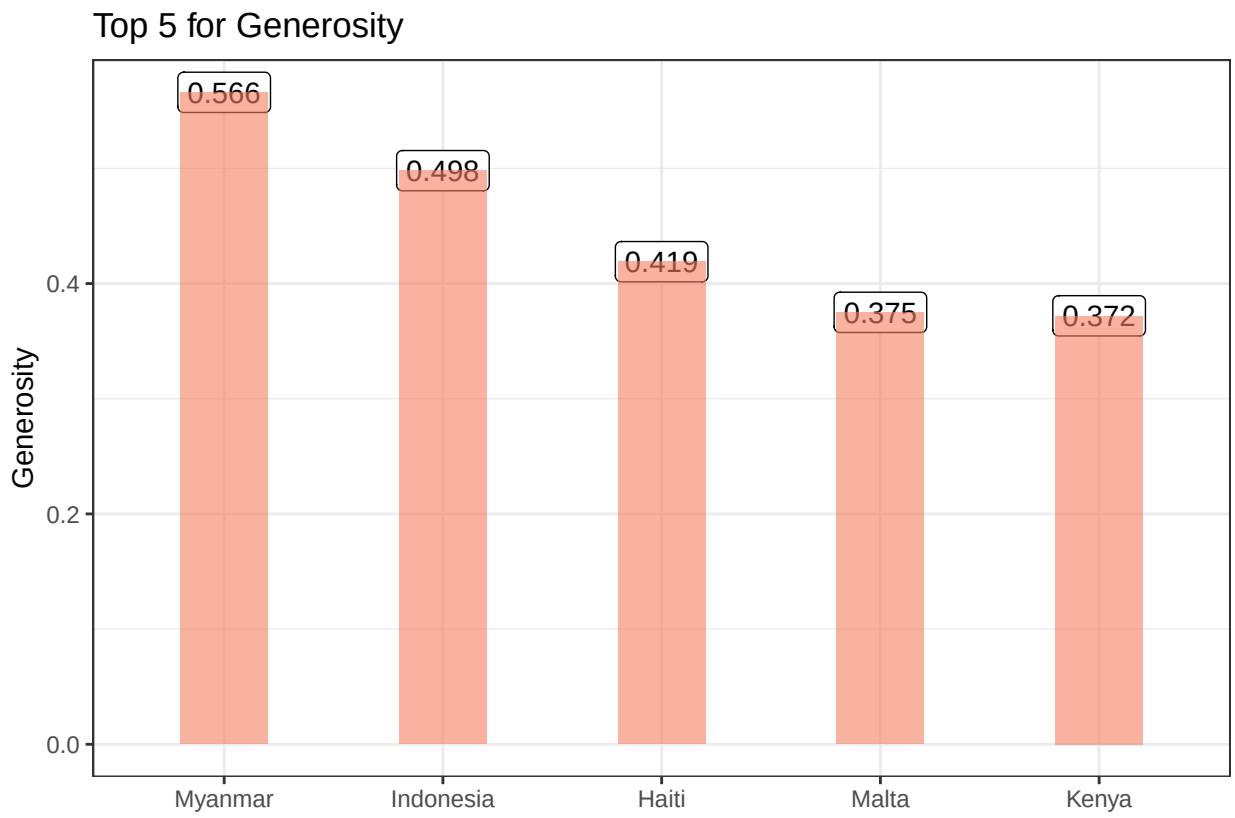


**** Graph for Top 5 countires with the highest Generosity ****

Below will output a bar chart of the top 5 countries with the Generosity

```
# outputs the top 5 countries with the highest Generosity
top_n(data,n=5,Generosity) %>% mutate(country = fct_reorder(country, desc(Generosity))) %>%
  ggplot( aes(x=country, y=Generosity)) +  geom_label(aes(label = Generosity))+
    geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +

    xlab("") +
    theme_bw() + labs(title="Top 5 for Generosity",y="Generosity")
```

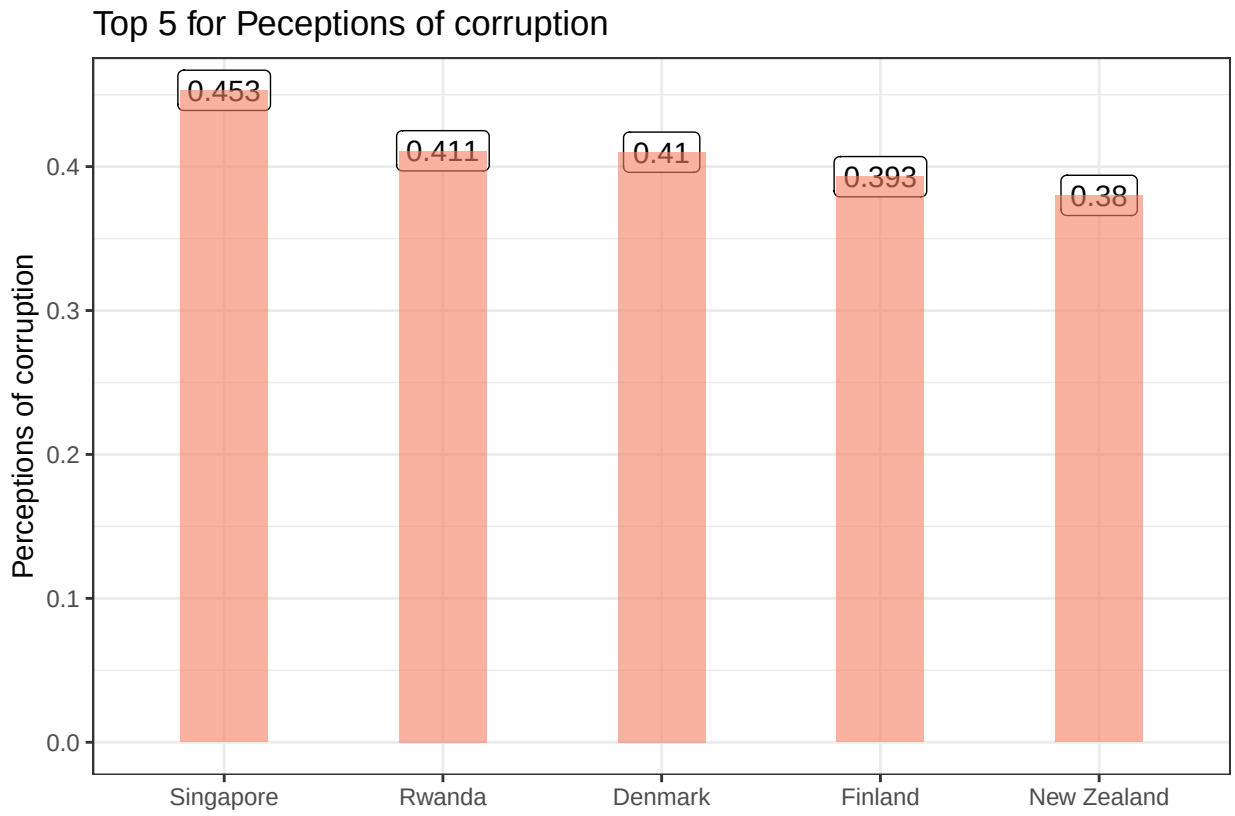


**** Graph for Top 5 countires with the highest Generosity ****

Below will output a bar chart of the top 5 countries with the Generosity

```
# outputs the top 5 countries with the Perceptions of corruption
top_n(data,n=5,corruption) %>% mutate(country = fct_reorder(country, desc(corruption))) %>%
  ggplot( aes(x=country, y=corruption)) + geom_label(aes(label = corruption))+
    geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +

    xlab("") +
    theme_bw() + labs(title="Top 5 for Peceptions of corruption", y="Perceptions of corruption")
```



Droping Columns

Here I droped the columns country and rank to only have columns with numeric values and put them into the variable num_data

```
# drops rows that do not contain numeric data
num_data <- data %>% subset(select=-c(country,rank))
num_data
```

```
##     Score   gdp support life_expectancy make_choices Generosity corruption
## 1   7.769 1.340   1.587           0.986        0.596      0.153      0.393
## 2   7.600 1.383   1.573           0.996        0.592      0.252      0.410
## 3   7.554 1.488   1.582           1.028        0.603      0.271      0.341
## 4   7.494 1.380   1.624           1.026        0.591      0.354      0.118
## 5   7.488 1.396   1.522           0.999        0.557      0.322      0.298
## 6   7.480 1.452   1.526           1.052        0.572      0.263      0.343
## 7   7.343 1.387   1.487           1.009        0.574      0.267      0.373
## 8   7.307 1.303   1.557           1.026        0.585      0.330      0.380
## 9   7.278 1.365   1.505           1.039        0.584      0.285      0.308
## 10  7.246 1.376   1.475           1.016        0.532      0.244      0.226
## 11  7.228 1.372   1.548           1.036        0.557      0.332      0.290
## 12  7.167 1.034   1.441           0.963        0.558      0.144      0.093
## 13  7.139 1.276   1.455           1.029        0.371      0.261      0.082
## 14  7.090 1.609   1.479           1.012        0.526      0.194      0.316
## 15  7.054 1.333   1.538           0.996        0.450      0.348      0.278
```

```
## 16  7.021 1.499   1.553          0.999        0.516      0.298      0.310
## 17  6.985 1.373   1.454          0.987        0.495      0.261      0.265
## 18  6.923 1.356   1.504          0.986        0.473      0.160      0.210
## 19  6.892 1.433   1.457          0.874        0.454      0.280      0.128
## 20  6.852 1.269   1.487          0.920        0.457      0.046      0.036
## 21  6.825 1.503   1.310          0.825        0.598      0.262      0.182
## 22  6.726 1.300   1.520          0.999        0.564      0.375      0.151
## 23  6.595 1.070   1.323          0.861        0.433      0.074      0.073
## 24  6.592 1.324   1.472          1.045        0.436      0.111      0.183
## 25  6.446 1.368   1.430          0.914        0.351      0.242      0.097
## 26  6.444 1.159   1.369          0.920        0.357      0.187      0.056
## 27  6.436 0.800   1.269          0.746        0.535      0.175      0.078
## 28  6.375 1.403   1.357          0.795        0.439      0.080      0.132
## 29  6.374 1.684   1.313          0.871        0.555      0.220      0.167
## 30  6.354 1.286   1.484          1.062        0.362      0.153      0.079
## 31  6.321 1.149   1.442          0.910        0.516      0.109      0.054
## 32  6.300 1.004   1.439          0.802        0.390      0.099      0.086
## 33  6.293 1.124   1.465          0.891        0.523      0.127      0.150
## 34  6.262 1.572   1.463          1.141        0.556      0.271      0.453
## 35  6.253 0.794   1.242          0.789        0.430      0.093      0.074
## 36  6.223 1.294   1.488          1.039        0.231      0.158      0.030
## 37  6.199 1.362   1.368          0.871        0.536      0.255      0.110
## 38  6.198 1.246   1.504          0.881        0.334      0.121      0.014
## 39  6.192 1.231   1.477          0.713        0.489      0.185      0.016
## 40  6.182 1.206   1.438          0.884        0.483      0.117      0.050
## 41  6.174 0.745   1.529          0.756        0.631      0.322      0.240
## 42  6.149 1.238   1.515          0.818        0.291      0.043      0.042
## 43  6.125 0.985   1.410          0.841        0.470      0.099      0.034
## 44  6.118 1.258   1.523          0.953        0.564      0.144      0.057
## 45  6.105 0.694   1.325          0.835        0.435      0.200      0.127
## 46  6.100 0.882   1.232          0.758        0.489      0.262      0.006
## 47  6.086 1.092   1.432          0.881        0.471      0.066      0.050
## 48  6.070 1.162   1.232          0.825        0.462      0.083      0.005
## 49  6.046 1.263   1.223          1.042        0.406      0.190      0.041
## 50  6.028 0.912   1.312          0.868        0.498      0.126      0.087
## 51  6.021 1.500   1.319          0.808        0.493      0.142      0.097
## 52  6.008 1.050   1.409          0.828        0.557      0.359      0.028
## 53  5.940 1.187   1.465          0.812        0.264      0.075      0.064
## 54  5.895 1.301   1.219          1.036        0.159      0.175      0.056
## 55  5.893 1.237   1.528          0.874        0.495      0.103      0.161
## 56  5.890 0.831   1.478          0.831        0.490      0.107      0.028
## 57  5.888 1.120   1.402          0.798        0.498      0.215      0.060
## 58  5.886 1.327   1.419          1.088        0.445      0.069      0.140
## 59  5.860 0.642   1.236          0.828        0.507      0.246      0.078
## 60  5.809 1.173   1.508          0.729        0.410      0.146      0.096
## 61  5.779 0.776   1.209          0.706        0.511      0.137      0.064
## 62  5.758 1.201   1.410          0.828        0.199      0.081      0.020
## 63  5.743 0.855   1.475          0.777        0.514      0.184      0.080
## 64  5.718 1.263   1.252          1.042        0.417      0.191      0.162
## 65  5.697 0.960   1.274          0.854        0.455      0.083      0.027
## 66  5.693 1.221   1.431          0.999        0.508      0.047      0.025
## 67  5.653 0.677   0.886          0.535        0.313      0.220      0.098
## 68  5.648 1.183   1.452          0.726        0.334      0.082      0.031
## 69  5.631 0.807   1.293          0.657        0.558      0.117      0.107
```

```
## 70  5.603 1.004   1.383          0.854        0.282      0.137      0.039
## 71  5.529 0.685   1.328          0.739        0.245      0.181      0.000
## 72  5.525 1.044   1.303          0.673        0.416      0.133      0.152
## 73  5.523 1.051   1.361          0.871        0.197      0.142      0.080
## 74  5.467 0.493   1.098          0.718        0.389      0.230      0.144
## 75  5.432 1.155   1.266          0.914        0.296      0.119      0.022
## 76  5.430 1.438   1.277          1.122        0.440      0.258      0.287
## 77  5.425 1.015   1.401          0.779        0.497      0.113      0.101
## 78  5.386 0.945   1.212          0.845        0.212      0.263      0.006
## 79  5.373 1.183   1.360          0.808        0.195      0.083      0.106
## 80  5.339 1.221   1.171          0.828        0.508      0.260      0.024
## 81  5.323 1.067   1.465          0.789        0.235      0.094      0.142
## 82  5.287 1.181   1.156          0.999        0.067      0.000      0.034
## 83  5.285 0.948   1.531          0.667        0.317      0.235      0.038
## 84  5.274 0.983   1.294          0.838        0.345      0.185      0.034
## 85  5.265 0.696   1.111          0.245        0.426      0.215      0.041
## 86  5.261 0.551   1.438          0.723        0.508      0.300      0.023
## 87  5.247 1.052   1.538          0.657        0.394      0.244      0.028
## 88  5.211 1.002   1.160          0.785        0.086      0.073      0.114
## 89  5.208 0.801   0.782          0.782        0.418      0.036      0.076
## 90  5.208 1.043   1.147          0.769        0.351      0.035      0.182
## 91  5.197 0.987   1.224          0.815        0.216      0.166      0.027
## 92  5.192 0.931   1.203          0.660        0.491      0.498      0.028
## 93  5.191 1.029   1.125          0.893        0.521      0.058      0.100
## 94  5.175 0.741   1.346          0.851        0.543      0.147      0.073
## 95  5.082 0.813   1.321          0.604        0.457      0.370      0.167
## 96  5.044 0.549   0.910          0.331        0.381      0.187      0.037
## 97  5.011 1.092   1.513          0.815        0.311      0.081      0.004
## 98  4.996 0.611   0.868          0.486        0.381      0.245      0.040
## 99  4.944 0.569   0.808          0.232        0.352      0.154      0.090
## 100 4.913 0.446   1.226          0.677        0.439      0.285      0.089
## 101 4.906 0.837   1.225          0.815        0.383      0.110      0.130
## 102 4.883 0.393   0.437          0.397        0.349      0.175      0.082
## 103 4.812 0.673   0.799          0.508        0.372      0.105      0.093
## 104 4.799 1.057   1.183          0.571        0.295      0.043      0.055
## 105 4.796 0.764   1.030          0.551        0.547      0.266      0.164
## 106 4.722 0.960   1.351          0.469        0.389      0.130      0.055
## 107 4.719 0.947   0.848          0.874        0.383      0.178      0.027
## 108 4.707 0.960   1.427          0.805        0.154      0.064      0.047
## 109 4.700 0.574   1.122          0.637        0.609      0.232      0.062
## 110 4.696 0.657   1.247          0.672        0.225      0.103      0.066
## 111 4.681 0.450   1.134          0.571        0.292      0.153      0.072
## 112 4.668 0.000   0.698          0.268        0.559      0.243      0.270
## 113 4.639 0.879   1.313          0.477        0.401      0.070      0.056
## 114 4.628 0.138   0.774          0.366        0.318      0.188      0.102
## 115 4.587 0.331   1.056          0.380        0.255      0.177      0.113
## 116 4.559 0.850   1.055          0.815        0.283      0.095      0.064
## 117 4.548 1.100   0.842          0.785        0.305      0.270      0.125
## 118 4.534 0.380   0.829          0.375        0.332      0.207      0.086
## 119 4.519 0.886   0.666          0.752        0.346      0.043      0.164
## 120 4.516 0.308   0.939          0.428        0.382      0.269      0.167
## 121 4.509 0.512   0.983          0.581        0.431      0.372      0.053
## 122 4.490 0.570   1.167          0.489        0.066      0.106      0.088
## 123 4.466 0.204   0.986          0.390        0.494      0.197      0.138
```

```
## 124 4.461 0.921   1.000           0.815        0.167      0.059      0.055
## 125 4.456 0.562   0.928           0.723        0.527      0.166      0.143
## 126 4.437 1.043   0.980           0.574        0.241      0.148      0.089
## 127 4.418 0.094   1.125           0.357        0.269      0.212      0.053
## 128 4.390 0.385   1.105           0.308        0.327      0.153      0.052
## 129 4.374 0.268   0.841           0.242        0.309      0.252      0.045
## 130 4.366 0.949   1.265           0.831        0.470      0.244      0.047
## 131 4.360 0.710   1.181           0.555        0.525      0.566      0.172
## 132 4.350 0.350   0.766           0.192        0.174      0.198      0.078
## 133 4.332 0.820   1.390           0.739        0.178      0.187      0.010
## 134 4.286 0.336   1.033           0.532        0.344      0.209      0.100
## 135 4.212 0.811   1.149           0.000        0.313      0.074      0.135
## 136 4.189 0.332   1.069           0.443        0.356      0.252      0.060
## 137 4.166 0.913   1.039           0.644        0.241      0.076      0.067
## 138 4.107 0.578   1.058           0.426        0.431      0.247      0.087
## 139 4.085 0.275   0.572           0.410        0.293      0.177      0.085
## 140 4.015 0.755   0.765           0.588        0.498      0.200      0.085
## 141 3.975 0.073   0.922           0.443        0.370      0.233      0.033
## 142 3.973 0.274   0.757           0.505        0.142      0.275      0.078
## 143 3.933 0.274   0.916           0.555        0.148      0.169      0.041
## 144 3.802 0.489   1.169           0.168        0.359      0.107      0.093
## 145 3.775 0.046   0.447           0.380        0.220      0.176      0.180
## 146 3.663 0.366   1.114           0.433        0.361      0.151      0.089
## 147 3.597 0.323   0.688           0.449        0.026      0.419      0.110
## 148 3.488 1.041   1.145           0.538        0.455      0.025      0.100
## 149 3.462 0.619   0.378           0.440        0.013      0.331      0.141
## 150 3.410 0.191   0.560           0.495        0.443      0.218      0.089
## 151 3.380 0.287   1.163           0.463        0.143      0.108      0.077
## 152 3.334 0.359   0.711           0.614        0.555      0.217      0.411
## 153 3.231 0.476   0.885           0.499        0.417      0.276      0.147
## 154 3.203 0.350   0.517           0.361        0.000      0.158      0.025
## 155 3.083 0.026   0.000           0.105        0.225      0.235      0.035
## 156 2.853 0.306   0.575           0.295        0.010      0.202      0.091
```

Correlation of the variables

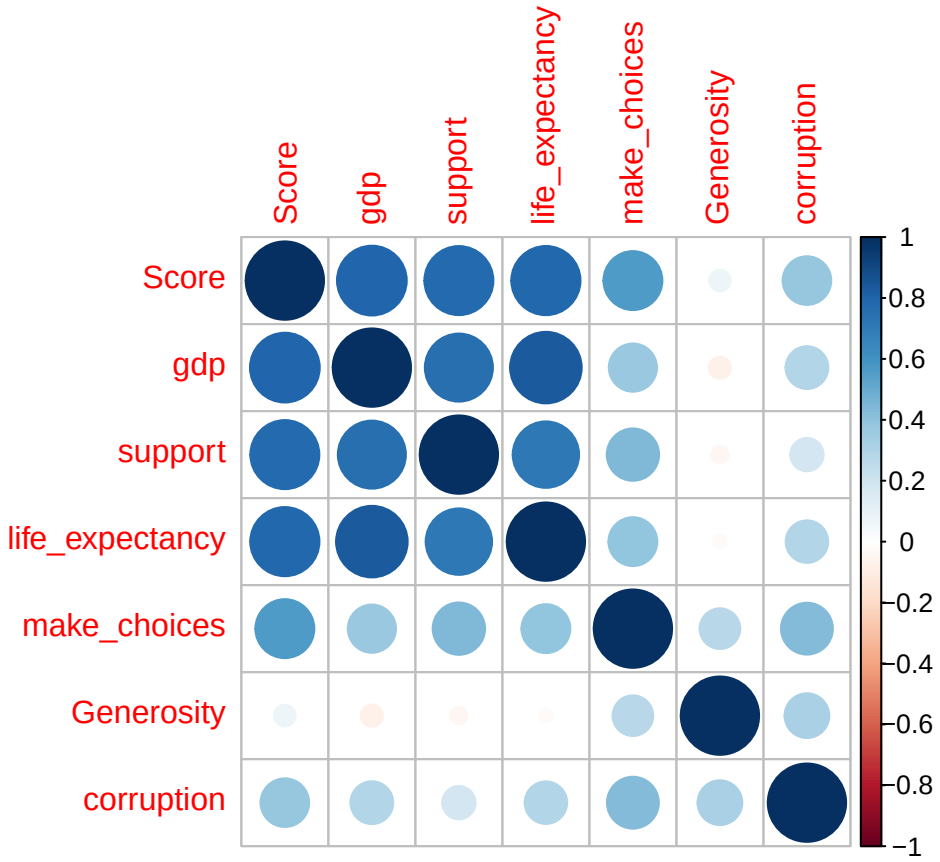
Below will show the correlation between the variables. This will show which variables has the highest correlation to the score variable.

```
# calculates the correlation of the data
cordata = num_data[,c(1,2,3,4,5,6,7)]
corr <- round(cor(cordata), 3)
corr
```

```
##                 Score    gdp support life_expectancy make_choices Generosity
## Score           1.000  0.794   0.777           0.780        0.567      0.076
## gdp             0.794  1.000   0.755           0.835        0.379     -0.080
## support         0.777  0.755   1.000           0.719        0.447     -0.048
## life_expectancy 0.780  0.835   0.719           1.000        0.390     -0.030
## make_choices    0.567  0.379   0.447           0.390        1.000      0.270
## Generosity      0.076 -0.080  -0.048          -0.030        0.270      1.000
## corruption      0.386  0.299   0.182           0.295        0.439      0.327
##                 corruption
```

```
## Score                0.386
## gdp                  0.299
## support              0.182
## life_expectancy      0.295
## make_choices         0.439
## Generosity           0.327
## corruption           1.000
```

```
corrplot(corr, method = "circle")
```



Conclusion

With the data has been collected and shown we are now able to see which variables have a high correlation to the Score of the country. From the Corrplot we see the two highest variables that affects Score is GDP per capita and Social Support. While corruption and genrosity had a negative correlation to the Score.