500 Project - World Happiness Score

Group 3 - Akhila Saineni, Nihar Garlapati, Pranitha Chandra

4/12/2020

R Markdown

```
#Loading required libraries
library(readr)
library(DataExplorer)
library(corrplot)

## corrplot 0.84 loaded

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(Metrics)

##
## Attaching package: 'Metrics'

## The following objects are masked from 'package:caret':
##
##     precision, recall

library(ggplot2)

#Loading the dataset
whs_data=read_csv("500_project.csv")

## Parsed with column specification:
## cols(
##   `Overall rank` = col_double(),
##   `Country or region` = col_character(),
##   Score = col_double(),
##   `GDP per capita` = col_double(),
##   `Social support` = col_double(),
##   `Healthy life expectancy` = col_double(),
##   `Freedom to make life choices` = col_double(),
##   Generosity = col_double(),
##   `Perceptions of corruption` = col_double(),
##   Region = col_character(),
##   Continent = col_character()
## )
```

```r
#Renaming long column names
names(whs_data)
```

```
##  [1] "Overall rank"                "Country or region"
##  [3] "Score"                       "GDP per capita"
##  [5] "Social support"              "Healthy life expectancy"
##  [7] "Freedom to make life choices" "Generosity"
##  [9] "Perceptions of corruption"    "Region"
## [11] "Continent"
```

```r
colnames(whs_data) = c("rank", "country", "score", "gdp", "socialsup",
"lifexp", "freedom",
                       "generosity", "corruption", "region", "continent")

#Viewing data
str(whs_data)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 156 obs. of  11
variables:
##  $ rank      : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ country   : chr  "Finland" "Denmark" "Norway" "Iceland" ...
##  $ score     : num  7.77 7.6 7.55 7.49 7.49 ...
##  $ gdp       : num  1.34 1.38 1.49 1.38 1.4 ...
##  $ socialsup : num  1.59 1.57 1.58 1.62 1.52 ...
##  $ lifexp    : num  0.986 0.996 1.028 1.026 0.999 ...
##  $ freedom   : 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 ...
##  $ corruption: num  0.393 0.41 0.341 0.118 0.298 0.343 0.373 0.38 0.308
0.226 ...
##  $ region    : chr  "Western Europe" "Western Europe" "Western Europe"
"Western Europe" ...
##  $ continent : chr  "Europe" "Europe" "Europe" "Europe" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `Overall rank` = col_double(),
##   ..   `Country or region` = col_character(),
##   ..   Score = col_double(),
##   ..   `GDP per capita` = col_double(),
##   ..   `Social support` = col_double(),
##   ..   `Healthy life expectancy` = col_double(),
##   ..   `Freedom to make life choices` = col_double(),
##   ..   Generosity = col_double(),
##   ..   `Perceptions of corruption` = col_double(),
##   ..   Region = col_character(),
##   ..   Continent = col_character()
##   .. )
```

```r
summary(whs_data)
```

```
##       rank          country              score            gdp        
##  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  
##    socialsup         lifexp          freedom         generosity      
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:1.056   1st Qu.:0.5477   1st Qu.:0.3080   1st Qu.:0.1087  
##  Median :1.272   Median :0.7890   Median :0.4170   Median :0.1775  
##  Mean   :1.209   Mean   :0.7252   Mean   :0.3926   Mean   :0.1848  
##  3rd Qu.:1.452   3rd Qu.:0.8818   3rd Qu.:0.5072   3rd Qu.:0.2482  
##  Max.   :1.624   Max.   :1.1410   Max.   :0.6310   Max.   :0.5660  
##    corruption        region            continent        
##  Min.   :0.0000   Length:156         Length:156         
##  1st Qu.:0.0470   Class :character   Class :character   
##  Median :0.0855   Mode  :character   Mode  :character   
##  Mean   :0.1106                                         
##  3rd Qu.:0.1412                                         
##  Max.   :0.4530
```

```
head(whs_data)
```

```
## # A tibble: 6 x 11
##    rank country score   gdp socialsup lifexp freedom generosity corruption
##   <dbl> <chr>   <dbl> <dbl>     <dbl>  <dbl>   <dbl>      <dbl>      <dbl>
## 1     1 Finland  7.77  1.34      1.59  0.986   0.596      0.153      0.393
## 2     2 Denmark  7.6   1.38      1.57  0.996   0.592      0.252      0.41 
## 3     3 Norway   7.55  1.49      1.58  1.03    0.603      0.271      0.341
## 4     4 Iceland  7.49  1.38      1.62  1.03    0.591      0.354      0.118
## 5     5 Nether…  7.49  1.40      1.52  0.999   0.557      0.322      0.298
## 6     6 Switze…  7.48  1.45      1.53  1.05    0.572      0.263      0.343
## # … with 2 more variables: region <chr>, continent <chr>
```

Data Screening

```
#Checking Data Accuracy and Missing values
#Data looks accurate and with no missing values
summary(whs_data)
```

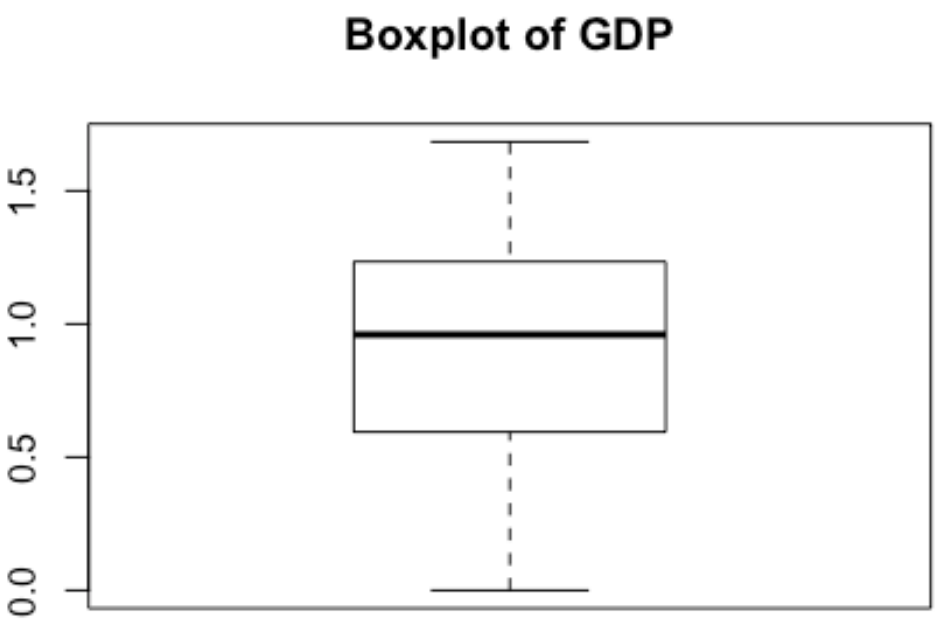
```
##       rank          country              score            gdp        
##  Min.   :  1.00   Length:156         Min.   :2.853   Min.   :0.0000  
##  1st Qu.: 39.75   Class :character   1st Qu.:4.545   1st Qu.:0.6028  
##  Median : 78.50   Mode  :character   Median :5.380   Median :0.9600  
##  Mean   : 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  
##    socialsup         lifexp          freedom         generosity      
##  Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:1.056   1st Qu.:0.5477   1st Qu.:0.3080   1st Qu.:0.1087
```

```
##  Median :1.272   Median :0.7890   Median :0.4170   Median :0.1775
##  Mean   :1.209   Mean   :0.7252   Mean   :0.3926   Mean   :0.1848
##  3rd Qu.:1.452   3rd Qu.:0.8818   3rd Qu.:0.5072   3rd Qu.:0.2482
##  Max.   :1.624   Max.   :1.1410   Max.   :0.6310   Max.   :0.5660
##    corruption        region           continent
##  Min.   :0.0000   Length:156         Length:156
##  1st Qu.:0.0470   Class :character   Class :character
##  Median :0.0855   Mode  :character   Mode  :character
##  Mean   :0.1106
##  3rd Qu.:0.1412
##  Max.   :0.4530
```

```
apply(whs_data,2,function(x) sum(is.na(x)))
```

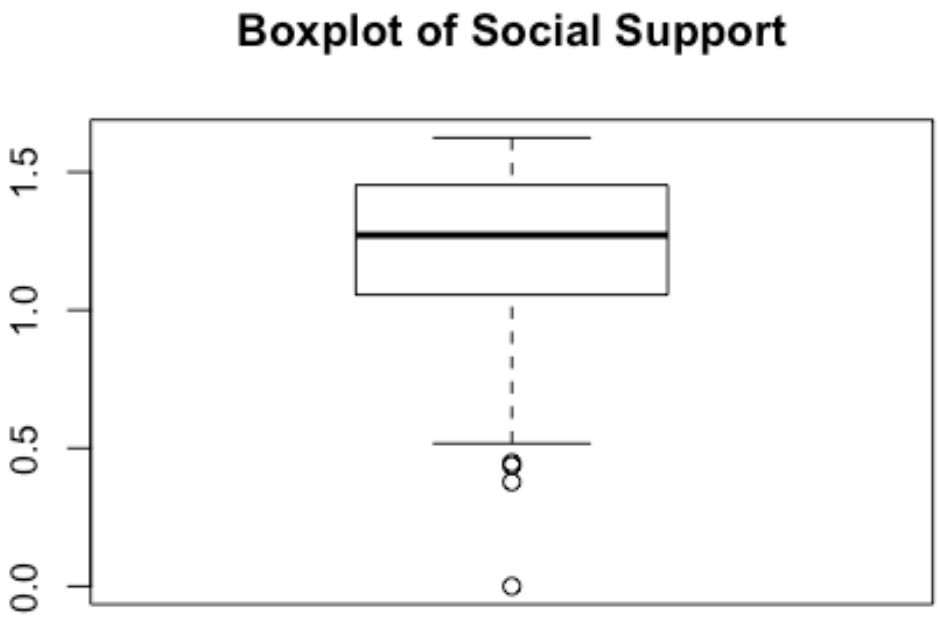
```
##       rank    country      score        gdp  socialsup     lifexp
freedom
##          0          0          0          0          0          0
0
## generosity corruption     region  continent
##          0          0          0          0
```

```
#Checking for outliers through boxplots and mahalanobis method
boxplot(whs_data$gdp, main = "Boxplot of GDP")$out
```



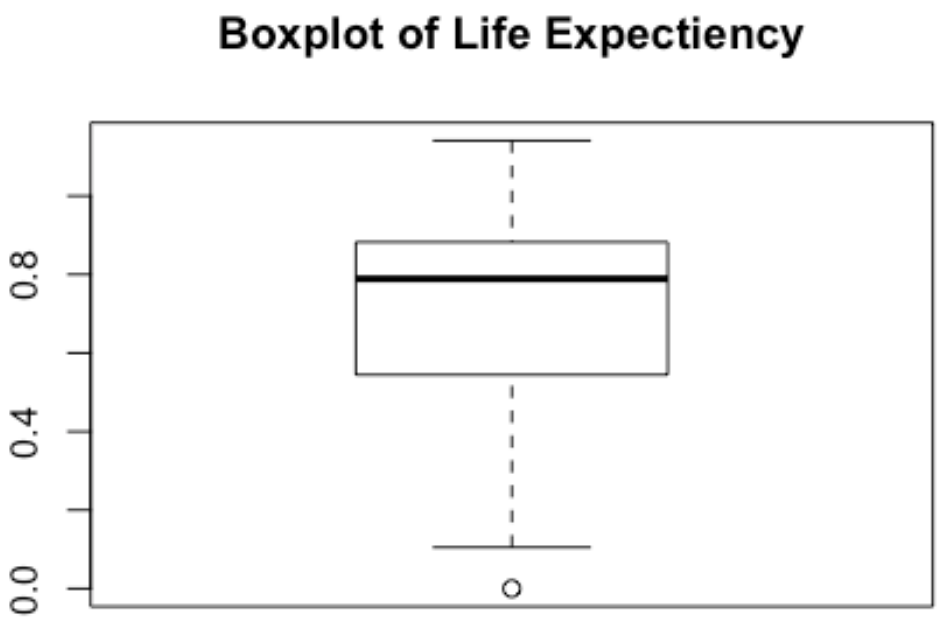
```
## numeric(0)
```

```
boxplot(whs_data$socialsup,main = "Boxplot of Social Support")$out
```



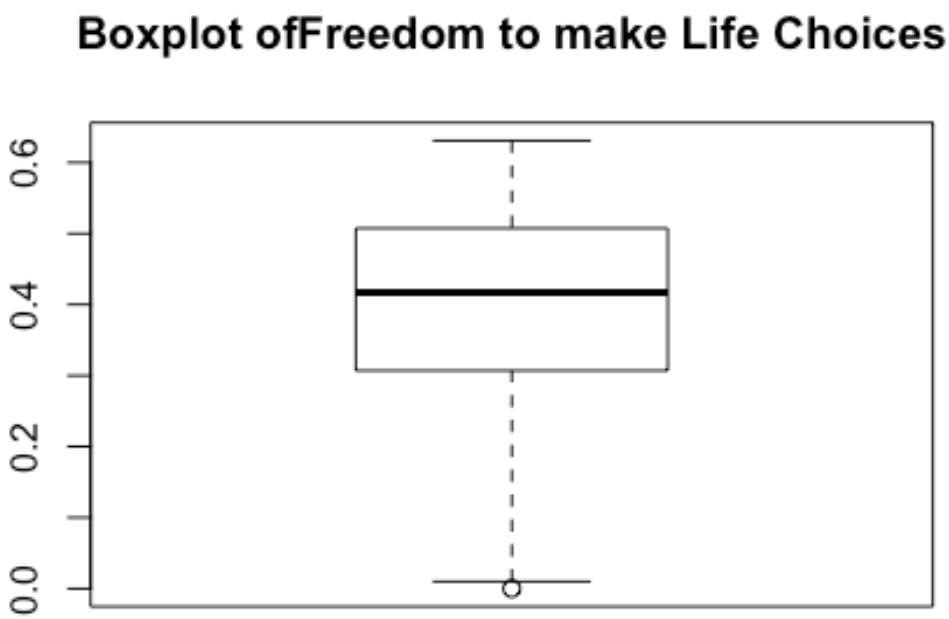

```
## [1] 0.437 0.447 0.378 0.000
```

```
boxplot(whs_data$lifexp,main = "Boxplot of Life Expectiency")$out
```



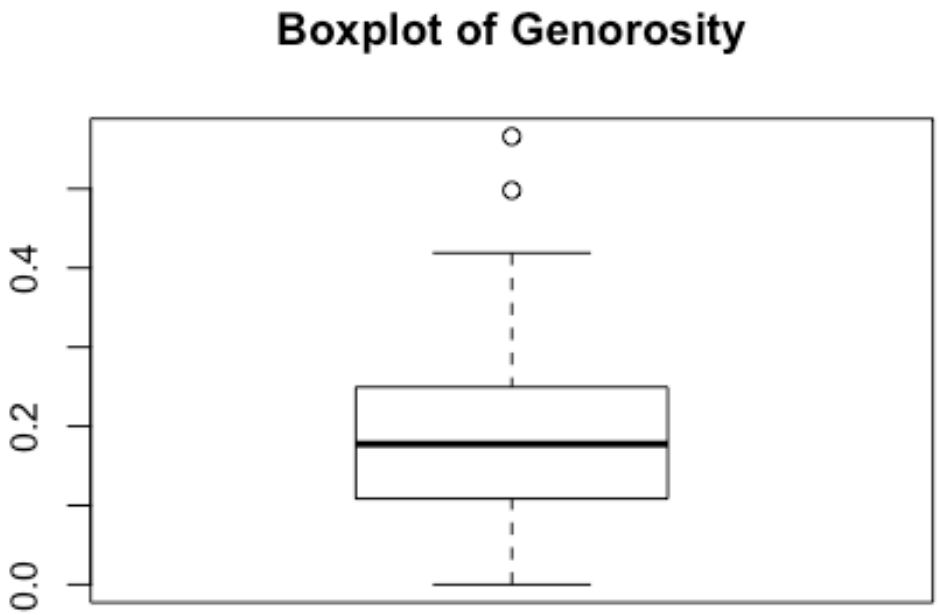

```
## [1] 0
```

```
boxplot(whs_data$freedom,main = "Boxplot ofFreedom to make Life Choices")$out
```



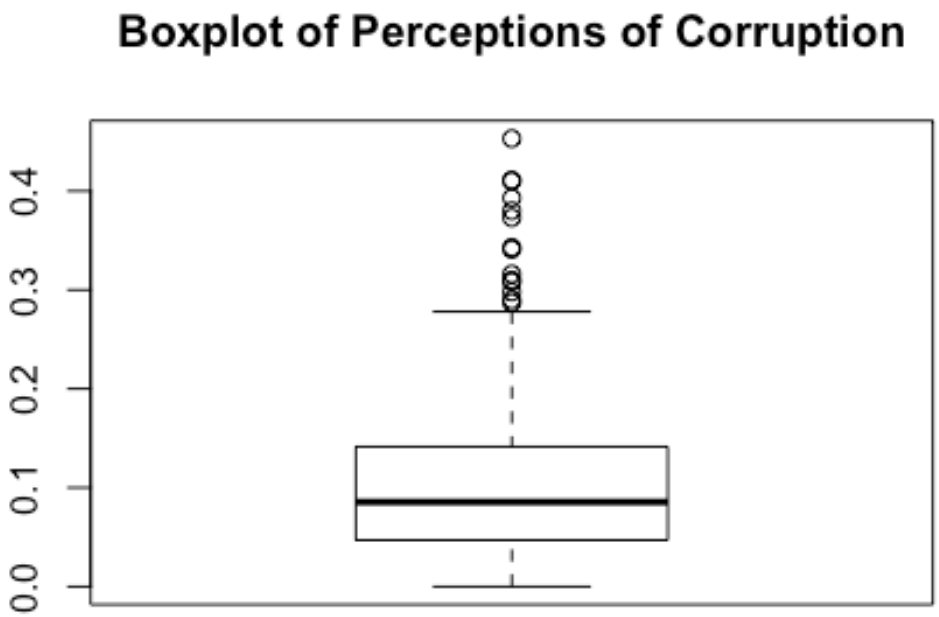

```
## [1] 0
```

```
boxplot(whs_data$generosity,main = "Boxplot of Genorosity")$out
```




```
## [1] 0.498 0.566
```

```
boxplot(whs_data$corruption,main = "Boxplot of Perceptions of
Corruption")$out
```




```
##  [1] 0.393 0.410 0.341 0.298 0.343 0.373 0.380 0.308 0.290 0.316 0.310
0.453
## [13] 0.287 0.411
```

```
outliers_mahal = mahalanobis(whs_data[ , 3:9],
                             colMeans(whs_data[, 3:9],na.rm=TRUE),
                             cov(whs_data[, 3:9], use
="pairwise.complete.obs")
)
```

```
outliers_mahal
```

```
##   [1] 13.875650 11.717751  7.311680  7.942363  6.622781  7.203125
8.465063
##   [8]  9.680386  5.541082  3.479421  5.831844  6.946783  6.658977
6.920947
##  [15]  7.186878  6.313328  3.860887  2.634448  5.196998  5.585212
8.257908
##  [22]  6.148742  4.475091  2.941282  3.617651  2.808620  4.991686
4.977473
##  [29] 10.664646  2.979876  2.981848  2.756804  1.950151 17.956749
5.111822
##  [36]  5.883963  3.920546  3.154615  5.524018  2.270188  8.598671
```

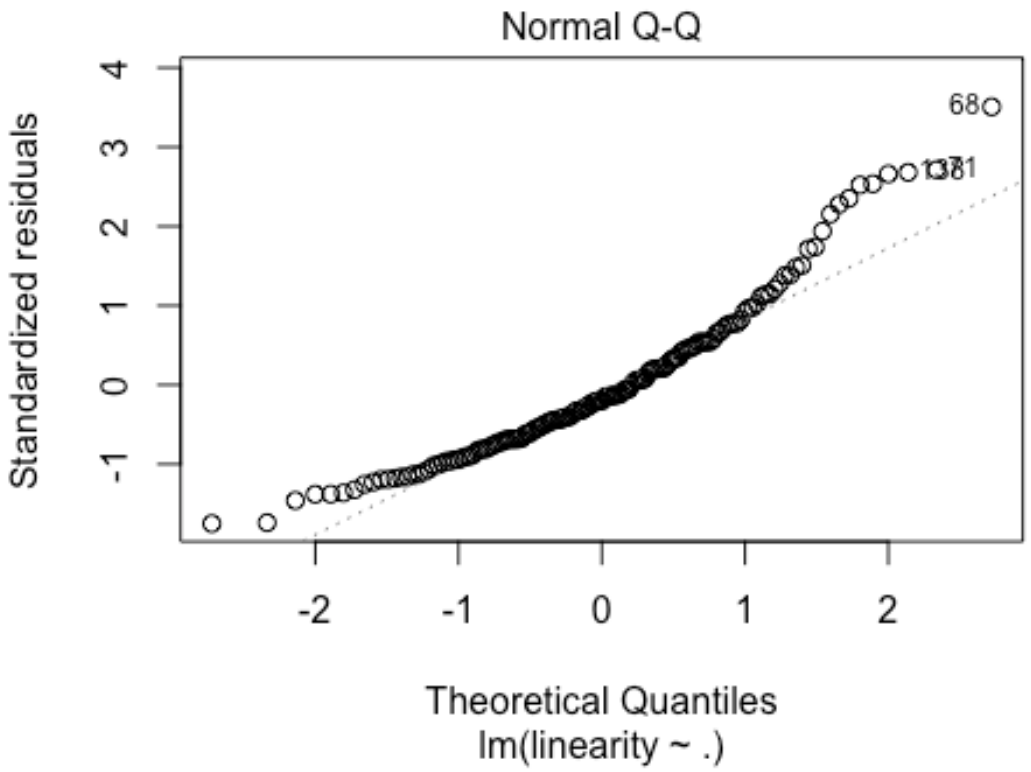
```
4.673255
##  [43]  3.109180  3.976566  4.736193  5.439580  3.138373  5.181828
5.424125
##  [50]  2.770468  6.791503  8.218622  3.619777  8.535204  3.261604
5.010226
##  [57]  2.046257  5.184385  5.471004  2.675849  3.004694  4.660716
2.962133
##  [64]  3.791225  3.313761  6.750506  6.025803  3.540949  3.664064
2.236702
##  [71]  5.380814  1.771324  3.901215  4.603893  2.937112 12.157472
2.515524
##  [78]  6.180791  4.663135  8.087960  5.590897 11.782507  5.791023
1.730364
##  [85] 11.476134  9.258723  6.644446  7.552361 10.476865  4.881353
2.859069
##  [92] 17.463069  7.077343  6.391385  6.232616  6.288967  5.385005
4.029920
##  [99] 10.096158  5.469934  2.993879 13.998130  4.406960  5.308424
4.084950
## [106]  7.116745  8.963606  7.669688  7.078406  3.708973  2.851168
15.384862
## [113]  6.636828  6.505145  5.143956  3.628908  8.994343  2.987649
11.978535
## [120]  3.848869  6.455180  8.998843  6.200411  5.834219  7.057191
5.512386
## [127]  8.910293  4.909341  5.740631  8.874602 21.918565  8.872984
8.211488
## [134]  3.194929 30.086302  3.618117  4.099736  4.038862  5.424252
8.799890
## [141]  6.389326  6.766637  5.883393 13.019825 10.973873  6.096734
17.646061
## [148] 19.087525 20.458123  9.821408 11.522710 28.565277  9.102803
9.548892
## [155] 20.937252 10.292037

cutoff = qchisq(1 - .001, ncol(whs_data[ , 3:9]))
print(cutoff)

## [1] 24.32189

summary(outliers_mahal < cutoff)

##    Mode   FALSE    TRUE
## logical       2     154

#Output gives two outliers

#Dataset with only outliers
whs_outliers = subset(whs_data, outliers_mahal > cutoff)
#Dataset without outliers
whs_nooutliers = subset(whs_data, outliers_mahal < cutoff)
```

Data Assumptions

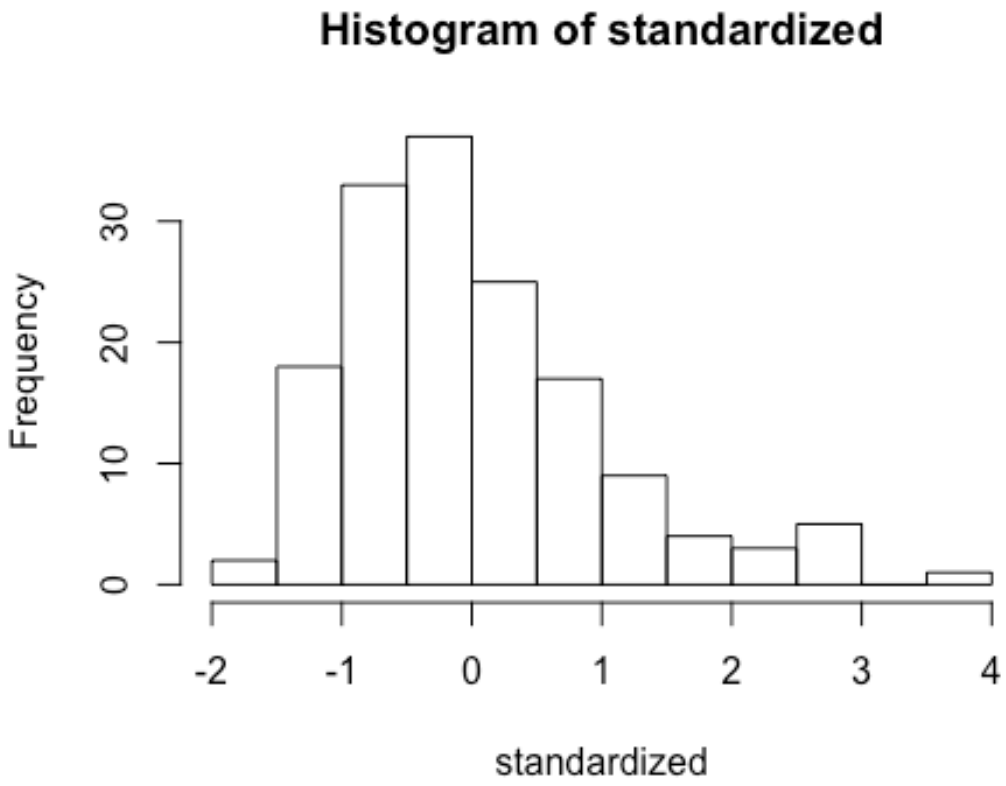
```r
#Linearity
linearity = rchisq(nrow(whs_nooutliers[ , 3:9]), 7)
model = lm(linearity~., data = whs_nooutliers[ , 3:9])
summary(model)
```

```
##
## Call:
## lm(formula = linearity ~ ., data = whs_nooutliers[, 3:9])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4985 -2.5559 -0.7149  1.9370 13.0021
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.2814     1.8855   2.801  0.00578 **
## score         0.4084     0.5981   0.683  0.49576
## gdp          -1.9929     1.6625  -1.199  0.23256
## socialsup    -0.5775     1.8098  -0.319  0.75012
## lifexp        1.0724     2.7142   0.395  0.69335
## freedom       4.7299     2.8171   1.679  0.09529 .
## generosity    0.7596     3.5673   0.213  0.83167
## corruption   -8.7039     4.1012  -2.122  0.03550 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.77 on 146 degrees of freedom
## Multiple R-squared:  0.05934,    Adjusted R-squared:  0.01424
## F-statistic: 1.316 on 7 and 146 DF,  p-value: 0.2468
```

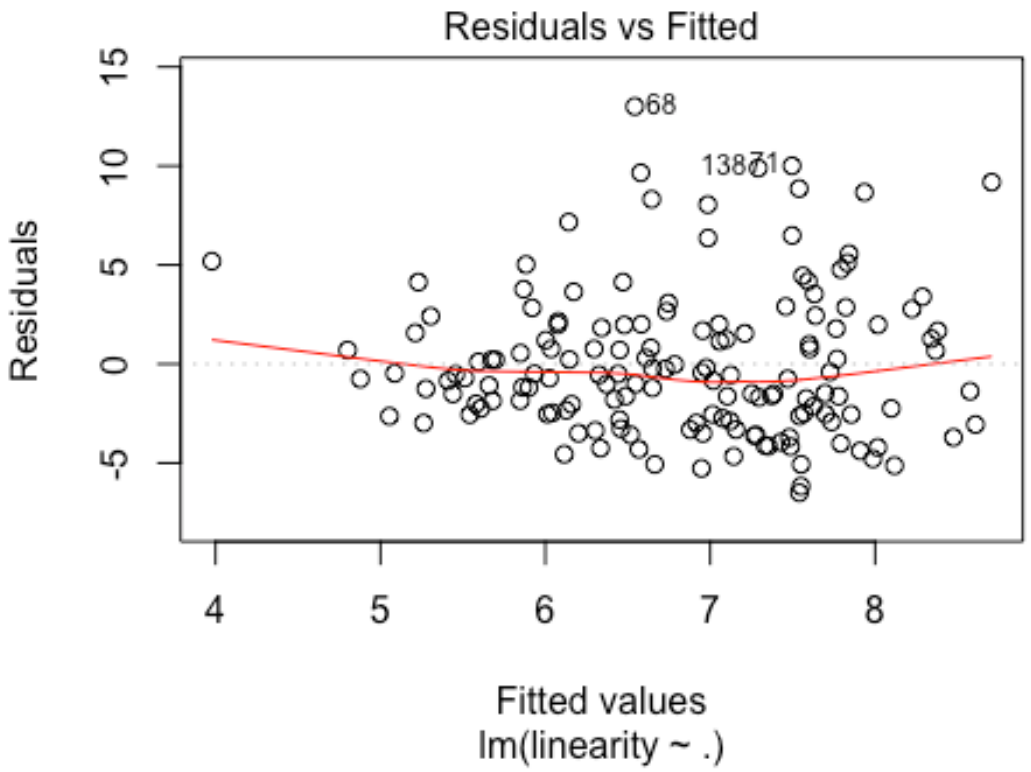
```r
plot(model,2)
```




```r
#Normality
standardized = rstudent(model)
hist(standardized, breaks = 15)
```




```r
#Homogeneity and Homoscedasticity
plot(model, 1)
```



Data Assumptions

Linearity - From the plot, it looks like most of the data lies on the line and all or most of the values are centered around zero lie on the line whereas the points at the tails deviate away from the tail. Hence, meeting the assumption of linearity.

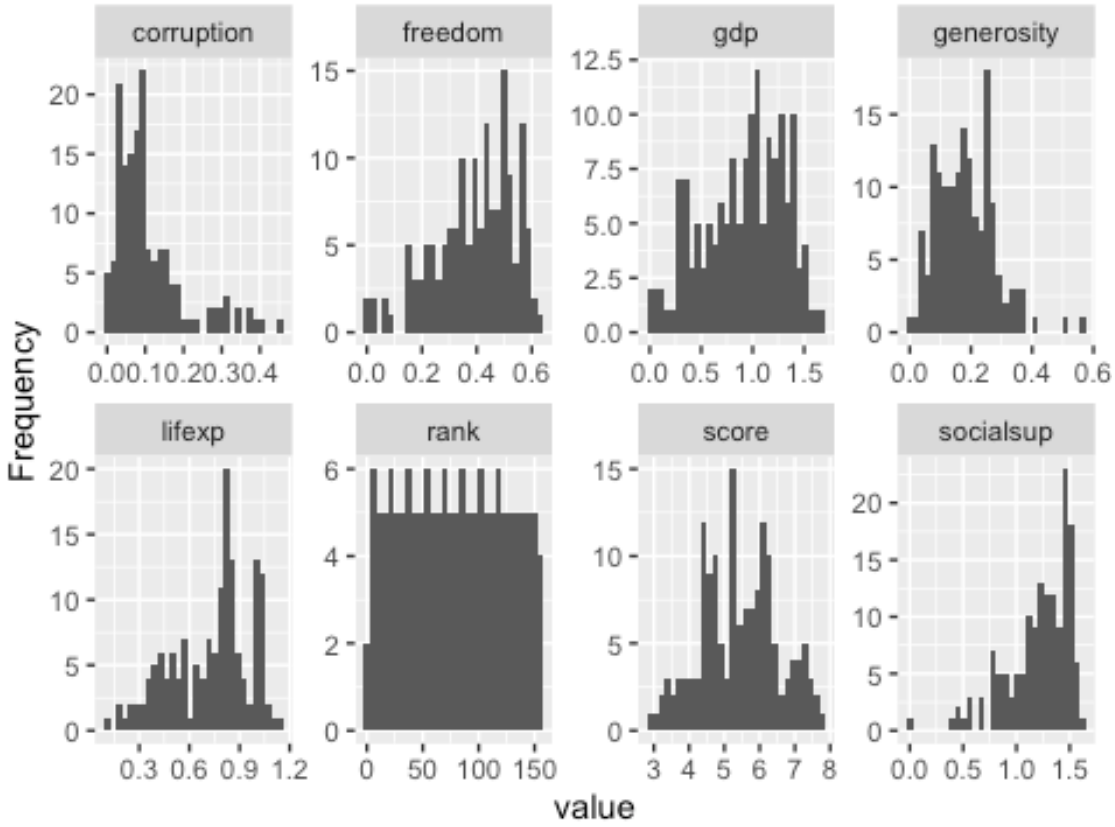
Normality - Considering the standardized histogram for the whole dataset of iris, it can be said that the distribution is concentrated with values centered around 0 between -1 and +1 but the spread doesn't seem to be even around zero since the x-axis ranges from -2 to 0 to 4. Hence, the assumption of normality is not met.

Homogeneity - The spread above the line is same as that below 0,0 line in both the directions, the points look random meeting the assumption of homogeneity.

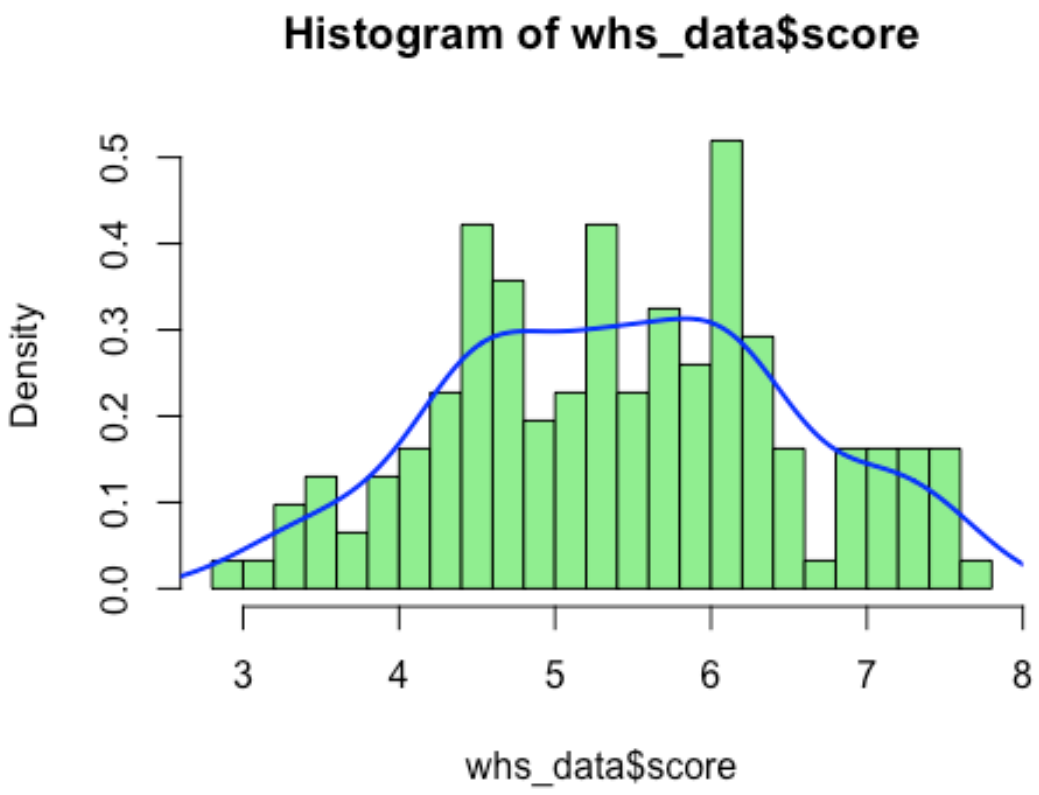
Homoscedasticity - The spread looks equal all the way across x-axis. The dots look like a bunch of random dots and do not form lumps or identified shapes hence meeting the assumption of homoscedasticity.

Exploratory Data Analysis

```
whs_data = whs_nooutliers

#Distribution of dataset
plot_histogram(whs_data)
```



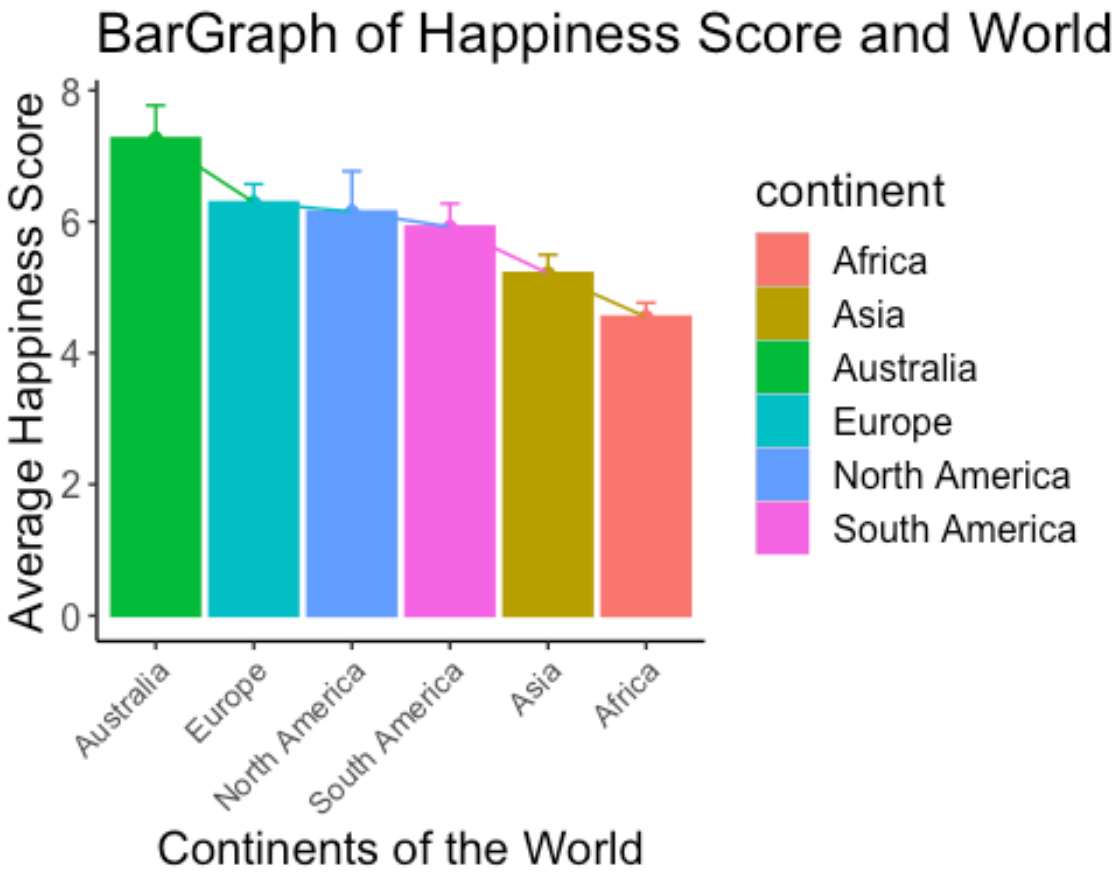

```r
#Distribution of variable score
hist(whs_data$score, breaks = 25, probability = T, col = "lightgreen")
lines(density(whs_data$score), col = "blue", lwd = 2)
```



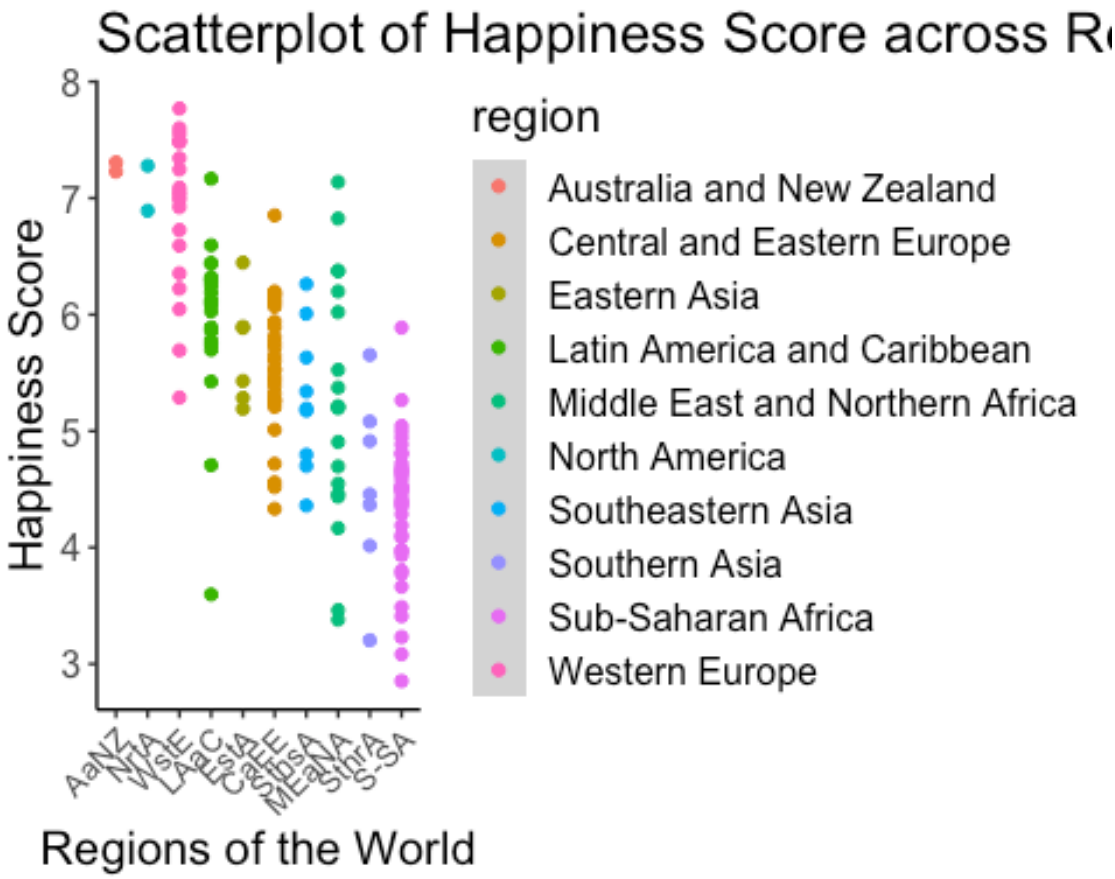

```
#The distribution of variable score looks multimodal, platykurtic and
#slightly negatively skewed.

table(is.na(whs_data))

##
## FALSE
##  1694

#there are no missing values

#Plot of Average Score Vs Continent
cleanup = theme(panel.grid.major = element_blank(),
                panel.grid.minor = element_blank(),
                panel.background = element_blank(),
                axis.line.x = element_line(color = 'black'),
                axis.line.y = element_line(color = 'black'),
                legend.key = element_rect(fill = 'lightgrey'),
                text = element_text(size = 15),
                axis.text.x = element_text(size  = 10, angle = 45, hjust = 1,
vjust = 1))

bargraph = ggplot(whs_data, aes(reorder(continent, -score), score,
```

```
                                    color = continent, fill = continent))

bargraph +
  stat_summary(fun.y = mean, ##adds the points
               geom = "point") +
  stat_summary(fun.y = mean, ##adds the line
               geom = "bar",
               aes(group=1)) +
  stat_summary(fun.y = mean, ##adds the line
               geom = "line",
               aes(group=1)) +
  stat_summary(fun.data = mean_cl_normal, ##adds the error bars
               geom = "errorbar",
               width = .2) +
  xlab("Continents of the World") +
  ylab("Average Happiness Score") +
  labs(title = "BarGraph of Happiness Score and World Continents") +
  cleanup
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: `fun.y` is deprecated. Use `fun` instead.

## Warning: `fun.y` is deprecated. Use `fun` instead.
```



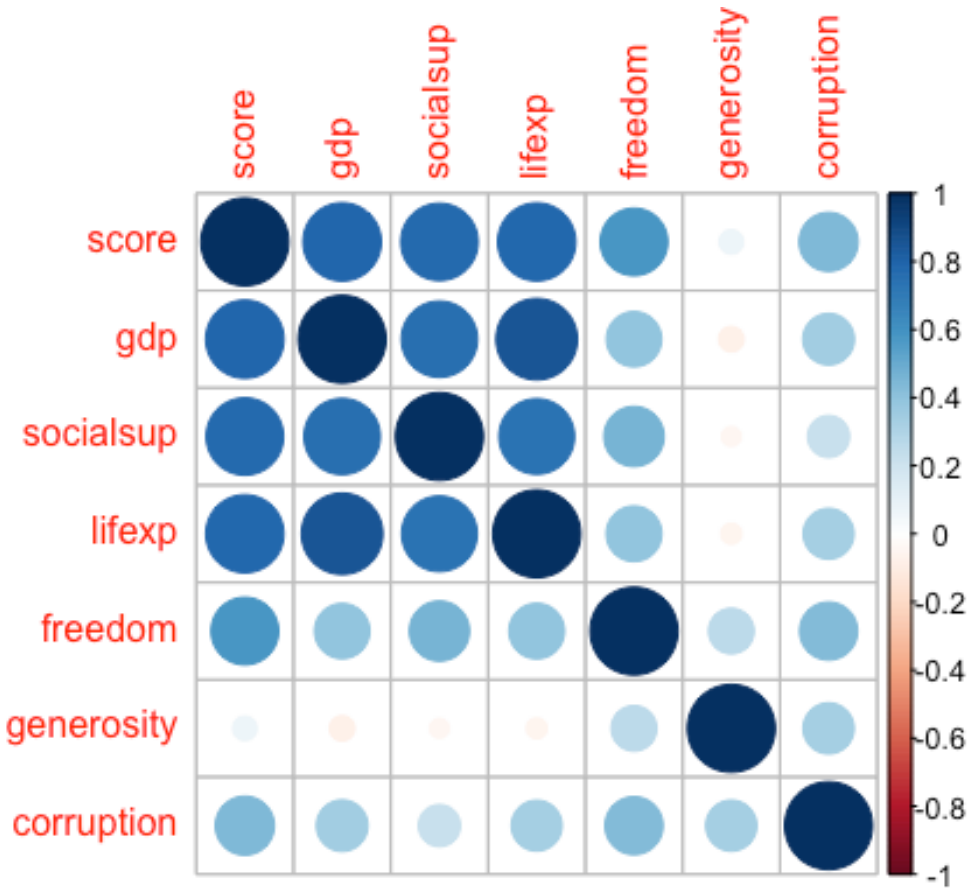

```
#the bargraph shows that on an average the happiness score for Australia is
#highest whereas happiness score for Asia and Africa is least of all.

# Plot of  Score Vs Region
scatterplot_1 = ggplot(whs_data, aes(reorder(region, -score), score,
                      fill = region,
                      color = region))

scatterplot_1 +
  scale_x_discrete(labels = abbreviate) +
  geom_point() +
  xlab("Regions of the World") +
  ylab("Happiness Score") +
  labs(title = "Scatterplot of Happiness Score across Regions") +
  cleanup
```



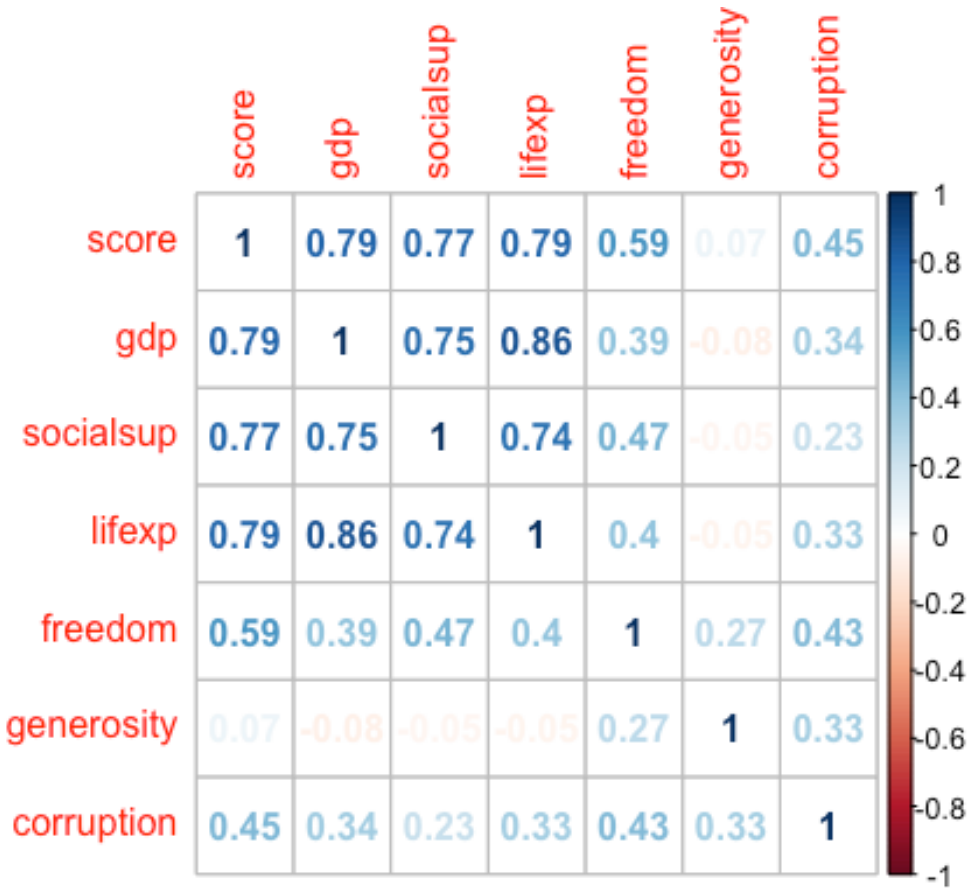

```
#the scatter plot shows the distribution of the happiness score across
various regions
```

Correlational Analysis

```
#Plotting Correlation of response variable with explanatory varibales.
#par(mfrow=c(1,1))
#corrplot(cor(whs_data[ , -c(2,10:11)]))
#symnum(cor(whs_data[ , -2]))

#Finding the correlation without the rank variable and categorical variables
#par(mfrow=c(1,1))
corrplot(cor(whs_data[ , c(-1, -2, -10, -11)]))
```

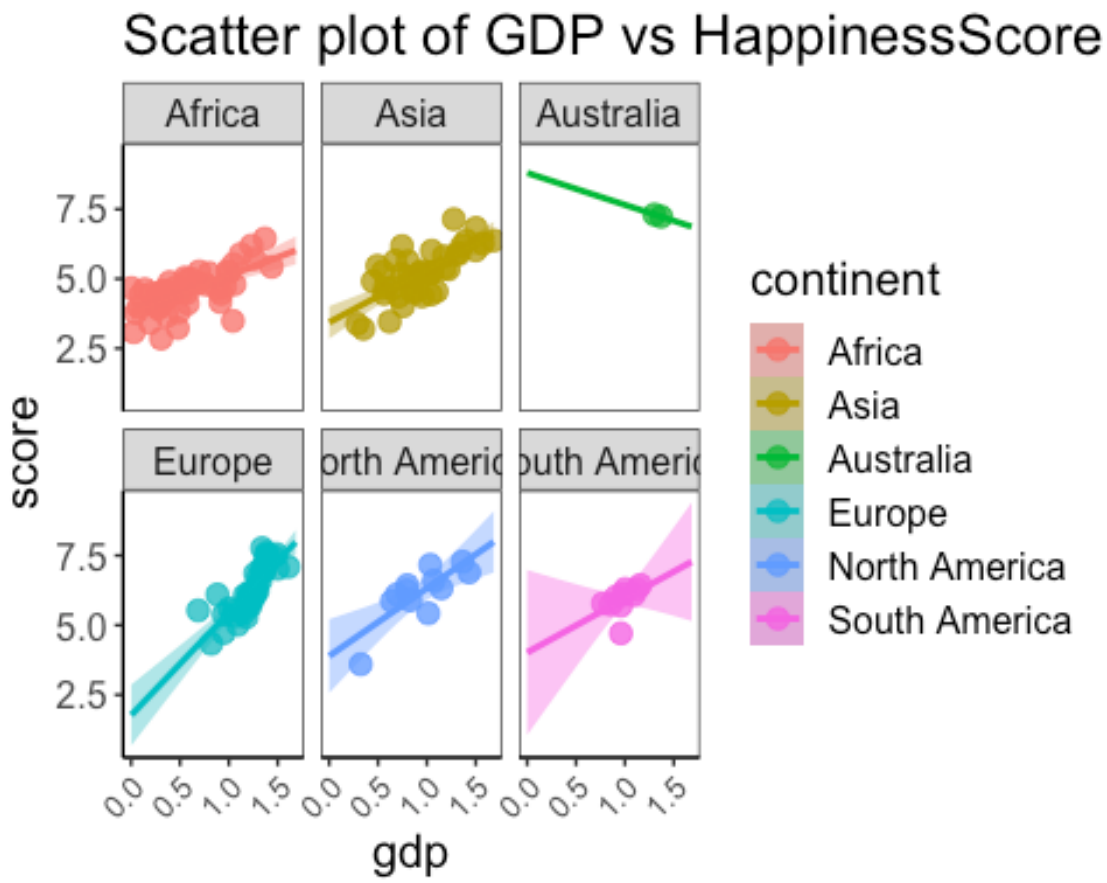



```
corrplot(cor(whs_data[ , c(-1, -2, -10, -11)]), method = "number")
```




```
#symnum(cor(whs_data[ , c(-1, -2)]))
#Looks like all the numeric variables are positively correlated with score.

#Scatter Plots with Regression Line
scatterplot_2 =
  ggplot(whs_data,
         aes(x = gdp, y = score)) +
  geom_point(aes(color=continent),
             size = 3,
             alpha = 0.8) +
  geom_smooth(aes(color = continent,
                  fill = continent),
              method = "lm",
              fullrange = TRUE)+
  facet_wrap(~continent) +
  theme_bw() +
  labs(title = "Scatter plot of GDP vs HappinessScore for Countries with
regression line") +
  cleanup
scatterplot_2

## `geom_smooth()` using formula 'y ~ x'
```

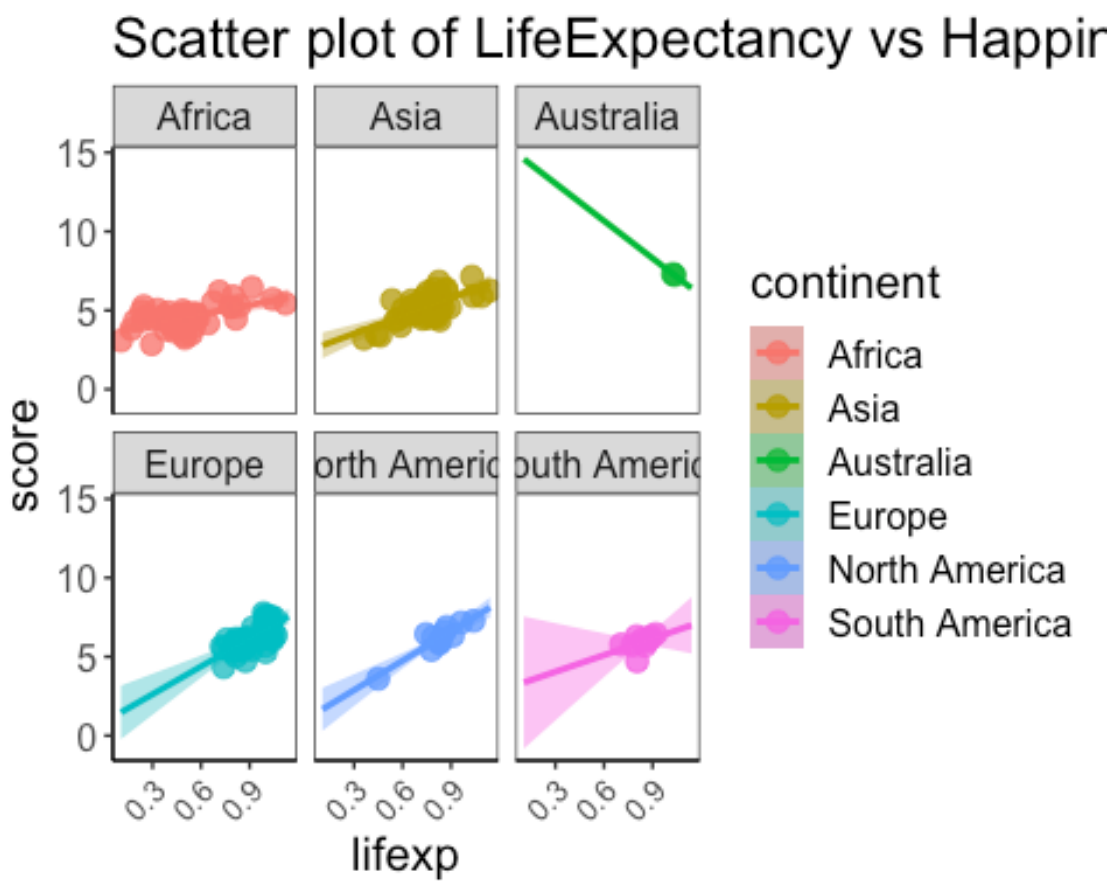
```
## Warning in qt((1 - level)/2, df): NaNs produced

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max;
returning -
## Inf
```




```
scatterplot_3 =
  ggplot(whs_data, aes(x = lifexp, y = score)) +
  geom_point(aes(color=continent),
             size = 3,
             alpha = 0.8) +
  geom_smooth(aes(color = continent,
                  fill = continent),
              method = "lm",
              fullrange = TRUE) +
  facet_wrap(~continent) +
  theme_bw() +
  labs(title = "Scatter plot of LifeExpectancy vs HappinessScore for
Countries with regression line") +
  cleanup
scatterplot_3
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning in qt((1 - level)/2, df): NaNs produced

## Warning in qt((1 - level)/2, df): no non-missing arguments to max;
returning -
## Inf
```



Analysis of Variance (ANOVA) and PostHoc (Bonferonni Correction)

```
#Initializing the dataset
whs_anova = whs_data

#Anova test
library("ez")

## Registered S3 methods overwritten by 'lme4':
##   method                           from
##   cooks.distance.influence.merMod car
##   influence.merMod                 car
##   dfbeta.influence.merMod          car
##   dfbetas.influence.merMod         car

whs_anova$no <- 1:nrow(whs_anova)

ezANOVA(data = whs_anova,
```

```
        dv = score,
        between = continent,
        wid = no,
        type = 3,
        detailed = T)

## Warning: Converting "no" to factor for ANOVA.

## Warning: Converting "continent" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified
a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## $ANOVA
##        Effect DFn DFd        SSn      SSd          F             p p<.05
## 1 (Intercept)   1 148 1663.51909 101.0663 2436.03296 8.279777e-94     *
## 2   continent   5 148   85.18892 101.0663   24.94988 3.504585e-18     *
##         ges
## 1 0.9427252
## 2 0.4573774
##
## $`Levene's Test for Homogeneity of Variance`
##   DFn DFd      SSn      SSd        F       p p<.05
## 1   5 148 2.206083 41.58687 1.570208 0.17195

ezANOVA(data = whs_anova,
        dv = score,
        between = region,
        wid = no,
        type = 3,
        detailed = T)

## Warning: Converting "no" to factor for ANOVA.

## Warning: Converting "region" to factor for ANOVA.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified
a
## well-considered value for the type argument to ezANOVA().

## Coefficient covariances computed by hccm()

## $ANOVA
##        Effect DFn DFd       SSn      SSd          F             p p<.05
## 1 (Intercept)   1 144 2050.7815 71.36009 4138.34315 5.628113e-108     *
## 2      region   9 144  114.8951 71.36009   25.76121  5.772061e-26     *
##         ges
## 1 0.9663735
## 2 0.6168693
```

```
##
## $`Levene's Test for Homogeneity of Variance`
##   DFn DFd      SSn      SSd        F         p p<.05
## 1   9 144 3.023101 30.77274 1.571833 0.1289702
```

```
#Since the p value of the anova test is less than 0.05. We can consider that
the Anova test is significant, therefore we reject the null hypothesis and
conclude that there is a difference in the average happiness score between
different continents and regions. Also from the levene's test, the p-values
are greater than 0.05 indicating that the variances are equal among both the
continents and regions.
```

```
#Bonferroni correction
post_continents = pairwise.t.test(whs_anova$score,
                        whs_data$continent,
                        p.adjust.method = "bonferroni",
                        paired = F,
                        var.equal = T)

post_continents
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  whs_anova$score and whs_data$continent
##
##               Africa  Asia    Australia Europe  North America
## Asia          0.00309 -       -         -       -
## Australia     0.00016 0.01169 -         -       -
## Europe        < 2e-16 4.0e-07 1.00000   -       -
## North America 1.9e-07 0.01075 1.00000   1.00000 -
## South America 6.2e-05 0.24975 0.55494   1.00000 1.00000
##
## P value adjustment method: bonferroni
```

```
#There is a significant difference in the score between continents with
higher number of developing nations such as asia, africa when compared to
continents with fewer developing nations such as europe or north america
```

Splitting the dataset

```
#Splitting the dataset into Train(75%) and Test(25%) data for Modelling and
Prediction
set.seed(100)
smp_size = floor(0.75 * nrow(whs_data))
train_ind = sample(seq_len(nrow(whs_data)), size = smp_size)

train_data = whs_data[train_ind, ]
test_data = whs_data[-train_ind, ]
```

Multiple Linear Regression Modelling

```r
#Building models without rank and score.
#Model1 includes all the variables as predictor variables
model_all = lm(score ~ ., data = train_data[ , c(-1, -2)])
summary(model_all)
```

```
##
## Call:
## lm(formula = score ~ ., data = train_data[, c(-1, -2)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.75950 -0.21188  0.02438  0.30210  1.14531
##
## Coefficients: (2 not defined because of singularities)
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           2.99239    0.75610   3.958 0.000145
***
## gdp                                   0.88776    0.28862   3.076 0.002734
**
## socialsup                             0.74970    0.25527   2.937 0.004150
**
## lifexp                                0.54465    0.54447   1.000 0.319663
## freedom                               1.96493    0.44222   4.443 2.37e-05
***
## generosity                            0.57821    0.57747   1.001 0.319200
## corruption                            0.24071    0.79315   0.303 0.762179
## regionCentral and Eastern Europe     -0.86647    0.58469  -1.482 0.141629
## regionEastern Asia                   -0.75280    0.57189  -1.316 0.191196
## regionLatin America and Caribbean    -0.36701    0.57451  -0.639 0.524454
## regionMiddle East and Northern Africa -0.75570    0.58131  -1.300 0.196716
## regionNorth America                  -0.22944    0.64535  -0.356 0.722972
## regionSoutheastern Asia              -1.00099    0.60387  -1.658 0.100658
## regionSouthern Asia                  -0.77968    0.64578  -1.207 0.230264
## regionSub-Saharan Africa             -0.75304    0.60622  -1.242 0.217188
## regionWestern Europe                 -0.40368    0.57704  -0.700 0.485887
## continentAsia                        -0.02748    0.23585  -0.117 0.907484
## continentAustralia                         NA         NA      NA       NA
## continentEurope                       0.32359    0.30843   1.049 0.296749
## continentNorth America                0.21319    0.22337   0.954 0.342265
## continentSouth America                     NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4776 on 96 degrees of freedom
## Multiple R-squared:  0.8482, Adjusted R-squared:  0.8197
## F-statistic: 29.79 on 18 and 96 DF,  p-value: < 2.2e-16
```

```r
prediction_all = predict(model_all, newdata = test_data)
```

```
## Warning in predict.lm(model_all, newdata = test_data): prediction from a
rank-
## deficient fit may be misleading
```

```
mean((prediction_all - test_data$score)^2)
```

```
## [1] 0.2401336
```

```
RMSE(prediction_all, test_data$score)
```

```
## [1] 0.4900343
```

```
R2(prediction_all, test_data$score)
```

```
## [1] 0.7781122
```

```
AIC(model_all)
```

```
## [1] 175.6393
```

```
BIC(model_all)
```

```
## [1] 230.538
```

```
confint(model_all)
```

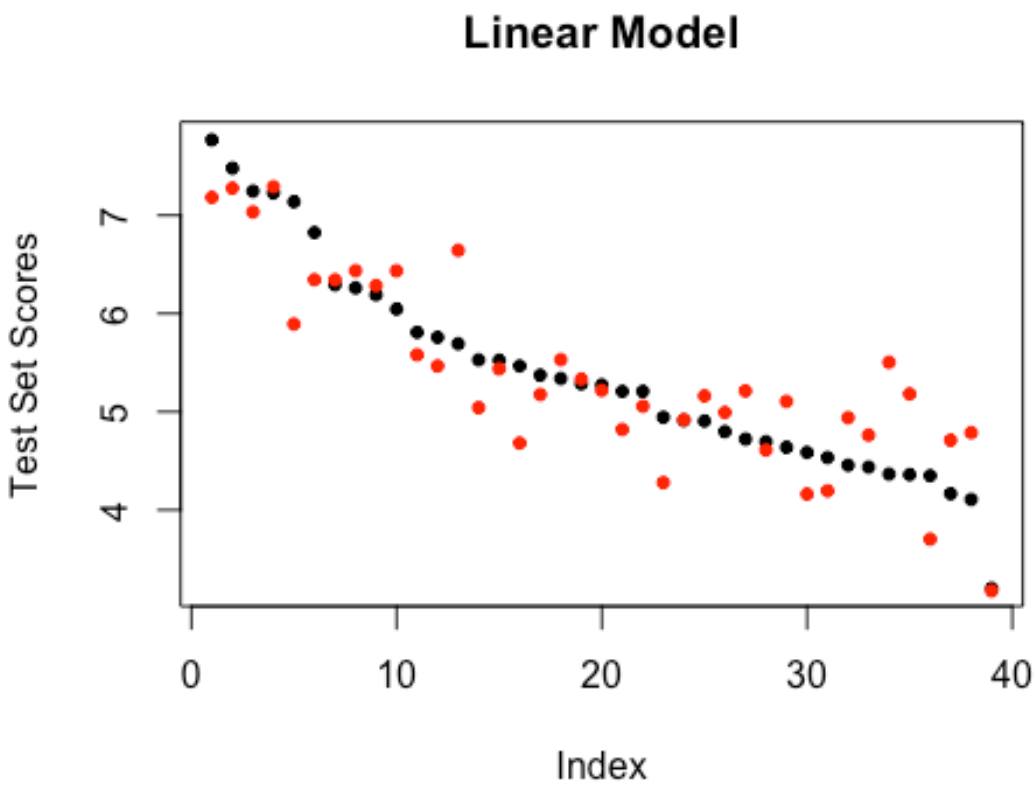
```
##                                         2.5 %    97.5 %
## (Intercept)                         1.4915447 4.4932273
## gdp                                 0.3148579 1.4606676
## socialsup                           0.2429937 1.2564037
## lifexp                             -0.5361115 1.6254164
## freedom                             1.0871299 2.8427349
## generosity                         -0.5680456 1.7244752
## corruption                         -1.3336801 1.8150936
## regionCentral and Eastern Europe   -2.0270669 0.2941184
## regionEastern Asia                 -1.8879927 0.3823952
## regionLatin America and Caribbean  -1.5074088 0.7733791
## regionMiddle East and Northern Africa -1.9096006 0.3981948
## regionNorth America                -1.5104543 1.0515705
## regionSoutheastern Asia            -2.1996604 0.1976834
## regionSouthern Asia                -2.0615321 0.5021785
## regionSub-Saharan Africa           -1.9563683 0.4502863
## regionWestern Europe               -1.5490954 0.7417343
## continentAsia                      -0.4956490 0.4406853
## continentAustralia                         NA        NA
## continentEurope                    -0.2886450 0.9358209
## continentNorth America             -0.2301978 0.6565825
## continentSouth America                     NA        NA
```

```
summary(prediction_all)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.178   4.804   5.182   5.383   6.089   7.291
```

```r
plot(test_data$score, main = "Linear Model", ylab = "Test Set Scores", pch =
20)
points(predict(model_all, newdata = test_data), col = "red", pch = 20)
```

```
## Warning in predict.lm(model_all, newdata = test_data): prediction from a
rank-
## deficient fit may be misleading
```




```r
names(train_data)
```

```
##  [1] "rank"       "country"    "score"      "gdp"        "socialsup"
##  [6] "lifexp"     "freedom"    "generosity" "corruption" "region"
## [11] "continent"
```

```r
#Running Backward Stepwise elimination
#Removing Region and Continent variable from the process as they were of low
or no importance in the model_all at all
#STEP1
model_1 = lm(score ~ socialsup+lifexp+freedom+generosity+corruption,
             data = train_data[ , c(-1, -2)])
#summary(model_1)

model_2 = lm(score ~ gdp+lifexp+freedom+generosity+corruption,
             data = train_data[ , c(-1, -2)])
```

```r
#summary(model_2)

model_3 = lm(score ~ gdp+socialsup+freedom+generosity+corruption,
             data = train_data[ , c(-1, -2)])
#summary(model_3)

model_4 = lm(score ~ gdp+socialsup+lifexp+generosity+corruption,
             data = train_data[ , c(-1, -2)])
#summary(model_4)

#************************************************************
#FINAL REGRESSION MODEL
model_5 = lm(score ~ gdp+socialsup+lifexp+freedom+corruption,
             data = train_data[ , c(-1, -2)])
summary(model_5)

##
## Call:
## lm(formula = score ~ gdp + socialsup + lifexp + freedom + corruption,
##     data = train_data[, c(-1, -2)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86526 -0.28852  0.04408  0.30062  1.07502
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8864     0.2194   8.599 6.57e-14 ***
## gdp           0.6933     0.2585   2.682 0.008452 **
## socialsup     1.0060     0.2581   3.898 0.000168 ***
## lifexp        1.1737     0.4476   2.622 0.009981 **
## freedom       1.8028     0.3915   4.605 1.12e-05 ***
## corruption    1.4151     0.6128   2.309 0.022816 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5108 on 109 degrees of freedom
## Multiple R-squared:  0.8028, Adjusted R-squared:  0.7938
## F-statistic: 88.77 on 5 and 109 DF,  p-value: < 2.2e-16

#************************************************************

model_6 = lm(score ~ gdp+socialsup+lifexp+freedom+generosity,
             data = train_data[ , c(-1, -2)])
#summary(model_6)


#STEP2
model_51 = lm(score ~ socialsup+lifexp+freedom+corruption,
              data = train_data[ , c(-1, -2)])
```

```r
#summary(model_51)

model_52 = lm(score ~ gdp+lifexp+freedom+corruption,
              data = train_data[ , c(-1, -2)])
#summary(model_52)

model_53 = lm(score ~ gdp+socialsup+freedom+corruption,
              data = train_data[ , c(-1, -2)])
#summary(model_53)

model_54 = lm(score ~ gdp+socialsup+lifexp+corruption,
              data = train_data[ , c(-1, -2)])
#summary(model_54)

model_55 = lm(score ~ gdp+socialsup+lifexp+freedom,
              data = train_data[ , c(-1, -2)])
#summary(model_55)


#STEP3
model_551 = lm(score ~ socialsup+lifexp+freedom,
              data = train_data[ , c(-1, -2)])
#summary(model_551)

model_552 = lm(score ~ gdp+freedom+corruption,
              data = train_data[ , c(-1, -2)])
#summary(model_552)

model_553 = lm(score ~ gdp+socialsup+corruption,
              data = train_data[ , c(-1, -2)])
#summary(model_553)

model_554 = lm(score ~ gdp+socialsup+lifexp,
              data = train_data[ , c(-1, -2)])
#summary(model_554)


#STEP4
model_5511 = lm(score ~ lifexp+freedom,
              data = train_data[ , c(-1, -2)])
#summary(model_5511)

model_5512 = lm(score ~ socialsup+freedom,
              data = train_data[ , c(-1, -2)])
#summary(model_5512)

model_5513 = lm(score ~ socialsup+lifexp,
              data = train_data[ , c(-1, -2)])
#summary(model_5513)
```

```r
#STEP5
model_55111 = lm(score ~ freedom,
              data = train_data[ , c(-1, -2)])
#summary(model_55111)

model_55112 = lm(score ~ lifexp,
              data = train_data[ , c(-1, -2)])
#summary(model_55112)


#Checking if the same model(final regression model) is obtained using Step
function
lm1 = lm(score ~ ., data = train_data[ , c(-1, -2, -10, -11)])
summary(lm1)
```

```
##
## Call:
## lm(formula = score ~ ., data = train_data[, c(-1, -2, -10, -11)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.78091 -0.31948  0.05906  0.30251  1.09512
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.7974     0.2419   7.431 2.67e-11 ***
## gdp           0.7128     0.2597   2.744 0.007099 **
## socialsup     1.0175     0.2587   3.933 0.000149 ***
## lifexp        1.1870     0.4483   2.648 0.009310 **
## freedom       1.7381     0.3988   4.358 3.00e-05 ***
## generosity    0.4993     0.5688   0.878 0.382023
## corruption    1.2249     0.6506   1.883 0.062446 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5113 on 108 degrees of freedom
## Multiple R-squared:  0.8042, Adjusted R-squared:  0.7934
## F-statistic: 73.95 on 6 and 108 DF,  p-value: < 2.2e-16
```

```r
#The model obtained is same as the model obtained in the backward stepwise
regression
slm1 <- step(lm1)
```

```
## Start:  AIC=-147.5
## score ~ gdp + socialsup + lifexp + freedom + generosity + corruption
##
##              Df Sum of Sq    RSS     AIC
## - generosity  1    0.2014 28.437 -148.68
## <none>                    28.235 -147.50
```

```
## - corruption  1    0.9266 29.162 -145.79
## - lifexp      1    1.8331 30.068 -142.27
## - gdp         1    1.9692 30.205 -141.75
## - socialsup   1    4.0437 32.279 -134.11
## - freedom     1    4.9658 33.201 -130.87
##
## Step:  AIC=-148.68
## score ~ gdp + socialsup + lifexp + freedom + corruption
##
##              Df Sum of Sq    RSS     AIC
## <none>                    28.437 -148.68
## - corruption  1    1.3912 29.828 -145.19
## - lifexp      1    1.7941 30.231 -143.65
## - gdp         1    1.8769 30.314 -143.33
## - socialsup   1    3.9631 32.400 -135.68
## - freedom     1    5.5317 33.968 -130.24
```

```
summary(slm1)
```

```
##
## Call:
## lm(formula = score ~ gdp + socialsup + lifexp + freedom + corruption,
##     data = train_data[, c(-1, -2, -10, -11)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86526 -0.28852  0.04408  0.30062  1.07502
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8864     0.2194   8.599 6.57e-14 ***
## gdp           0.6933     0.2585   2.682 0.008452 **
## socialsup     1.0060     0.2581   3.898 0.000168 ***
## lifexp        1.1737     0.4476   2.622 0.009981 **
## freedom       1.8028     0.3915   4.605 1.12e-05 ***
## corruption    1.4151     0.6128   2.309 0.022816 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5108 on 109 degrees of freedom
## Multiple R-squared:  0.8028, Adjusted R-squared:  0.7938
## F-statistic: 88.77 on 5 and 109 DF,  p-value: < 2.2e-16
```

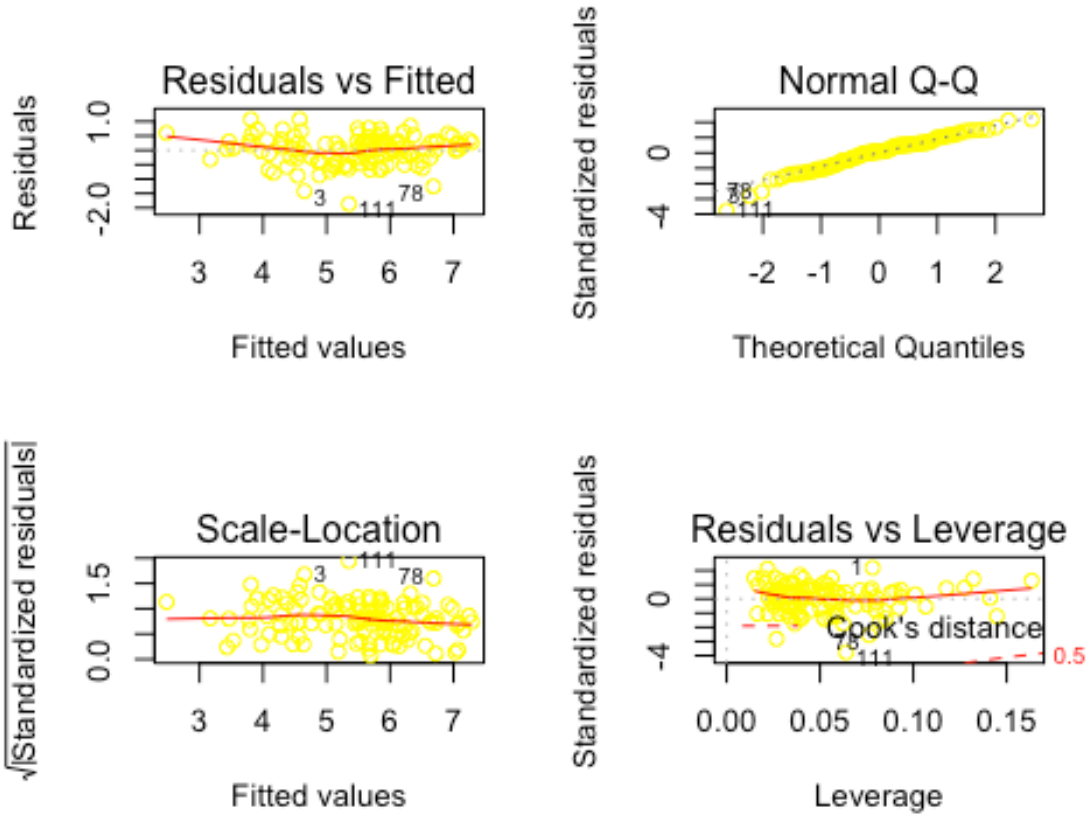
```
slm1$anova
```

```
##           Step Df Deviance Resid. Df Resid. Dev       AIC
## 1               NA       NA       108   28.23531 -147.5013
## 2 - generosity  1 0.201428       109   28.43674 -148.6838
```

```
#output model provided score ~ gdp + socialsup + lifexp + freedom +
corruption
#i.e. model without generosity
```

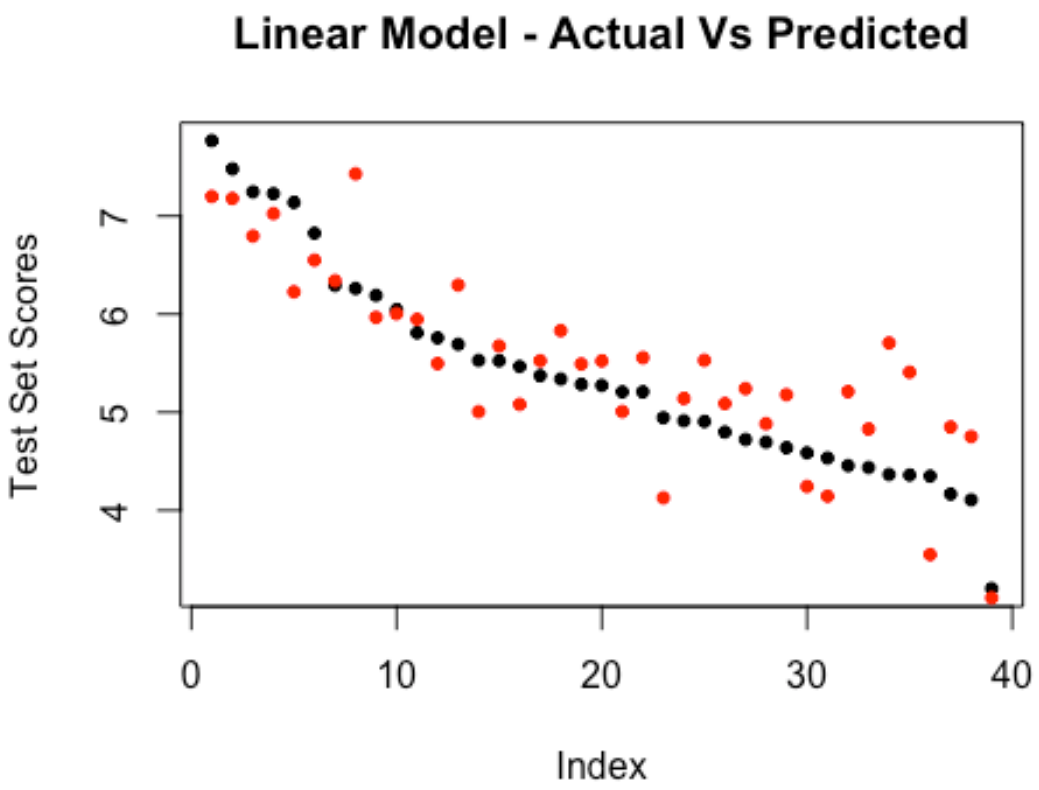
Residual plots of Model5

```
par(mfrow= c(2,2))
plot(model_5, col = "yellow")
```



Prediction Model

```
#Checking the Predictive ability of the model
plot(test_data$score, main = "Linear Model - Actual Vs Predicted", ylab =
"Test Set Scores",
     pch = 20)
prediction_model5 = predict(model_5, newdata = test_data)
points(prediction_model5, col = "red", pch = 20)
```




```
mean((prediction_model5 - test_data$score)^2)
```

```
## [1] 0.2980288
```

```
RMSE(prediction_model5, test_data$score)
```

```
## [1] 0.5459202
```

```
R2(prediction_model5, test_data$score)
```

```
## [1] 0.7360158
```

```
AIC(model_5)
```

```
## [1] 179.6721
```

```
BIC(model_5)
```

```
## [1] 198.8866
```

```
confint(model_5)
```

```
##                 2.5 %   97.5 %
## (Intercept) 1.4515936 2.321220
## gdp         0.1810053 1.205653
```

```
## socialsup   0.4944320 1.517571
## lifexp      0.2866143 2.060686
## freedom     1.0268545 2.578809
## corruption  0.2005446 2.629690
```

```
summary(model_5)
```

```
##
## Call:
## lm(formula = score ~ gdp + socialsup + lifexp + freedom + corruption,
##     data = train_data[, c(-1, -2)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86526 -0.28852  0.04408  0.30062  1.07502
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8864     0.2194   8.599 6.57e-14 ***
## gdp           0.6933     0.2585   2.682 0.008452 **
## socialsup     1.0060     0.2581   3.898 0.000168 ***
## lifexp        1.1737     0.4476   2.622 0.009981 **
## freedom       1.8028     0.3915   4.605 1.12e-05 ***
## corruption    1.4151     0.6128   2.309 0.022816 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5108 on 109 degrees of freedom
## Multiple R-squared:  0.8028, Adjusted R-squared:  0.7938
## F-statistic: 88.77 on 5 and 109 DF,  p-value: < 2.2e-16
```