# World Hapiness

*Silpa Velagapudi*

*December 30, 2019*

## Contents



#Introduction: Analysing the data using different algorithms used as part of the course.

The happiness scores and rankings use data from the Gallup World Poll. The scores are based on answers to the main life evaluation question asked in the poll. This question, known as the Cantril ladder, asks respondents to think of a ladder with the best possible life for them being a 10 and the worst possible life being a 0 and to rate their own current lives on that scale. The scores are from nationally representative samples for the years 2013-2016 and use the Gallup weights to make the estimates representative. The columns following the happiness score estimate the extent to which each of six factors – economic production, social support, life expectancy, freedom, absence of corruption, and generosity – contribute to making life evaluations higher in each country than they are in Dystopia, a hypothetical country that has values equal to the world's lowest national averages for each of the six factors. They have no impact on the total score reported for each country, but they do explain why some countries rank higher than others..

### 0.1 Method/Analysis

The following 5-step process was utilized:

1. Data Extraction
2. In-depth Analysis of the Data
3. Using clustering method for analysis

### 0.2 Pre-requisites: Loading Packages and Libraries

1. First, we load all the necessary packages and libraries.

```
## Loading required package: 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

## Loading required package: tidyr

## Loading required package: factoextra

## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

## Loading required package: tidyverse
```

```
## -- Attaching packages ---------------------------------------------------------------------------
## v tibble  2.1.3     v purrr   0.3.2
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.3     v forcats 0.4.0
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## Loading required package: cluster
## Loading required package: RColorBrewer
```

2. Using the file downloaded from https://www.kaggle.com/unsdsn/world-happiness ## Data Extraction Data set contains 156 observations with 9 different columns as mentioned below
   (a) Overallrank
   (b) Country
   (c) Score
   (d) Socialsupport
   (e) Healthylifeexpectancy
   (f) Freedomtomakelifechoices
   (g) Generosity
   (h) Perceptionsofcorruption

```
world_happiness <- read_csv("./Data_files/2019.csv")
```

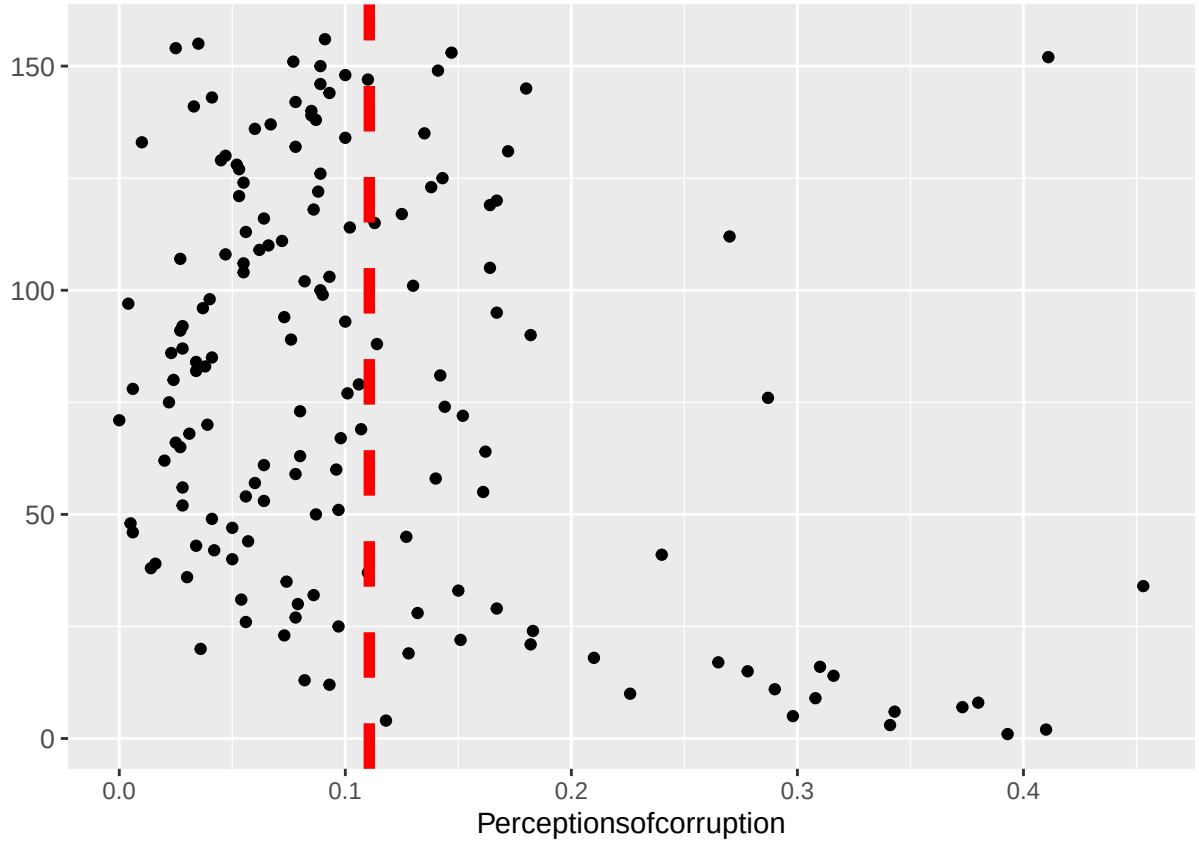
```
## Parsed with column specification:
## cols(
##   Overallrank = col_double(),
##   Country = col_character(),
##   Score = col_double(),
##   GDPpercapita = col_double(),
##   Socialsupport = col_double(),
##   Healthylifeexpectancy = col_double(),
##   Freedomtomakelifechoices = col_double(),
##   Generosity = col_double(),
##   Perceptionsofcorruption = col_double()
## )
```

```
head(world_happiness)
```

```
## # A tibble: 6 x 9
##   Overallrank Country Score GDPpercapita Socialsupport Healthylifeexpe~
##         <dbl> <chr>   <dbl>        <dbl>         <dbl>            <dbl>
## 1           1 Finland  7.77         1.34          1.59            0.986
## 2           2 Denmark  7.6          1.38          1.57            0.996
## 3           3 Norway   7.55         1.49          1.58            1.03
## 4           4 Iceland  7.49         1.38          1.62            1.03
## 5           5 Nether~  7.49         1.40          1.52            0.999
## 6           6 Switze~  7.48         1.45          1.53            1.05
## # ... with 3 more variables: Freedomtomakelifechoices <dbl>, Generosity <dbl>,
## #   Perceptionsofcorruption <dbl>
```

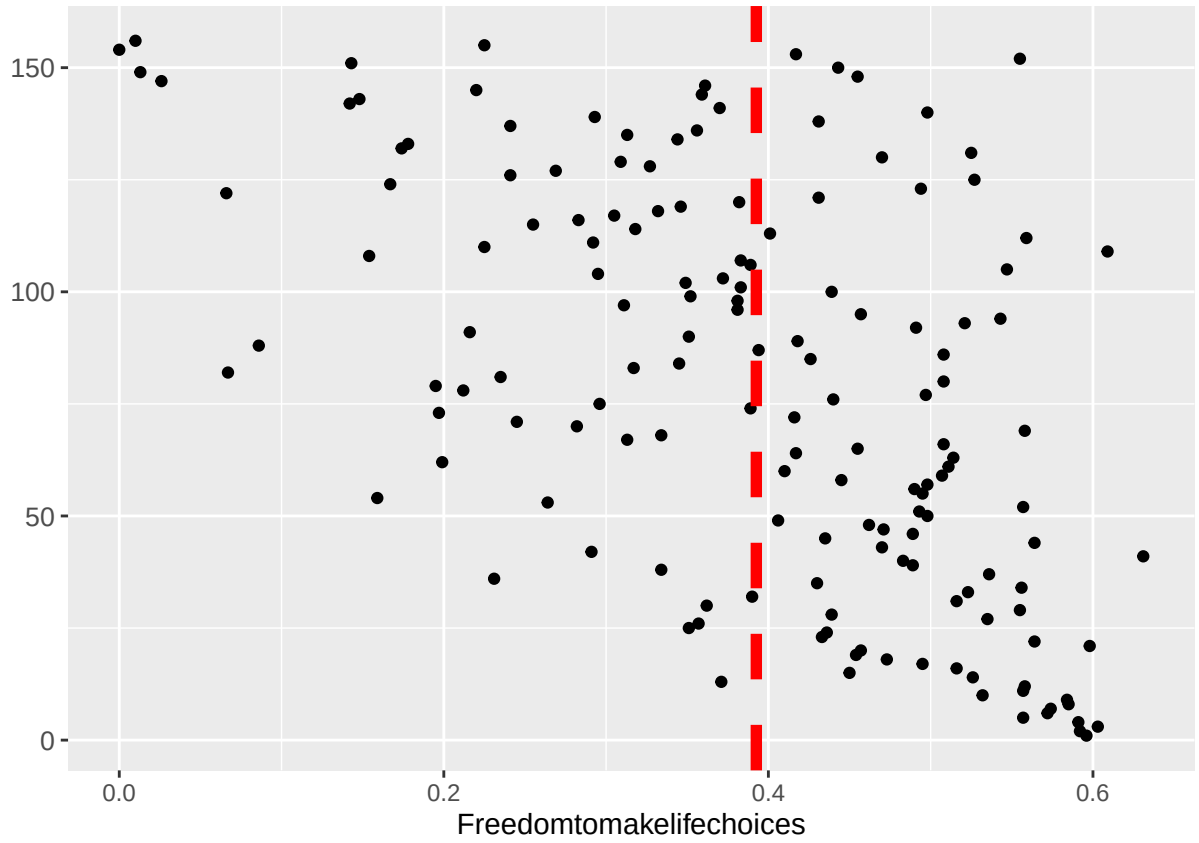
Below graph shows relation between rank and corruption. From graph we can see that, though there are countries with less than mean corruption are having high ranking and some of the countries are having high corruption are having less ranking also. There might be slight dependency on this but not totally.

```
world_happiness  %>% ggplot(aes(Overallrank,Perceptionsofcorruption)) +  geom_point(stat="identity") +
  coord_flip() + geom_hline(aes(yintercept = mean(Perceptionsofcorruption)), col="red", lwd=2, lty=2) +
  theme(axis.text.y = element_text(size = 10)) +
  xlab("")
```



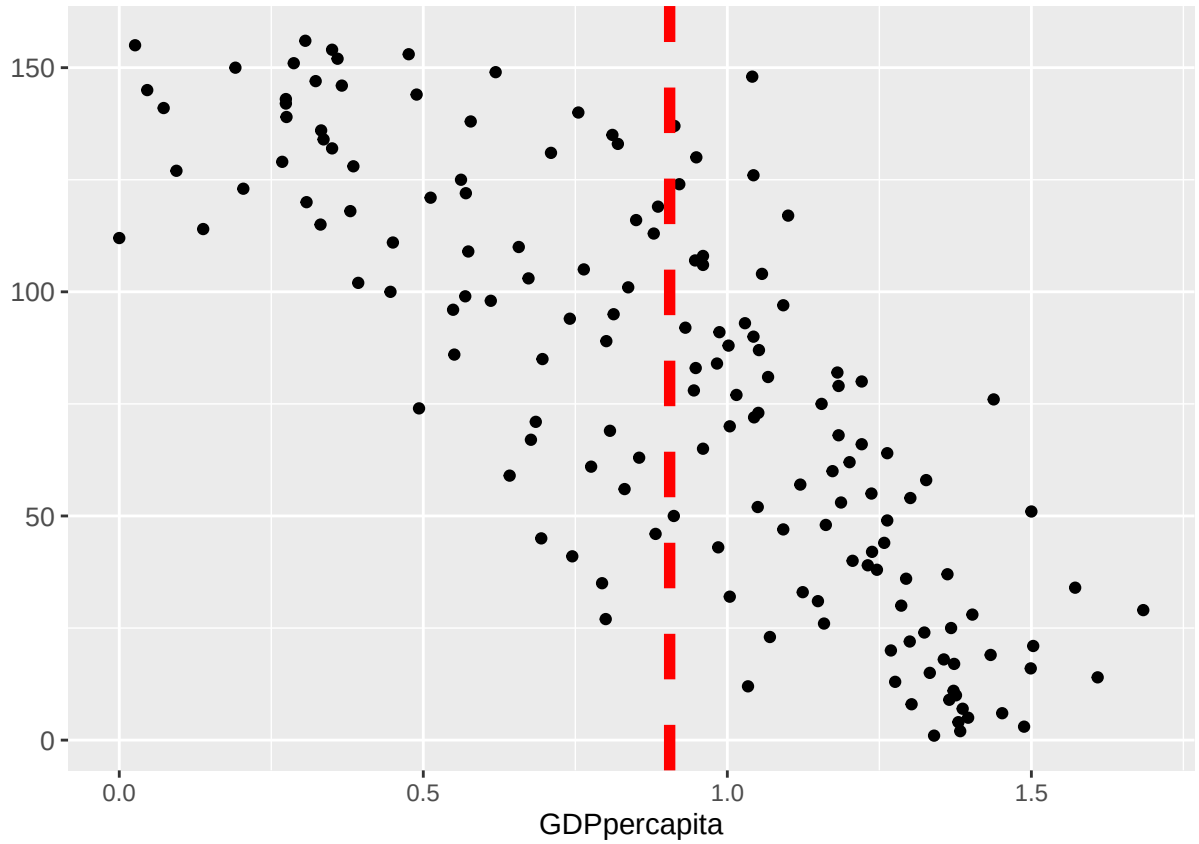

Below graph shows relation between rank and freedom. From graph we can see that, though there are countries with less than mean freedom are having high ranking and some of the countries are having high freedom are also having high ranking. There might be slight dependency on this but not totally.

```
world_happiness  %>% ggplot(aes(Overallrank,Freedomtomakelifechoices)) +  geom_point(stat="identity") +
  coord_flip() + geom_hline(aes(yintercept = mean(Freedomtomakelifechoices)), col="red", lwd=2, lty=2) +
  theme(axis.text.y = element_text(size = 10)) +
  xlab("")
```



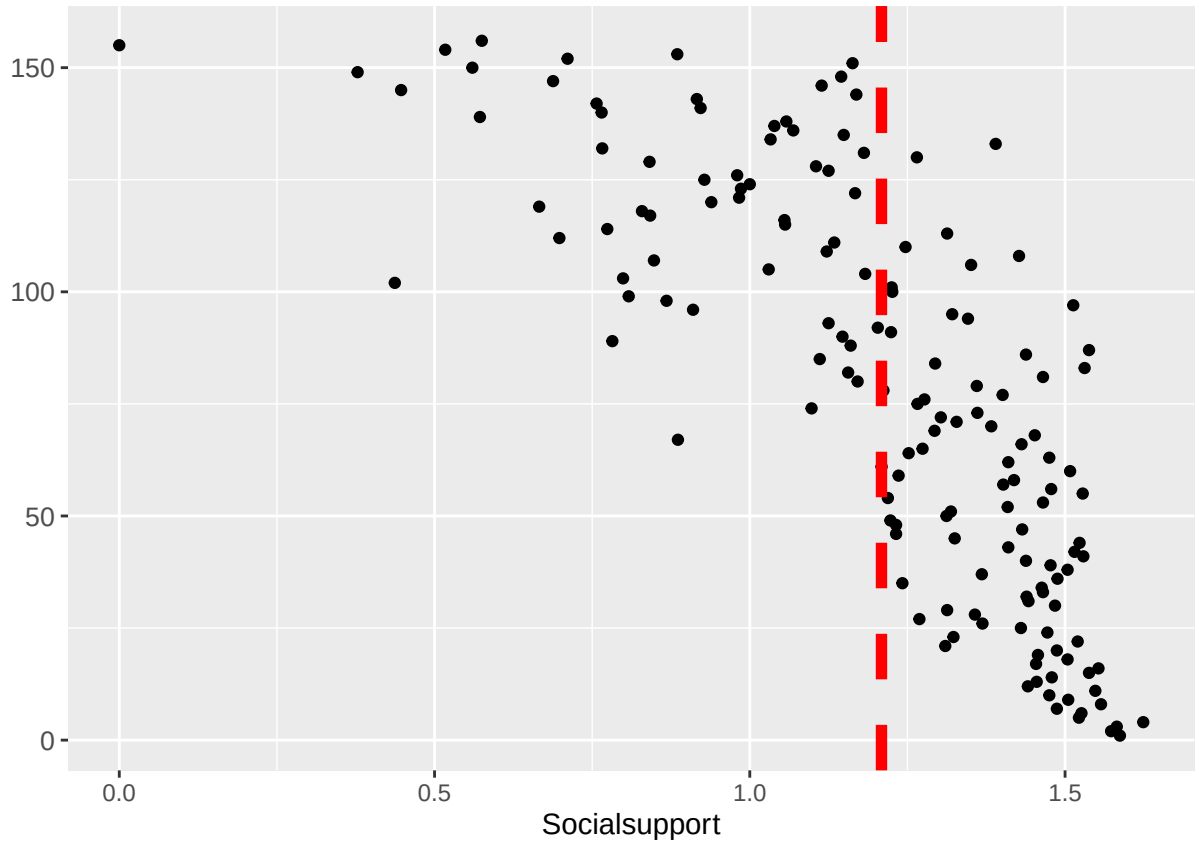

Below graph shows relation between rank and GPA. From graph we can see that, we can clearly see that, when GPA per capita is greater than mean we see those countries are having less ranking and countries which are having righ ranking are having GPA per captia less than mean.

```
world_happiness  %>% ggplot(aes(Overallrank,GDPpercapita)) +  geom_point(stat="identity") +
  coord_flip() + geom_hline(aes(yintercept = mean(GDPpercapita)), col="red", lwd=2, lty=2) +
  theme(axis.text.y = element_text(size = 10)) +
  xlab("")
```



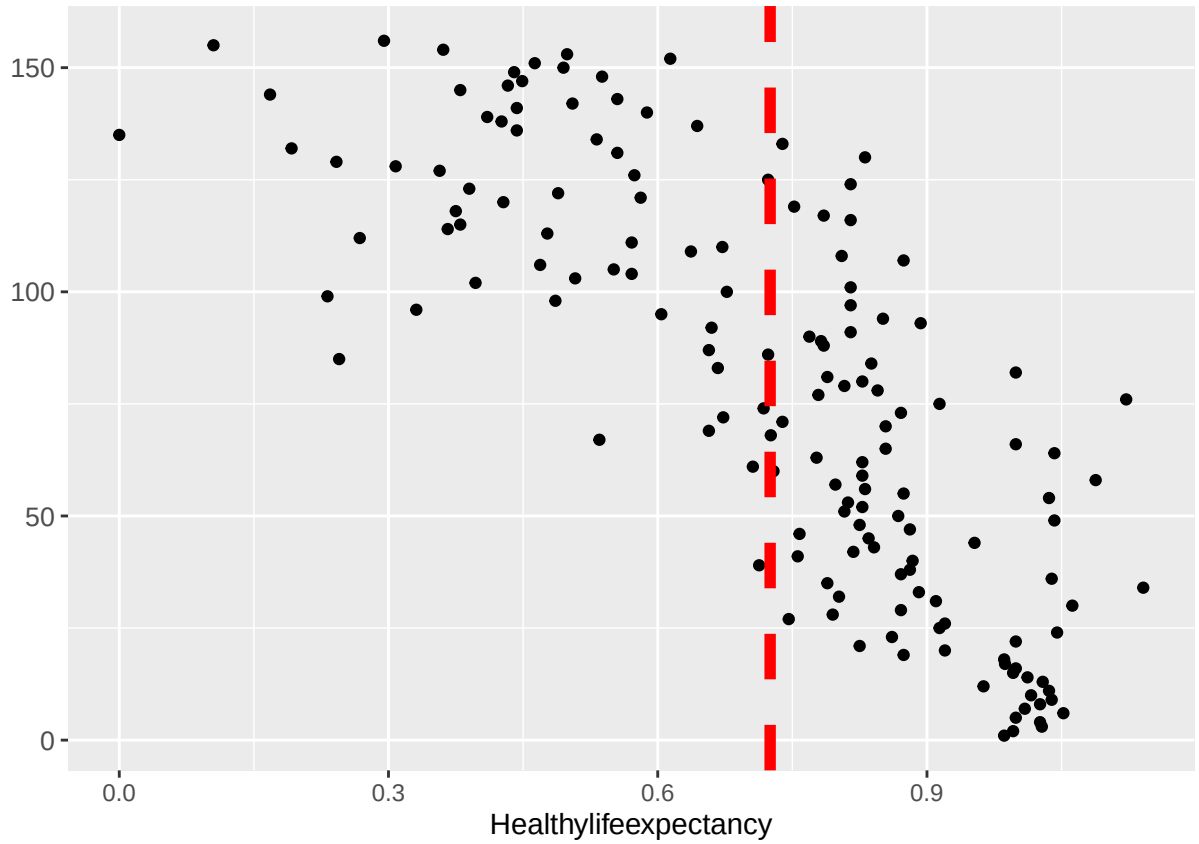

Below graph shows relation between rank and social support. From graph we can see that, we can clearly see that, when social support is greater than mean we see those countries are having good ranking and countries which are having high ranking are having Social support less than mean.

```
world_happiness  %>% ggplot(aes(Overallrank,Socialsupport)) +  geom_point(stat="identity") +
  coord_flip() + geom_hline(aes(yintercept = mean(Socialsupport)), col="red", lwd=2, lty=2) +
  theme(axis.text.y = element_text(size = 10)) +
  xlab("")
```



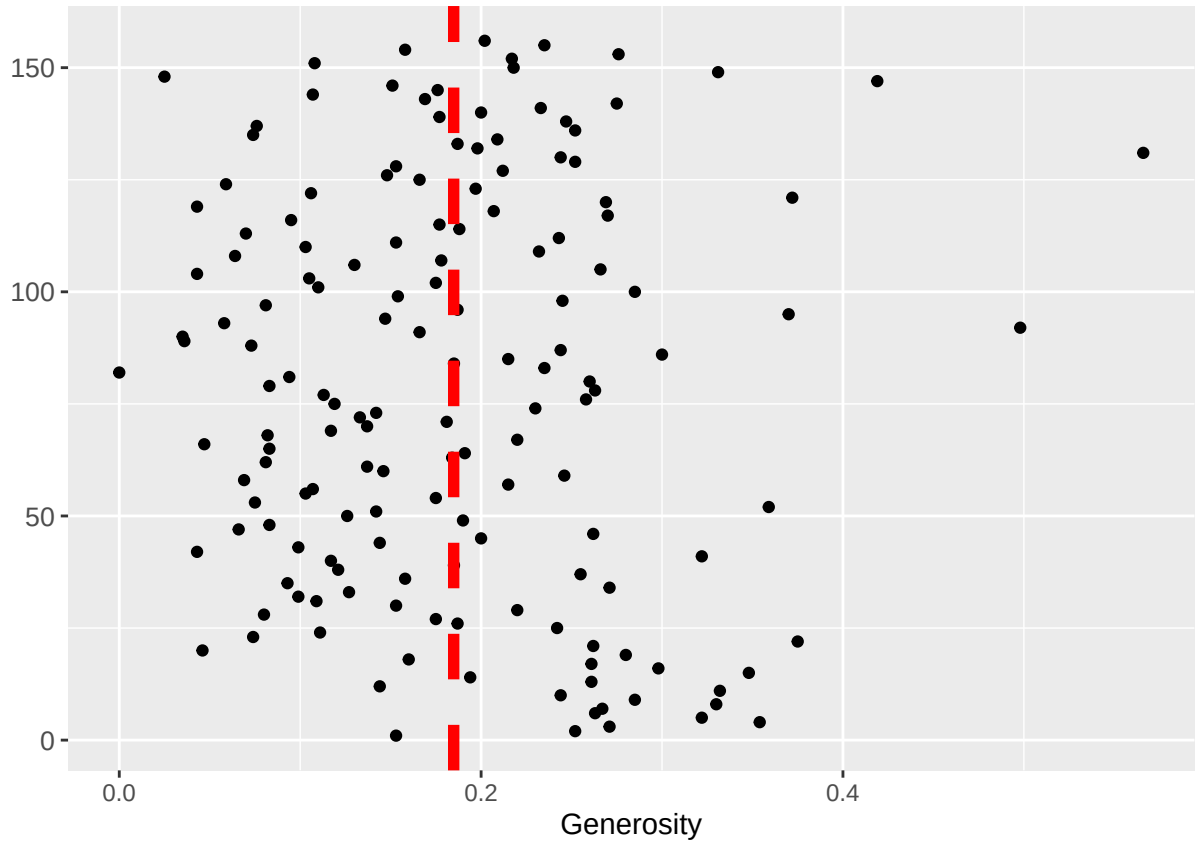

Below graph shows relation between rank and Health. From graph we can see that, we can clearly see that, when Health is greater than mean we see those countries are having good ranking and countries which are having high ranking are having Health less than mean.

```
world_happiness  %>% ggplot(aes(Overallrank,Healthylifeexpectancy)) +  geom_point(stat="identity") +
  coord_flip() + geom_hline(aes(yintercept = mean(Healthylifeexpectancy)), col="red", lwd=2, lty=2) +
  theme(axis.text.y = element_text(size = 10)) +
  xlab("")
```



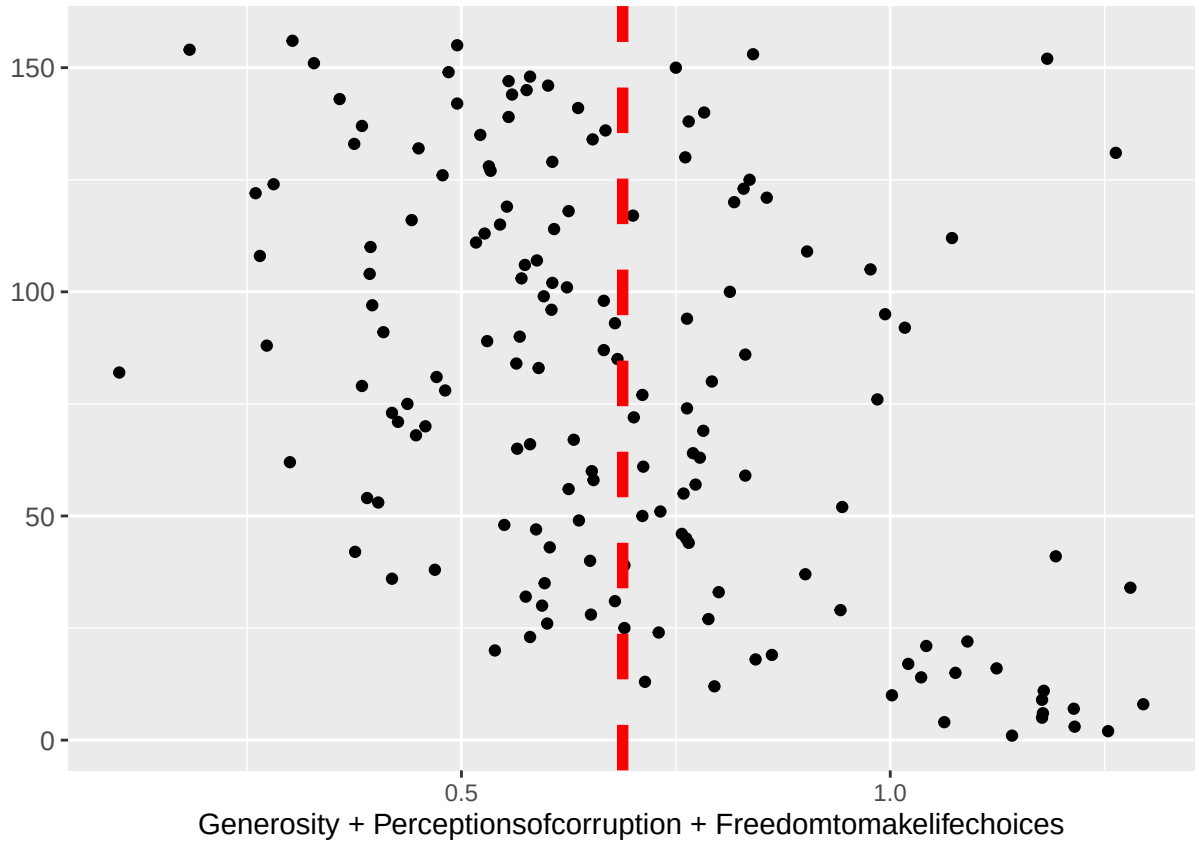

Below graph shows relation between rank and Generosity. From graph we can see that, though there are countries with less than mean Generosity are having high ranking and some of the countries are having high Generosity are also having high ranking. There might be slight dependency on this but not totally.

```
world_happiness  %>% ggplot(aes(Overallrank,Generosity)) +  geom_point(stat="identity") +
  coord_flip() + geom_hline(aes(yintercept = mean(Generosity)), col="red", lwd=2, lty=2) +
  theme(axis.text.y = element_text(size = 10)) +
  xlab("")
```



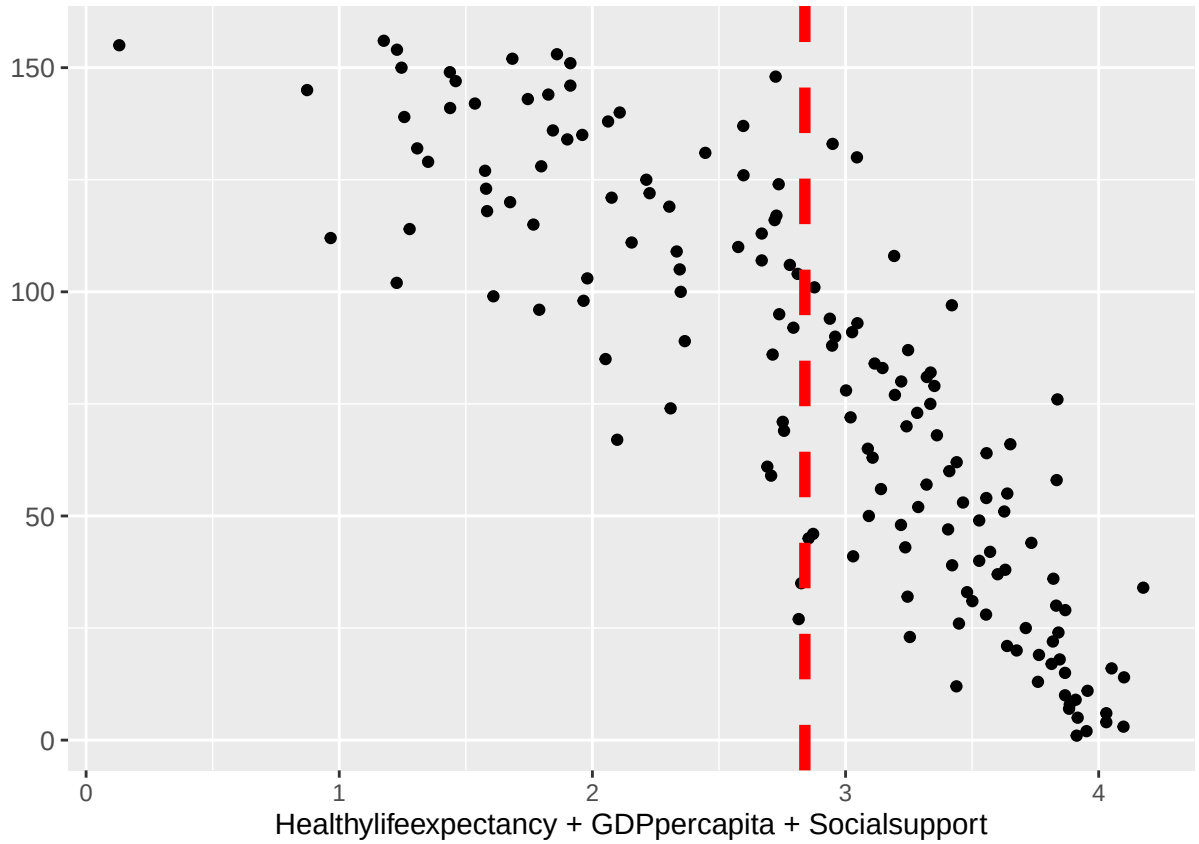

Below graph shows relation between rank and Generosity+Perceptionsofcorruption+Freedomtomakelifechoices.
Still the graph doesnt show proper dependency on them

```
world_happiness  %>% ggplot(aes(Overallrank,Generosity+Perceptionsofcorruption+Freedomtomakelifechoices
  coord_flip() + geom_hline(aes(yintercept = mean(Generosity+Perceptionsofcorruption+Freedomtomakelifec
  theme(axis.text.y = element_text(size = 10)) +
  xlab("")
```




Below graph shows relation between rank and Healthylifeexpectancy+GDPpercapita+Socialsupport. Any ranking of the country mainly depends on health , social support and GDP per capita.

```
world_happiness  %>% ggplot(aes(Overallrank,Healthylifeexpectancy+GDPpercapita+Socialsupport)) +  geom_
  coord_flip() + geom_hline(aes(yintercept = mean(Healthylifeexpectancy+GDPpercapita+Socialsupport)), c
  theme(axis.text.y = element_text(size = 10)) +
  xlab("")
```




The purpose of clustering analysis is to identify patterns in data and create groups according to those patterns. Therefore, if two points have similar characteristics, that means they have the same pattern and consequently, they belong to the same group. By doing clustering analysis we should be able to check what features usually appear together and see what characterizes a group.

The bigger is the K is choosed, the lower will be the variance within the groups in the clustering. If K is equal to the number of observations, then each point will be a group and the variance will be 0. It's interesting to find a balance between the number of groups and their variance. A variance of a group means how different the members of the group are. The bigger is the variance, the bigger is the dissimilarity in a group.

```
set.seed(1)
input <- world_happiness[,4:9]
#checking the details using number of centers as 4
kmeans(input, centers = 4, nstart = 20)
```

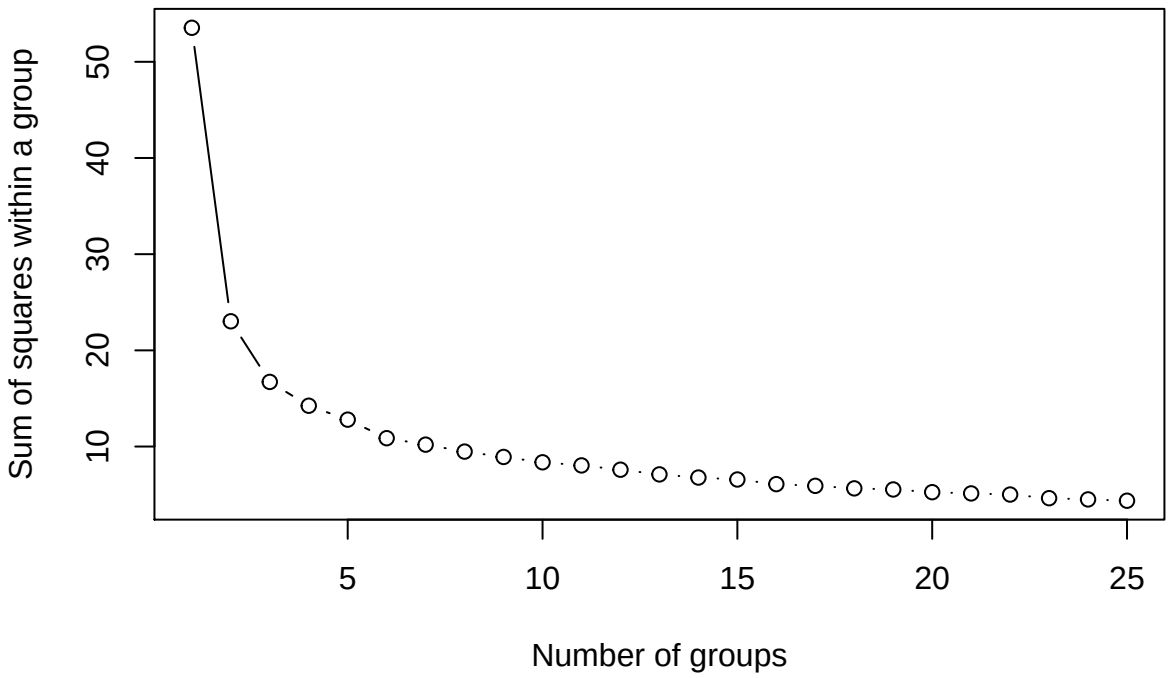
```
## K-means clustering with 4 clusters of sizes 44, 63, 34, 15
##
## Cluster means:
##   GDPpercapita Socialsupport Healthylifeexpectancy Freedomtomakelifechoices
## 1    1.3520000     1.4579545             0.9690682                 0.4835455
## 2    0.9646825     1.2971746             0.7749048                 0.3764444
## 3    0.5090588     0.9932647             0.4803529                 0.3764118
## 4    0.2421333     0.5954667             0.3565333                 0.2300667
##   Generosity Perceptionsofcorruption
## 1  0.2061136              0.18275000
## 2  0.1504921              0.06680952
## 3  0.2001765              0.10470588
## 4  0.2320000              0.09626667
```

```
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 2 2 1 1 1 1 2 1 1 2 1 1
##  [38] 1 1 1 2 1 2 1 2 2 2 2 1 2 1 2 2 1 1 2 2 1 2 2 2 2 2 1 2 1 3 2 2 2 2 2 2 3
##  [75] 2 1 2 2 2 2 2 2 2 2 3 2 2 2 3 2 2 2 2 2 2 3 2 3 3 3 2 4 3 2 3 2 2 2 3 2 3
## [112] 4 2 4 3 2 2 3 3 3 3 3 3 2 3 2 3 3 4 2 3 4 2 3 3 3 2 3 4 3 4 4 3 3 4 3 4 2
## [149] 4 4 3 3 3 4 4 4
##
## Within cluster sum of squares by cluster:
## [1] 2.849936 5.505285 3.843145 1.830729
##  (between_SS / total_SS =  73.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

The function below plots a chart showing the "within sum of squares" (withinss) by the number of groups (K value) chosen for several executions of the algorithm. The within sum of squares is a metric that shows how dissimilar are the members of a group., the greater is the sum, the greater is the dissimilarity within a group.

```
wssplot <- function(data, nc=15, seed=1){
               wss <- (nrow(data)-1)*sum(apply(data,2,var))
               for (i in 2:nc){
                    set.seed(seed)
                    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
                plot(1:nc, wss, type="b", xlab="Number of groups",
                     ylab="Sum of squares within a group")}

wssplot(input, nc = 25)
```




By Analysing the chart from right to left, we can see that when the number of groups (K) reduces from 20 to 19 there is a last increase in the sum of squares. That means that when it passes from 20 to 19 groups there is a reduction in the clustering compactness. Our goal, however, is not to achieve compactness of 100% — for that, we would just take each observation as a group. The main purpose is to find a fair number of groups that could explain satisfactorily a considerable part of the data.

Using 3 groups (K = 4) we had 78.3% of well-grouped data. Using 4 groups (K = 20) that value raised to 90.3%, which is a good value for us.

```
set.seed(1)
clustering <- kmeans(input, centers = 20, nstart = 20)
clustering
```

```
## K-means clustering with 20 clusters of sizes 1, 9, 3, 13, 4, 5, 6, 10, 6, 8, 8, 12, 17, 3, 7, 5, 9,
##
## Cluster means:
##    GDPpercapita Socialsupport Healthylifeexpectancy Freedomtomakelifechoices
## 1     0.0260000     0.0000000             0.1050000                0.22500000
## 2     0.9427778     1.4677778             0.7708889                0.47333333
## 3     1.2123333     1.2136667             0.9830000                0.17400000
## 4     0.7163077     1.2781538             0.7535385                0.44676923
## 5     0.3952500     1.0950000             0.5195000                0.16225000
## 6     0.9962000     1.2590000             0.5456000                0.39120000
## 7     0.3703333     0.6135000             0.3736667                0.06083333
## 8     1.3195000     1.4004000             1.0369000                0.39320000
## 9     1.4808333     1.3540000             0.8406667                0.51250000
## 10    0.9326250     0.9015000             0.7551250                0.29800000
## 11    1.0432500     1.2311250             0.8411250                0.46225000
```

```
## 12    1.1750000     1.4529167             0.8921667               0.49325000
## 13    1.4075294     1.5290000             1.0204118               0.55588235
## 14    0.8180000     1.2350000             0.6063333               0.49100000
## 15    0.2002857     0.5998571             0.4185714               0.39100000
## 16    0.6228000     1.0294000             0.1952000               0.36620000
## 17    1.1744444     1.4657778             0.8006667               0.28588889
## 18    0.6068182     0.9474545             0.5683636               0.44681818
## 19    0.2797273     1.0017273             0.3937273               0.34536364
## 20    0.9690000     1.3063750             0.8190000               0.20875000
##    Generosity Perceptionsofcorruption
## 1  0.23500000              0.03500000
## 2  0.19577778              0.07366667
## 3  0.09800000              0.03733333
## 4  0.18123077              0.07038462
## 5  0.13400000              0.06950000
## 6  0.08020000              0.08360000
## 7  0.26383333              0.08716667
## 8  0.17930000              0.13110000
## 9  0.20650000              0.13600000
## 10 0.11312500              0.08337500
## 11 0.12037500              0.06812500
## 12 0.12416667              0.06733333
## 13 0.28352941              0.30900000
## 14 0.47800000              0.12233333
## 15 0.19914286              0.17414286
## 16 0.14740000              0.07920000
## 17 0.08955556              0.05766667
## 18 0.23263636              0.10145455
## 19 0.21018182              0.08509091
## 20 0.15212500              0.04462500
##
## Clustering vector:
##   [1] 13 13 13 13 13 13 13 13 13 13 13 12  8 13 13 13 13  8  9 12  9 13 11  8  8
##  [26] 12  4  9  9  8 12  2 12 13  4  8  9 17 12 12  2 17  2 12  4  4 12 11  8 11
##  [51]  9  2 17  3 12  2 12  8  4 17  4 17  2  8 11 12 18 17  4 20  4  6 20 18  3
##  [76]  8  2 20 17 11 17  3  2 20 16  4  2 20 10 11 20 14 11  4 14 16 17 18 16  4
## [101]  4 15 18  6 18  6 10 20 18  4  5 15  6 15 19 10 10 19 10 19 18  5 19 10 18
## [126] 10 19 19 19 11 14  7 20 19 16 19 10 18 15 18 19  7  5 16 15 19  7  6  7 15
## [151]  5 15 18  7  1  7
##
## Within cluster sum of squares by cluster:
##  [1] 0.00000000 0.34506556 0.06917600 0.50138538 0.14229550 0.11381880
##  [7] 0.34536867 0.29730640 0.14201517 0.37615150 0.22282350 0.23302717
## [13] 0.38290835 0.07679533 0.51416171 0.25860560 0.13414489 0.47102982
## [19] 0.36647364 0.19431812
##  (between_SS / total_SS =  90.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Using silhouette coefficient (silhouette width) to evaluate the goodness of our clustering.

```r
sil <- silhouette(clustering$cluster, dist(input))
fviz_silhouette(sil)
```

```
##    cluster size ave.sil.width
## 1        1    1          0.00
## 2        2    9          0.07
## 3        3    3          0.20
## 4        4   13          0.19
## 5        5    4          0.15
## 6        6    5          0.32
## 7        7    6          0.26
## 8        8   10          0.15
## 9        9    6          0.31
## 10      10    8          0.20
## 11      11    8          0.18
## 12      12   12          0.26
## 13      13   17          0.43
## 14      14    3          0.33
## 15      15    7          0.10
## 16      16    5          0.19
## 17      17    9          0.33
## 18      18   11          0.21
## 19      19   11          0.26
## 20      20    8          0.21
```