World Happiness

Silpa Velagapudi

December 30, 2019

Contents

1. **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.

We are trying to analyse the data to see what factors depends on the happiness of the people in the country based on the details provided.

2. ***Data Extraction***:

Pre-requisites: Loading Packages and Libraries

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: caret

## Loading required package: lattice

## Loading required package: ggplot2

## Loading required package: factoextra

## 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 tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()

## Loading required package: cluster

## Loading required package: 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
```

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:

1. Overallrank
2. Country
3. Score
4. GDPCapita
5. Socialsupport
6. Healthylifeexpectancy
7. Freedomtomakelifechoices
8. Generosity
9. Perceptionsofcorruption

```
#Reading data file
world_happiness <- (read.csv("./Data_files/2019.csv", header=TRUE))
#Viewing the first 6 records of the file
head(world_happiness)
```

```
##   Overallrank     Country Score GDPpercapita Socialsupport
## 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
##   Healthylifeexpectancy Freedomtomakelifechoices 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
##   Perceptionsofcorruption
## 1                   0.393
## 2                   0.410
## 3                   0.341
## 4                   0.118
## 5                   0.298
## 6                   0.343
```

3. **In-Depth analysis of data**

Perceptionsofcorruption

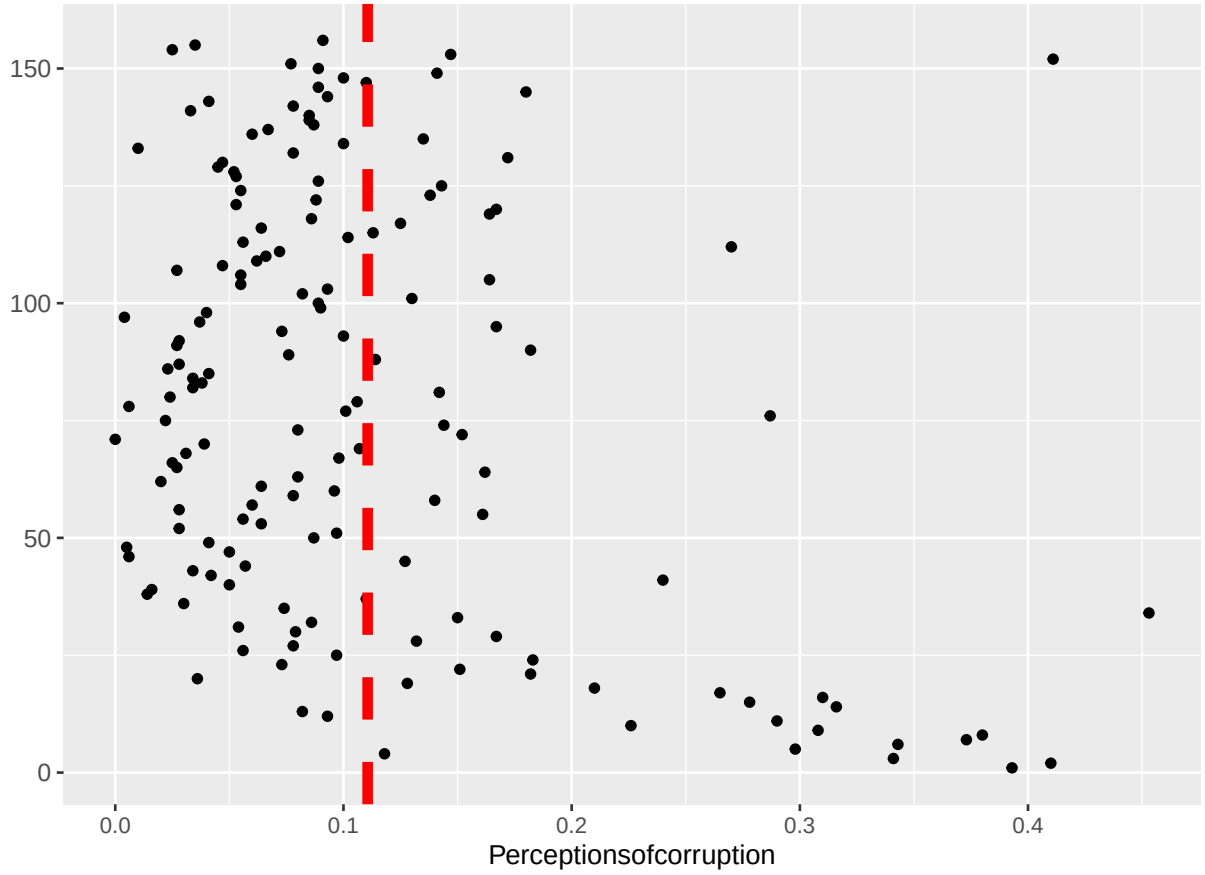
Below graph shows relation between rank and corruption. From graph we can see,

a. Some countries having good ranks though they have high corruption

b. some countries have poor ranking though they have corruption less than mean.

For this we can say that rank is not depending on corruption.

```
#viewing the Perceptionsofcorruption
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("")
```



Freedomtomakelifechoices

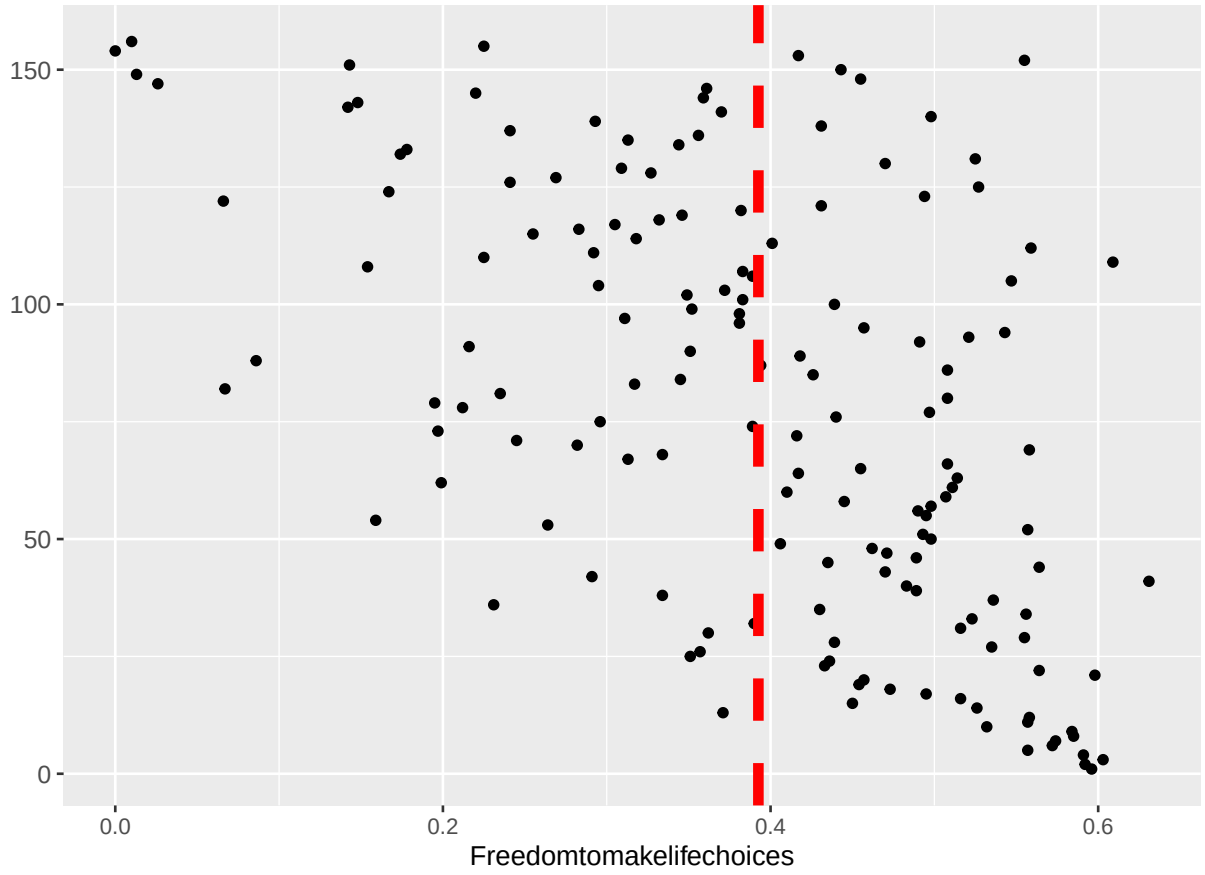
Below graph shows relation between rank and Freedom. From graph we can see,

a. Some countries having good ranks though they have high Freedom

b. some countries have poor ranking though they have Freedom less than mean.

For this we can say that rank is not depending on Freedom

```
#viewing the Freedomtomakelifechoices
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("")
```



GDPpercapita

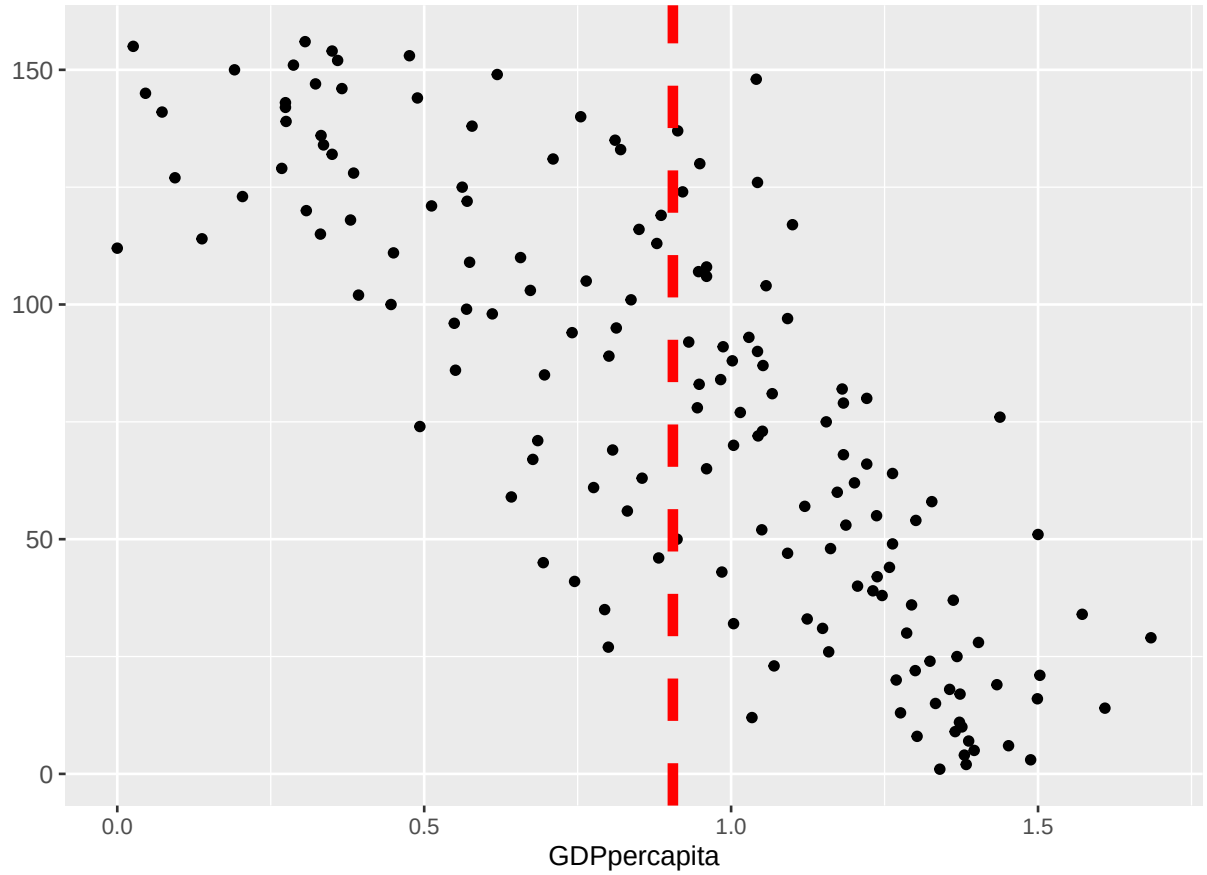
Below graph shows relation between rank and GDP. From graph we can see,

a. countries which are having good GDP greater than mean are having good rankings

b. Countries which are having less GDP less than mean are having high rankings

For this we can say that rank is depends on GDP per capita

```
#viewing GDPpercapita
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("")
```



Socialsupport

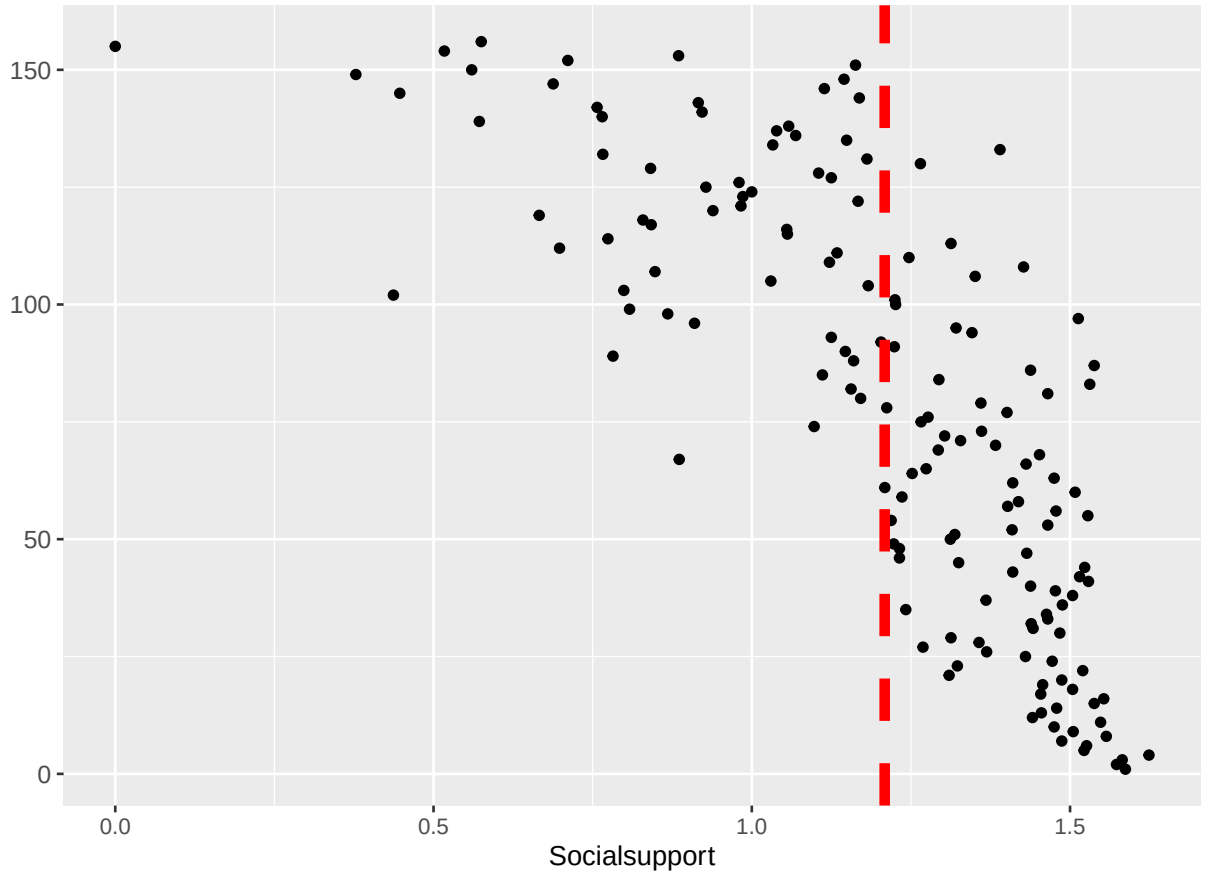
Below graph shows relation between rank and Social Support. From graph we can see,

a. countries which are having good social support greater than mean are having good rankings

b. Countries which are having less social support less than mean are having high rankings

For this we can say that rank is depends on social support

```
#viewing Socialsupport
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("")
```



Healthylifeexpectancy

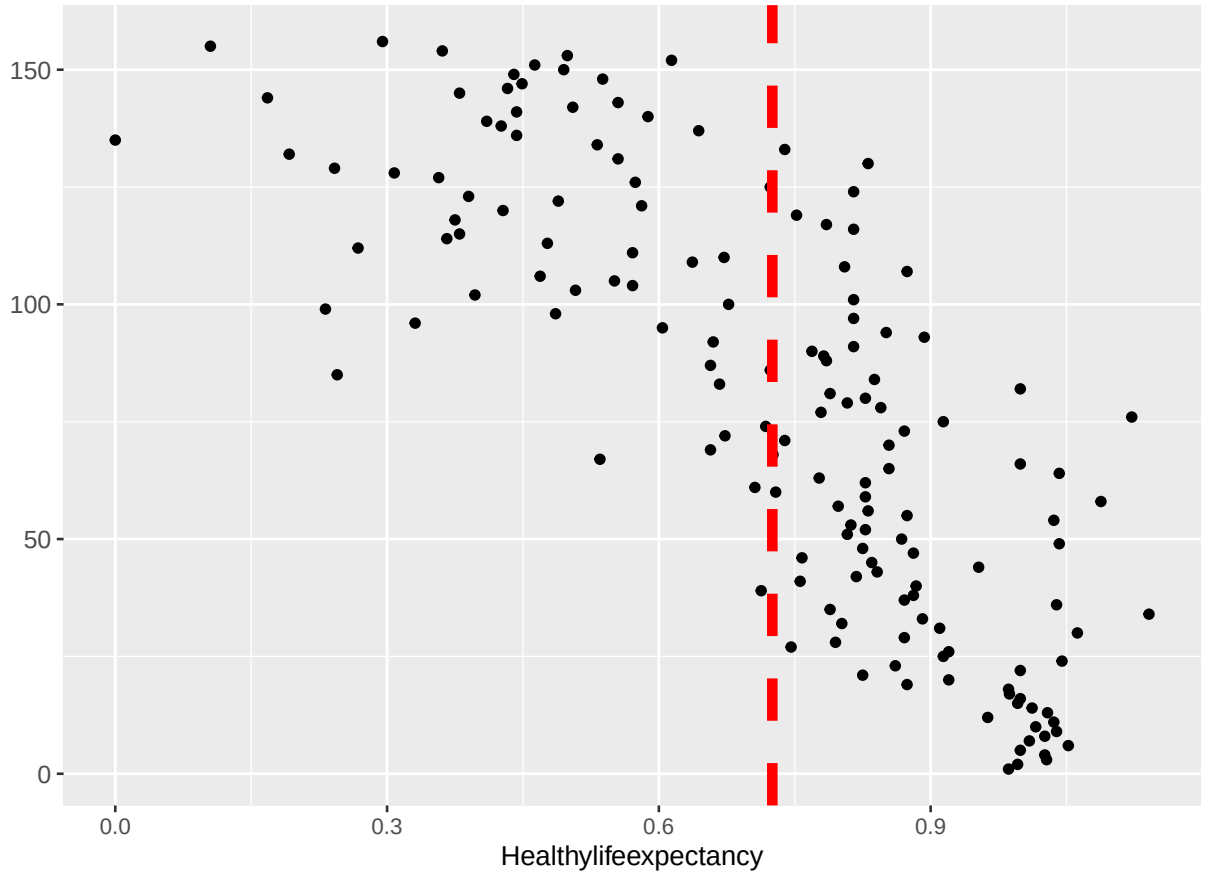
Below graph shows relation between rank and Healthy Life. From graph we can see,

a. countries which are having good healthy life greater than mean are having good rankings

b. Countries which are having less healthy life less than mean are having high rankings

For this we can say that rank is depends on healthy life

```
#viewing Healthylifeexpectancy
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("")
```



Generosity

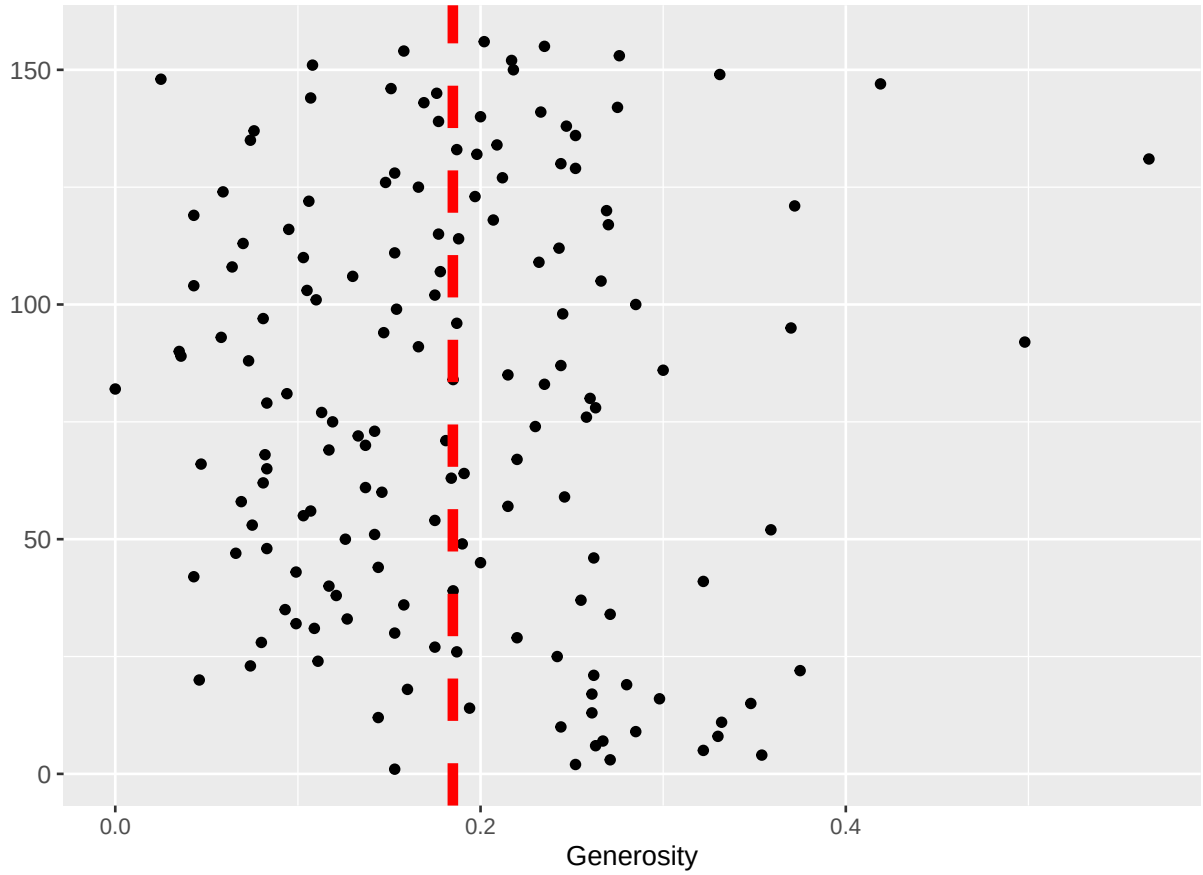
Below graph shows relation between rank and Generosity. From graph we can see,

a. Some countries having good ranks though they have high generosity

b. some countries have poor ranking though they have generosity less than mean.

For this we can say that rank is not depending on generosity

```
#viewing Generosity
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("")
```



4. **Clustering:**

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)
#creating inut file for clustering
input <- world_happiness[,4:9]
#checking the details using number of centers as 3
kmeans(input, centers = 3, nstart = 20)
```

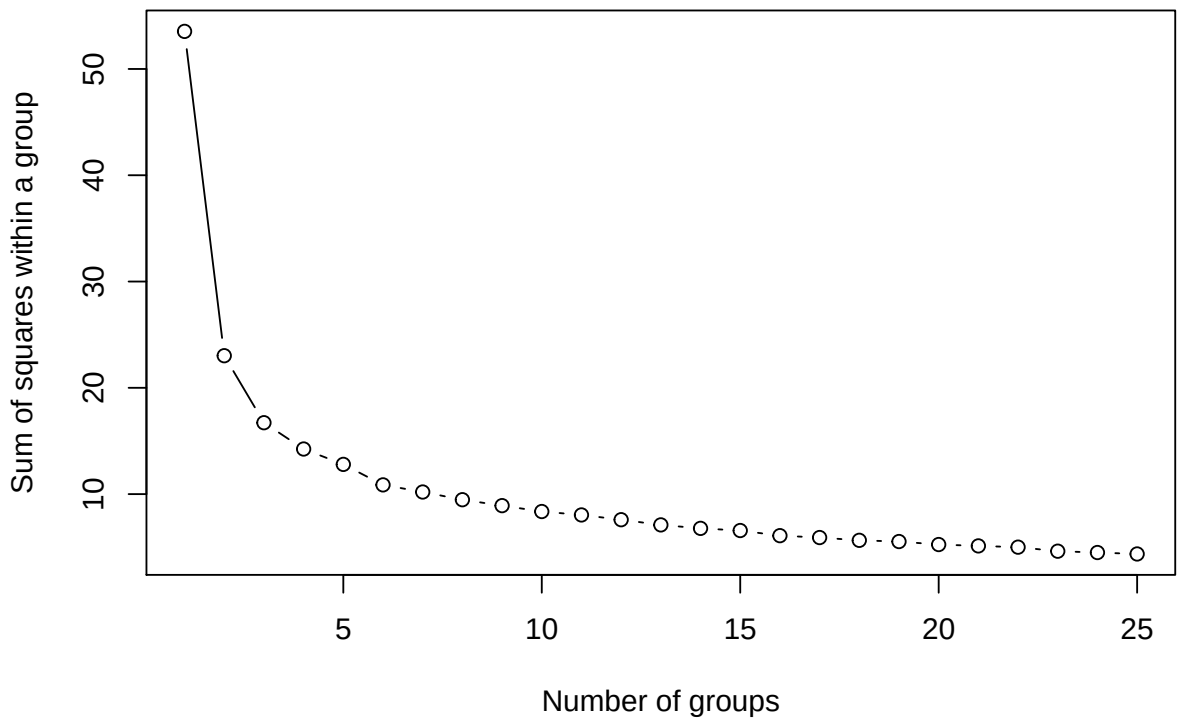
```
## K-means clustering with 3 clusters of sizes 64, 48, 44
##
## Cluster means:
##   GDPpercapita Socialsupport Healthylifeexpectancy Freedomtomakelifechoices
## 1    0.9368281     1.2612500             0.7605000                 0.3839375
## 2    1.3341875     1.4569375             0.9596458                 0.4763542
## 3    0.3910227     0.8618636             0.4182500                 0.3137273
##   Generosity Perceptionsofcorruption
## 1  0.1573750              0.07090625
## 2  0.2004375              0.17395833
## 3  0.2077955              0.09922727
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 1 2 2 1 2 2
##  [38] 2 2 2 1 2 1 2 1 1 1 1 2 1 2 1 2 2 2 1 1 2 1 2 1 1 1 2 1 2 3 1 1 1 1 1 1 3
##  [75] 1 2 1 1 1 1 1 1 1 1 3 1 1 1 1 1 1 1 1 1 3 1 3 3 3 1 3 3 1 1 1 1 1 1 1 3
## [112] 3 1 3 3 1 1 3 1 3 3 3 3 1 3 1 3 3 3 1 1 3 1 3 3 3 1 3 3 3 3 3 3 3 3 3 1
## [149] 3 3 3 3 3 3 3 3
##
## Within cluster sum of squares by cluster:
## [1] 6.545869 3.261740 6.909874
##  (between_SS / total_SS =  68.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Within Sum of Squares(WSS) Plot:

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)
```



K-Means

By Analysing the chart from right to left, we can see that when the number of groups (K) reduces from 4 to 3 there is a big increase in the sum of squares. That means that when it passes from 4 to 3 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 = 3) we had 68.8% of well-grouped data. Using 4 groups (K = 4) that value raised to 78.3%, which is a good value for us.

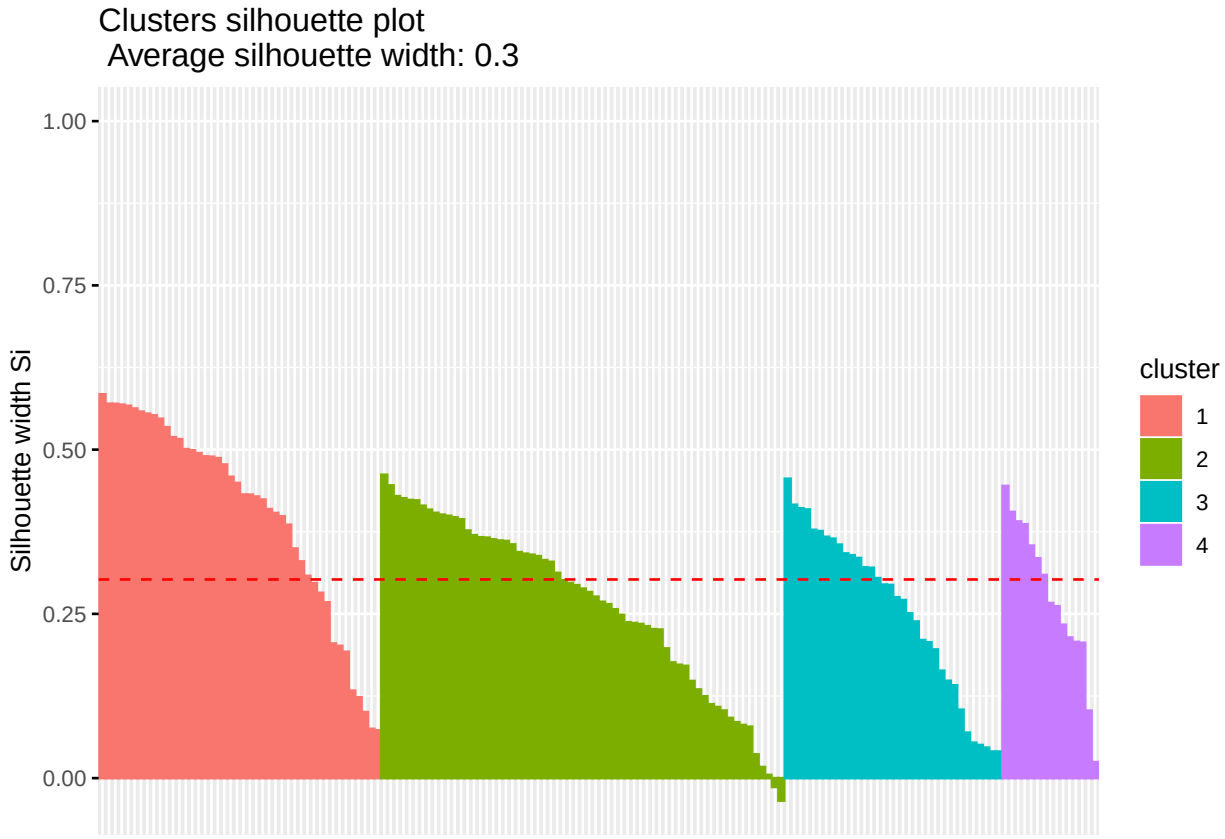
```
set.seed(1)
clustering <- kmeans(input, centers = 4, nstart = 20)
clustering
```

```
## 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 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"
```

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

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

```
##   cluster size ave.sil.width
## 1       1   44          0.40
## 2       2   63          0.26
## 3       3   34          0.25
## 4       4   15          0.28
```



5. **Conclusion:**

k-means is a pretty good clustering algorithm. But, it has some drawbacks. The biggest disadvantage is that it requires us to pre-specify the number of clusters (k). Hierarchical clustering is an alternative approach that does not require a particular choice of clusters. An additional disadvantage of k-means is that it is sensitive to outliers and different results can occur if you change the ordering of the data.K-means requires a possibly large amount of memory to store the data, and each request involves starting the identification of a local model from scratch.