Predicting World Happiness for year 2023 using Data Analysis

2024-04-22

TEAM MEMBERS

1. Guruteja Kanderi
2. Alphy Benny
3. Chandini Nalamalapu

BUSINESS CONTEXT

Understanding the dynamics of global happiness and well-being is paramount for organizations aiming to foster environments conducive to growth and prosperity. By leveraging insights from the World Happiness Report, businesses can craft strategies that prioritize employee satisfaction, societal well-being, and sustainable development. These informed decisions not only enhance workplace culture but also contribute to the creation of resilient and thriving economies on a global scale.

PROBLEM DESCRIPTION

In our modern, interconnected world, grasping the complex tapestry of global happiness and well-being stands as a vital endeavor. The mission of the World Happiness Report is to delve deep into the realms of human satisfaction, unraveling the diverse factors that shape it. This pursuit transcends mere academic inquiry; it represents a journey to uncover the essence of human fulfillment, guiding informed policy making and precise interventions. Through the development of a predictive framework that integrates socio-economic, health, and other relevant variables, our aim is to illuminate a path toward a world that is not only happier but also more resilient. (Please note that the analysis is based on data from the year 2023.)

DATASET DESCRIPTION

The World Happiness Report dataset consist of several variables, such as:

1. Country : The name of the country
2. Region : Region to which the country belongs
3. Ladder Score : the overall score of happiness in a country on an average from the collective responses of the people based on the factors impacting well being.
4. GDP : the gross domestic product per capita in purchasing power parity (PPP) adjusted dollars.
5. Freedom : the perceived freedom to make life choices in the country.
6. Corruption : the perceived level of corruption in the government and business sectors of the country.
7. Life Expectancy : Healthy life expectancy at birth
8. Social Support : the perceived social support (ability to count on others) in the country
9. Dystopia residual : the extent to which the country's happiness score deviates from an imagined dystopian society with the least happy possible outcomes.
10. Generosity : the perceived generosity of people in the country.

PROJECT FLOW

STEP 1: INCLUDING LIBRARIES:

```
library(MASS)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.3.3
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

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

```
## Warning: package 'data.table' was built under R version 4.3.3
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:reshape2':
##
##     dcast, melt
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
```

```
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(DataExplorer)
```

```
## Warning: package 'DataExplorer' was built under R version 4.3.3
```

```
library(viridis)
```

```
## Warning: package 'viridis' was built under R version 4.3.3
```

```
## Loading required package: viridisLite
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

```
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

STEP 2: DATA PREPARATION:

Data Loading

```
OriginalData <- read.csv("C:/Users/18722/Downloads/WHR2023.csv", na.strings = c('N/A'))
```

Checking data type

```
print(class(OriginalData))
```

```
## [1] "data.frame"
```

Reviewing the data

```
#Review the Original Data
OriginalData <- as.data.table(OriginalData)
OriginalData
```

```
##          Country.name           Regional.indicator Ladder.score
##                <char>                       <char>        <num>
##   1:          Finland              Western Europe        7.804
##   2:          Denmark              Western Europe        7.586
##   3:          Iceland              Western Europe        7.530
##   4:           Israel Middle East and North Africa        7.473
##   5:      Netherlands              Western Europe        7.403
##  ---
## 133: Congo (Kinshasa)          Sub-Saharan Africa        3.207
## 134:         Zimbabwe          Sub-Saharan Africa        3.204
## 135:     Sierra Leone          Sub-Saharan Africa        3.138
## 136:          Lebanon Middle East and North Africa        2.392
## 137:      Afghanistan                  South Asia        1.859
##      Standard.error.of.ladder.score upperwhisker lowerwhisker
##                               <num>        <num>        <num>
##   1:                          0.036        7.875        7.733
##   2:                          0.041        7.667        7.506
##   3:                          0.049        7.625        7.434
##   4:                          0.032        7.535        7.411
##   5:                          0.029        7.460        7.346
##  ---
## 133:                          0.095        3.394        3.020
## 134:                          0.061        3.323        3.084
## 135:                          0.082        3.299        2.976
## 136:                          0.044        2.479        2.305
## 137:                          0.033        1.923        1.795
##      Logged.GDP.per.capita Social.support Healthy.life.expectancy
##                      <num>          <num>                   <num>
##   1:                10.792          0.969                    71.150
##   2:                10.962          0.954                    71.250
##   3:                10.896          0.983                    72.050
##   4:                10.639          0.943                    72.697
##   5:                10.942          0.930                    71.550
##  ---
## 133:                 7.007          0.652                    55.375
## 134:                 7.641          0.690                    54.050
## 135:                 7.394          0.555                    54.900
## 136:                 9.478          0.530                    66.149
## 137:                 7.324          0.341                    54.712
##      Freedom.to.make.life.choices Generosity Perceptions.of.corruption
##                             <num>      <num>                    <num>
##   1:                        0.961     -0.019                    0.182
##   2:                        0.934      0.134                    0.196
##   3:                        0.936      0.211                    0.668
##   4:                        0.809     -0.023                    0.708
##   5:                        0.887      0.213                    0.379
##  ---
## 133:                        0.664      0.086                    0.834
## 134:                        0.654     -0.046                    0.766
## 135:                        0.660      0.105                    0.858
## 136:                        0.474     -0.141                    0.891
## 137:                        0.382     -0.081                    0.847
##      Ladder.score.in.Dystopia Explained.by..Log.GDP.per.capita
##                         <num>                           <num>
```

```
##   1:                    1.778                             1.888
##   2:                    1.778                             1.949
##   3:                    1.778                             1.926
##   4:                    1.778                             1.833
##   5:                    1.778                             1.942
##  ---
## 133:                    1.778                             0.531
## 134:                    1.778                             0.758
## 135:                    1.778                             0.670
## 136:                    1.778                             1.417
## 137:                    1.778                             0.645
##      Explained.by..Social.support Explained.by..Healthy.life.expectancy
##                             <num>                                 <num>
##   1:                        1.585                                 0.535
##   2:                        1.548                                 0.537
##   3:                        1.620                                 0.559
##   4:                        1.521                                 0.577
##   5:                        1.488                                 0.545
##  ---
## 133:                        0.784                                 0.105
## 134:                        0.881                                 0.069
## 135:                        0.540                                 0.092
## 136:                        0.476                                 0.398
## 137:                        0.000                                 0.087
##      Explained.by..Freedom.to.make.life.choices Explained.by..Generosity
##                                           <num>                   <num>
##   1:                                      0.772                   0.126
##   2:                                      0.734                   0.208
##   3:                                      0.738                   0.250
##   4:                                      0.569                   0.124
##   5:                                      0.672                   0.251
##  ---
## 133:                                      0.375                   0.183
## 134:                                      0.363                   0.112
## 135:                                      0.371                   0.193
## 136:                                      0.123                   0.061
## 137:                                      0.000                   0.093
##      Explained.by..Perceptions.of.corruption Dystopia...residual
##                                        <num>               <num>
##   1:                                   0.535               2.363
##   2:                                   0.525               2.084
##   3:                                   0.187               2.250
##   4:                                   0.158               2.691
##   5:                                   0.394               2.110
##  ---
## 133:                                   0.068               1.162
## 134:                                   0.117               0.905
## 135:                                   0.051               1.221
## 136:                                   0.027              -0.110
## 137:                                   0.059               0.976
```

Finding number of observations and columns in dataset

```
dim(OriginalData)
```

```
## [1] 137  20
```

There are 137 observations and 20 features in dataset.

Reviewing the structure of data frame

```
str(OriginalData)
```

```
## Classes 'data.table' and 'data.frame':   137 obs. of  20 variables:
##  $ Country.name                             : chr  "Finland" "Denmark" "Iceland" "Israel" ...
##  $ Regional.indicator                       : chr  "Western Europe" "Western Europe" "Western Europ
##  $ Ladder.score                             : num  7.8 7.59 7.53 7.47 7.4 ...
##  $ Standard.error.of.ladder.score           : num  0.036 0.041 0.049 0.032 0.029 0.037 0.044 0.043 
##  $ upperwhisker                             : num  7.88 7.67 7.62 7.54 7.46 ...
##  $ lowerwhisker                             : num  7.73 7.51 7.43 7.41 7.35 ...
##  $ Logged.GDP.per.capita                    : num  10.8 11 10.9 10.6 10.9 ...
##  $ Social.support                           : num  0.969 0.954 0.983 0.943 0.93 0.939 0.943 0.92 0.8
##  $ Healthy.life.expectancy                  : num  71.2 71.2 72 72.7 71.5 ...
##  $ Freedom.to.make.life.choices             : num  0.961 0.934 0.936 0.809 0.887 0.948 0.947 0.891 
##  $ Generosity                               : num  -0.019 0.134 0.211 -0.023 0.213 0.165 0.141 0.027
##  $ Perceptions.of.corruption                : num  0.182 0.196 0.668 0.708 0.379 0.202 0.283 0.266 
##  $ Ladder.score.in.Dystopia                 : num  1.78 1.78 1.78 1.78 1.78 ...
##  $ Explained.by..Log.GDP.per.capita         : num  1.89 1.95 1.93 1.83 1.94 ...
##  $ Explained.by..Social.support             : num  1.58 1.55 1.62 1.52 1.49 ...
##  $ Explained.by..Healthy.life.expectancy    : num  0.535 0.537 0.559 0.577 0.545 0.562 0.544 0.582 
##  $ Explained.by..Freedom.to.make.life.choices: num  0.772 0.734 0.738 0.569 0.672 0.754 0.752 0.678 
##  $ Explained.by..Generosity                 : num  0.126 0.208 0.25 0.124 0.251 0.225 0.212 0.151 0
##  $ Explained.by..Perceptions.of.corruption  : num  0.535 0.525 0.187 0.158 0.394 0.52 0.463 0.475 0
##  $ Dystopia...residual                      : num  2.36 2.08 2.25 2.69 2.11 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

All Columns are assigned with correct data types.

Loading required data in data frame.

```
data <- subset(OriginalData, select=c("Country.name",
                                      "Regional.indicator",
                                      "Ladder.score",
                                      "Logged.GDP.per.capita",
                                      "Social.support",
                                      "Healthy.life.expectancy",
                                      "Freedom.to.make.life.choices",
                                      "Generosity",
                                      "Perceptions.of.corruption",
                                      "Dystopia...residual"))

data
```

```
##          Country.name           Regional.indicator Ladder.score
```

```
##                <char>                     <char>        <num>
##   1:          Finland             Western Europe        7.804
##   2:          Denmark             Western Europe        7.586
##   3:          Iceland             Western Europe        7.530
##   4:           Israel Middle East and North Africa        7.473
##   5:      Netherlands             Western Europe        7.403
##  ---
## 133: Congo (Kinshasa)         Sub-Saharan Africa        3.207
## 134:         Zimbabwe         Sub-Saharan Africa        3.204
## 135:     Sierra Leone         Sub-Saharan Africa        3.138
## 136:          Lebanon Middle East and North Africa        2.392
## 137:      Afghanistan                 South Asia        1.859
##      Logged.GDP.per.capita Social.support Healthy.life.expectancy
##                      <num>          <num>                  <num>
##   1:                10.792          0.969                 71.150
##   2:                10.962          0.954                 71.250
##   3:                10.896          0.983                 72.050
##   4:                10.639          0.943                 72.697
##   5:                10.942          0.930                 71.550
##  ---
## 133:                 7.007          0.652                 55.375
## 134:                 7.641          0.690                 54.050
## 135:                 7.394          0.555                 54.900
## 136:                 9.478          0.530                 66.149
## 137:                 7.324          0.341                 54.712
##      Freedom.to.make.life.choices Generosity Perceptions.of.corruption
##                             <num>      <num>                    <num>
##   1:                        0.961     -0.019                    0.182
##   2:                        0.934      0.134                    0.196
##   3:                        0.936      0.211                    0.668
##   4:                        0.809     -0.023                    0.708
##   5:                        0.887      0.213                    0.379
##  ---
## 133:                        0.664      0.086                    0.834
## 134:                        0.654     -0.046                    0.766
## 135:                        0.660      0.105                    0.858
## 136:                        0.474     -0.141                    0.891
## 137:                        0.382     -0.081                    0.847
##      Dystopia...residual
##                    <num>
##   1:               2.363
##   2:               2.084
##   3:               2.250
##   4:               2.691
##   5:               2.110
##  ---
## 133:               1.162
## 134:               0.905
## 135:               1.221
## 136:              -0.110
## 137:               0.976
```

Renaming the names of columns for better readability

```
library(dplyr)
data <- data %>% rename( Country=Country.name,
                         Region=Regional.indicator,
                         Happiness_score=Ladder.score,
                         GDP=Logged.GDP.per.capita,
                         Social_support=Social.support,
                         Life_expectancy=Healthy.life.expectancy,
                         Freedom=Freedom.to.make.life.choices,
                         Corruption=Perceptions.of.corruption,
                         Dystopia_residual=Dystopia...residual)

data
```

```
##                Country                       Region Happiness_score    GDP
##                 <char>                       <char>           <num>  <num>
##   1:           Finland               Western Europe           7.804 10.792
##   2:           Denmark               Western Europe           7.586 10.962
##   3:           Iceland               Western Europe           7.530 10.896
##   4:            Israel Middle East and North Africa           7.473 10.639
##   5:       Netherlands               Western Europe           7.403 10.942
##  ---
## 133: Congo (Kinshasa)           Sub-Saharan Africa           3.207  7.007
## 134:          Zimbabwe           Sub-Saharan Africa           3.204  7.641
## 135:      Sierra Leone           Sub-Saharan Africa           3.138  7.394
## 136:           Lebanon Middle East and North Africa           2.392  9.478
## 137:       Afghanistan                  South Asia           1.859  7.324
##      Social_support Life_expectancy Freedom Generosity Corruption
##                <num>           <num>   <num>      <num>      <num>
##   1:           0.969          71.150   0.961     -0.019      0.182
##   2:           0.954          71.250   0.934      0.134      0.196
##   3:           0.983          72.050   0.936      0.211      0.668
##   4:           0.943          72.697   0.809     -0.023      0.708
##   5:           0.930          71.550   0.887      0.213      0.379
##  ---
## 133:           0.652          55.375   0.664      0.086      0.834
## 134:           0.690          54.050   0.654     -0.046      0.766
## 135:           0.555          54.900   0.660      0.105      0.858
## 136:           0.530          66.149   0.474     -0.141      0.891
## 137:           0.341          54.712   0.382     -0.081      0.847
##      Dystopia_residual
##                  <num>
##   1:             2.363
##   2:             2.084
##   3:             2.250
##   4:             2.691
##   5:             2.110
##  ---
## 133:             1.162
## 134:             0.905
## 135:             1.221
## 136:            -0.110
## 137:             0.976
```

Reviewing summary of data

```
summary(data)
```

```
##    Country             Region          Happiness_score      GDP
##  Length:137         Length:137         Min.   :1.859   Min.   : 5.527
##  Class :character   Class :character   1st Qu.:4.724   1st Qu.: 8.591
##  Mode  :character   Mode  :character   Median :5.684   Median : 9.567
##                                        Mean   :5.540   Mean   : 9.450
##                                        3rd Qu.:6.334   3rd Qu.:10.540
##                                        Max.   :7.804   Max.   :11.660
##  Social_support   Life_expectancy    Freedom         Generosity
##  Min.   :0.3410   Min.   : 0.00   Min.   :0.3820   Min.   :-0.25400
##  1st Qu.:0.7220   1st Qu.:60.50   1st Qu.:0.7240   1st Qu.:-0.07400
##  Median :0.8270   Median :65.83   Median :0.8010   Median : 0.00100
##  Mean   :0.7991   Mean   :64.49   Mean   :0.7874   Mean   : 0.02243
##  3rd Qu.:0.8960   3rd Qu.:69.35   3rd Qu.:0.8740   3rd Qu.: 0.11700
##  Max.   :0.9830   Max.   :77.28   Max.   :0.9610   Max.   : 0.53100
##    Corruption     Dystopia_residual
##  Min.   :0.1460   Min.   :-0.110
##  1st Qu.:0.6680   1st Qu.: 1.553
##  Median :0.7740   Median : 1.845
##  Mean   :0.7254   Mean   : 1.765
##  3rd Qu.:0.8460   3rd Qu.: 2.078
##  Max.   :0.9290   Max.   : 2.955
```

Descriptive Statistics

```
Mean_Happiness_Score <- mean(data$Happiness_score)
Variance_Happiness_Score <- var(data$Happiness_score)
StdDev_Happiness_Score <- sd(data$Happiness_score)
Median_Happiness_Score <- median(data$Happiness_score)

cat("Mean Happiness score is ",Mean_Happiness_Score)
```

```
## Mean Happiness score is  5.539796
```

```
cat("\nVariance of Happiness score is ",Variance_Happiness_Score)
```

```
##
## Variance of Happiness score is  1.299438
```

```
cat("\nStandard Deviation of Happiness score is ",StdDev_Happiness_Score)
```

```
##
## Standard Deviation of Happiness score is  1.139929
```

```
cat("\nMedian Happiness score is ",Median_Happiness_Score)
```

```
##
## Median Happiness score is  5.684
```

STEP 3: DATA CLEANING

Checking missing values

```
MissingValueColumnName <- colnames(data)[colSums(is.na(data)) > 0]
NumberOfMissingValues <- sum(is.na(data))
cat("There are", NumberOfMissingValues, "Missing Values in Column name", MissingValueColumnName)
```

```
## There are 0 Missing Values in Column name
```

Unique values in Region

```
# Get the unique values of a Region column
unique_values <- data %>% distinct(Region)

# Print the unique values
print(unique_values)
```

```
##                          Region
##                          <char>
## 1:               Western Europe
## 2: Middle East and North Africa
## 3:        North America and ANZ
## 4:   Central and Eastern Europe
## 5:  Latin America and Caribbean
## 6:                    East Asia
## 7:               Southeast Asia
## 8:           Sub-Saharan Africa
## 9:                   South Asia
```

Combining Multiple similar named region into one

```
library(dplyr)
library(forcats)
# Creating data frame for unique region
dataRegion <- data
# change region to factor
dataRegion$Region <- as.factor(dataRegion$Region)
# combine south asia and southeast asia and east asia
dataRegion <- dataRegion %>% mutate(Region = fct_recode(Region,
                            # new name         old name
                            "Southeast Asia" = "South Asia",
                            "Southeast Asia" = "East Asia"))

# Get the unique values of a Region column
unique_values <- unique(dataRegion$Region)

# Print the unique values
print(unique_values)
```

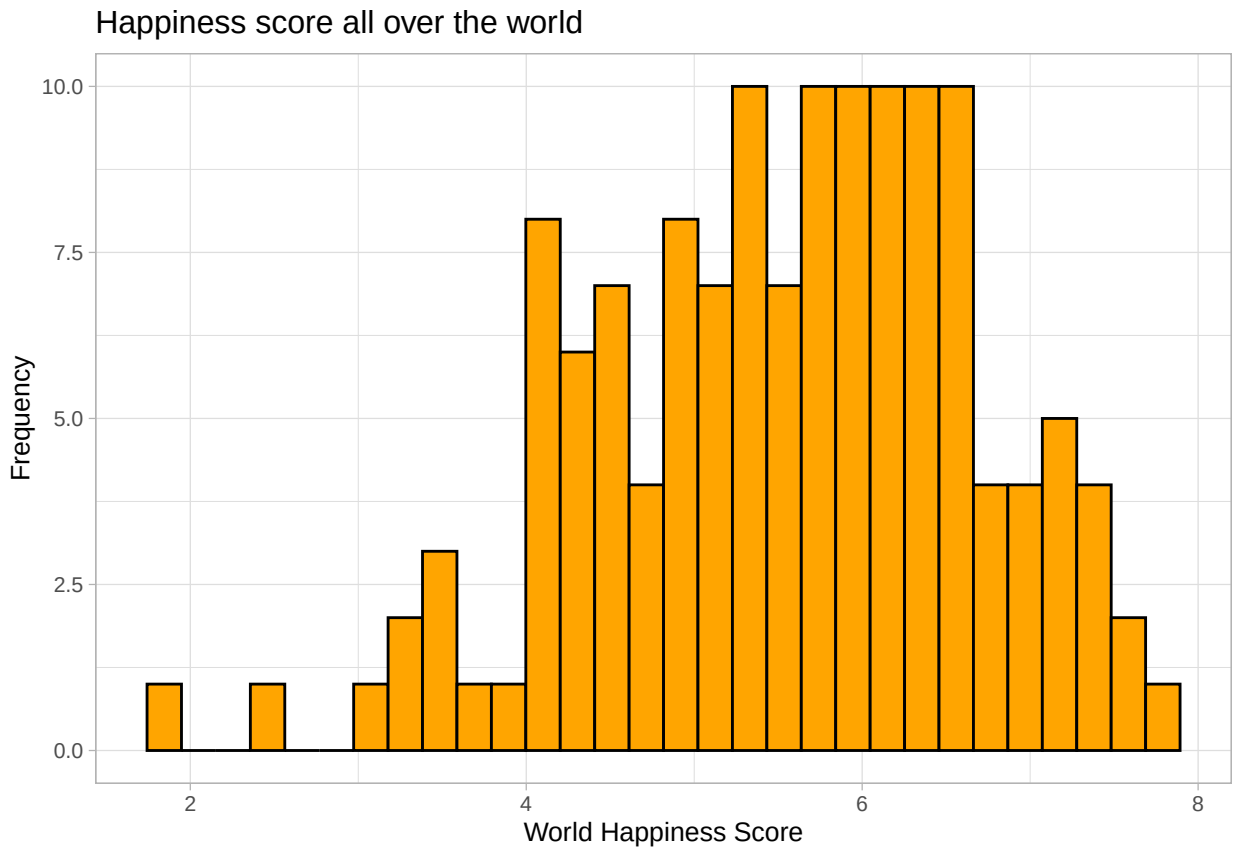
```
## [1] Western Europe               Middle East and North Africa
## [3] North America and ANZ        Central and Eastern Europe
```

```
## [5] Latin America and Caribbean  Southeast Asia
## [7] Sub-Saharan Africa
## 7 Levels: Central and Eastern Europe ... Western Europe
```

STEP 4: CHECKING NORMAL DISTRIBUTION OF TARGET VARIABLE

1. Checking distribution of target variable Happiness_score using Histogram as we know that target variable is a continuous variable

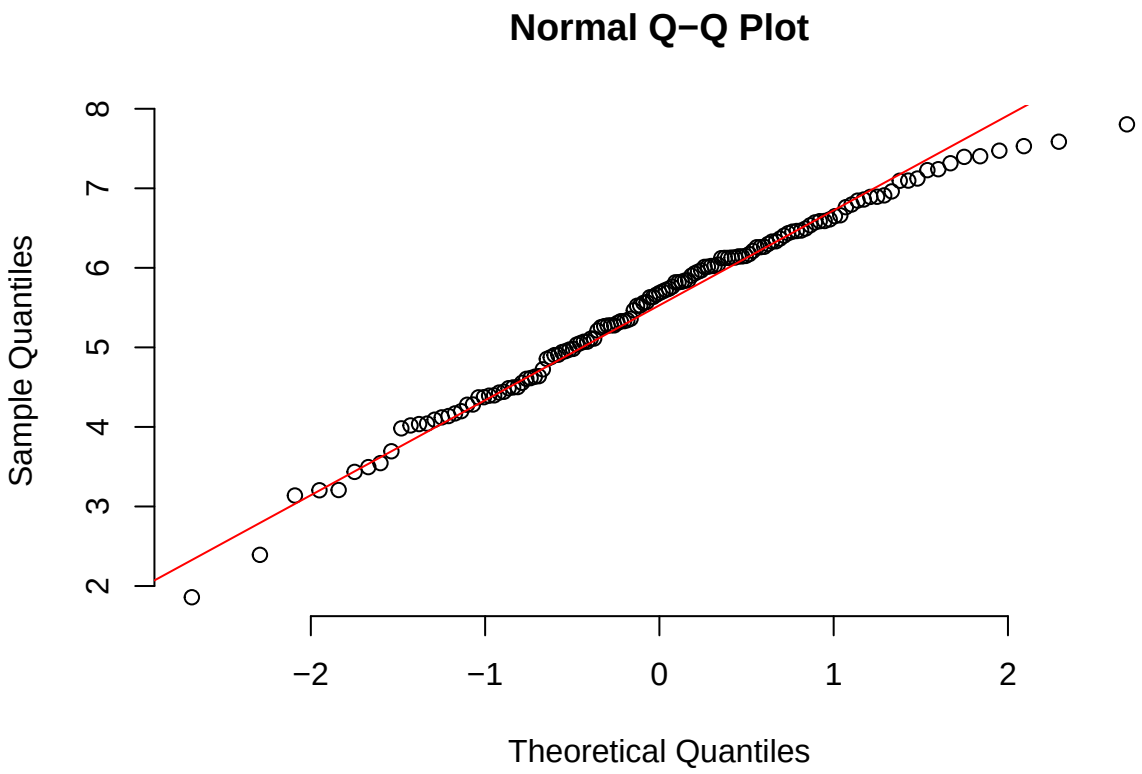
```
library(ggplot2)
ggplot(data=data, aes(data$Happiness_score,y = after_stat(count))) +
  geom_histogram( fill = "orange", color = "black", bins = 30)+
  theme_light(base_size = 10)+
  labs(title="Happiness score all over the world", x = "World Happiness Score", y = "Frequency")
```



The plot suggests that the distribution is roughly bell-shaped, with a peak around the score of 6. Additionally, the plot suggests that the distribution is slightly skewed to the right.

2. Plot to check normal distribution of target variable

```
qqnorm(data$Happiness_score, pch = 1, frame = FALSE)
qqline(data$Happiness_score, col = "red")
```

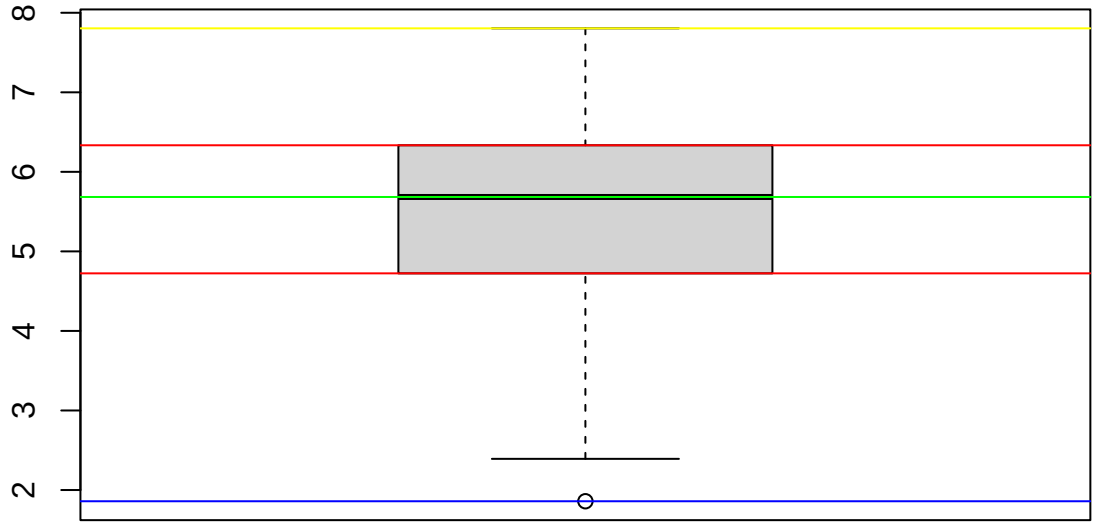


In the above normal probability plot , the data points roughly follow a straight line which states that the data follows a normal distribution.

STEP 5 : CHECKING OUTLIERS IN TARGET VARIABLE

1. Checking outliers in target variable using Boxplot

```
boxplot(data$Happiness_score)
abline(h = min(data$Happiness_score), col = "Blue")
abline(h = max(data$Happiness_score), col = "Yellow")
abline(h = median(data$Happiness_score), col = "Green")
abline(h = quantile(data$Happiness_score, c(0.25, 0.75)), col = "Red")
```



The resulting plot shows a box representing the interquartile range (IQR) of the Happiness_score variable, with whiskers extending to the minimum and maximum values within 1.5 times the IQR from the lower and upper quartiles, respectively. The horizontal reference lines are overlaid on the box plot, highlighting the minimum, maximum, median, and quartile values. By inspecting the plot, any values beyond the whiskers can be considered potential outliers. There is one data point beyond the whiskers, it should be investigated further to determine whether it is a true outliers in the data or not.

STEP 6 : EXPLORATORY DATA ANALYSIS

1. Checking Top 10 Happiest Country

1.a. Top 10 Happiest country in the world

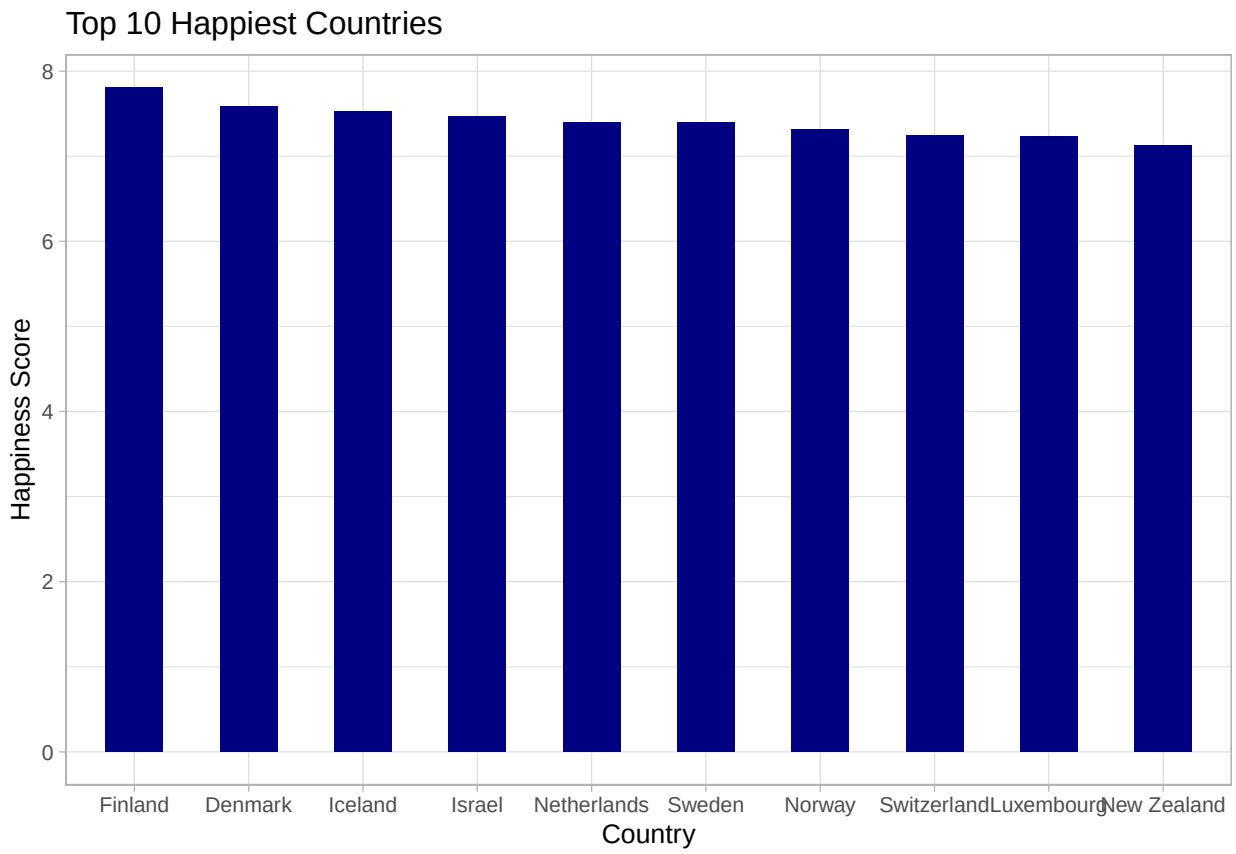
```
# Arrange the data by Happiness_score in descending order and select the top 10 rows
Top_10_Happiest_Country <- data %>%
  arrange(desc(Happiness_score)) %>%
  slice_head(n = 10)

# Print the top 10 happiest countries
Top_10_Happiest_Country
```

```
##         Country                      Region Happiness_score    GDP
##          <char>                      <char>           <num>  <num>
##  1:     Finland              Western Europe           7.804 10.792
```

```
##  2:     Denmark               Western Europe           7.586 10.962
##  3:     Iceland               Western Europe           7.530 10.896
##  4:      Israel Middle East and North Africa           7.473 10.639
##  5: Netherlands               Western Europe           7.403 10.942
##  6:      Sweden               Western Europe           7.395 10.883
##  7:      Norway               Western Europe           7.315 11.088
##  8: Switzerland               Western Europe           7.240 11.164
##  9:  Luxembourg               Western Europe           7.228 11.660
## 10: New Zealand        North America and ANZ           7.123 10.662
##     Social_support Life_expectancy Freedom Generosity Corruption
##              <num>           <num>   <num>      <num>      <num>
##  1:          0.969          71.150   0.961     -0.019      0.182
##  2:          0.954          71.250   0.934      0.134      0.196
##  3:          0.983          72.050   0.936      0.211      0.668
##  4:          0.943          72.697   0.809     -0.023      0.708
##  5:          0.930          71.550   0.887      0.213      0.379
##  6:          0.939          72.150   0.948      0.165      0.202
##  7:          0.943          71.500   0.947      0.141      0.283
##  8:          0.920          72.900   0.891      0.027      0.266
##  9:          0.879          71.675   0.915      0.024      0.345
## 10:          0.952          70.350   0.887      0.175      0.271
##     Dystopia_residual
##                 <num>
##  1:             2.363
##  2:             2.084
##  3:             2.250
##  4:             2.691
##  5:             2.110
##  6:             1.903
##  7:             1.829
##  8:             1.870
##  9:             1.845
## 10:             1.852
```

Bar plot is used to visualize the distribution of categorical variable Country with continuous variable Happiness Score for top 10 Happiest Countries in the world.

```
ggplot(Top_10_Happiest_Country,aes(x=factor(Country,levels=Country),y=Happiness_score))+
  geom_bar(stat="identity",width=0.5,fill="navyblue")+
  theme(axis.text.x = element_text(angle=90, vjust=0.6))+
  theme_light(base_size = 10)+
  labs(title="Top 10 Happiest Countries",x="Country",y="Happiness Score")
```



This plot can help us compare the happiness scores of different countries and understand which countries have the highest levels of happiness.

2. EDA for some predictor variables like Life Expectancy, Freedom against target variable Happiness Score.

2.a. Happiness Score & Life Expectency

```
library(ggplot2)
library(viridis)
ggplot(data = data, aes(x = Happiness_score, y =  Life_expectancy)) +
  geom_point(aes(color = Life_expectancy), show.legend = T, pch=1, stroke=3, size = 2) +
  theme_light(base_size = 10) + scale_color_viridis(discrete = F) +
  labs(title = "Happiness Score & Life Expectancy", fill = "Happiness Score",
       x = "Happiness Score" , y = "Life Expectancy")
```

We are using scatter plot to investigate the distribution between 2 continuous variable Happiness score and Life expectancy where each point represents a country.This plot shows that there appears to be a pretty significant relationship between Life Expectancy and the happiness score as the data points appear to show a positive trend.

2.d. Happiness Score & Freedom

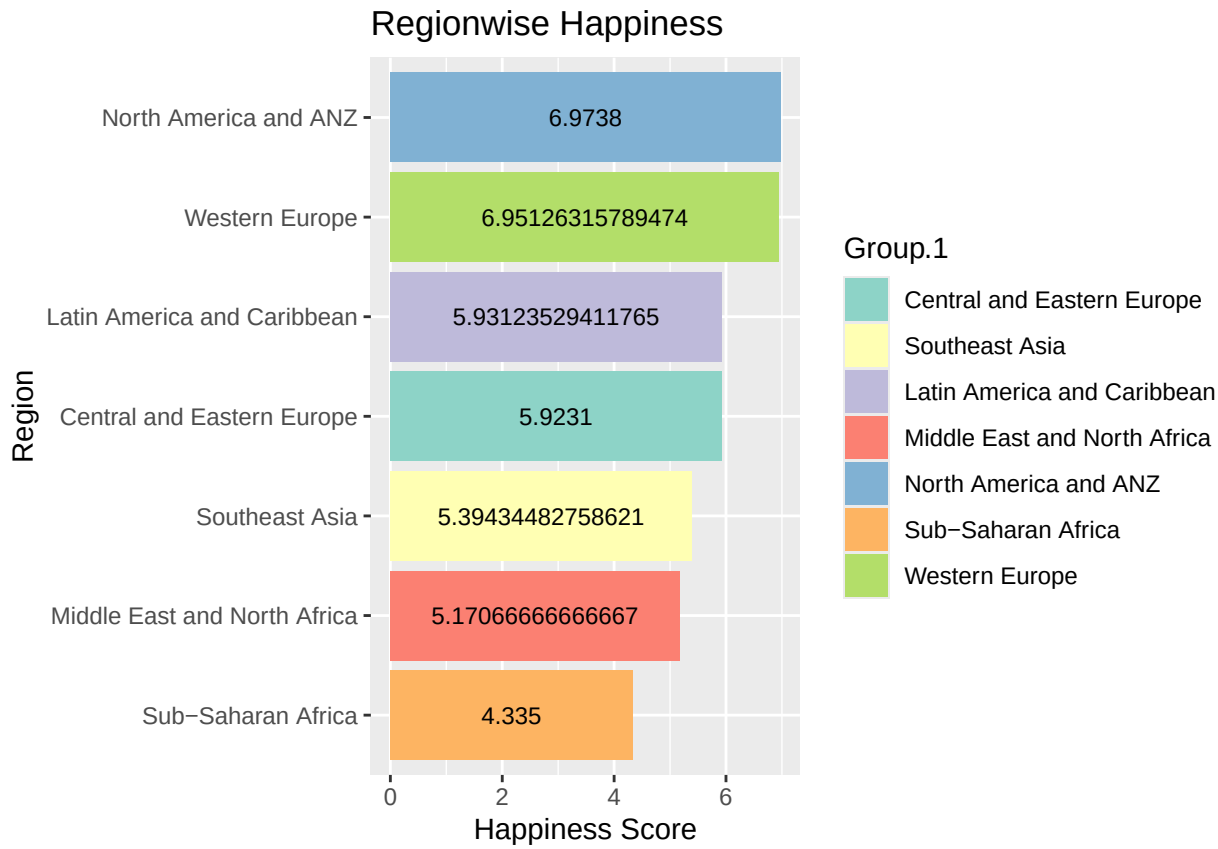
```
ggplot(data = data, aes(x = Happiness_score, y =  Freedom)) +
  geom_point(aes(color = Freedom), show.legend = T, pch=1, stroke=3, size = 2) +
  theme_light(base_size = 10) + scale_color_viridis(discrete = F) +
  labs(title = "Happiness Score & Freedom", fill = "Happiness Score",
       x = "Happiness Score" , y = "Freedom")
```

As predicted this plot shows that as freedom increases so does the happiness score, but we do not know if this relationship is causal and there may be confounds.

3. Happiness Score & Region

3.a. Regionwise Happiness score using overlay barplot

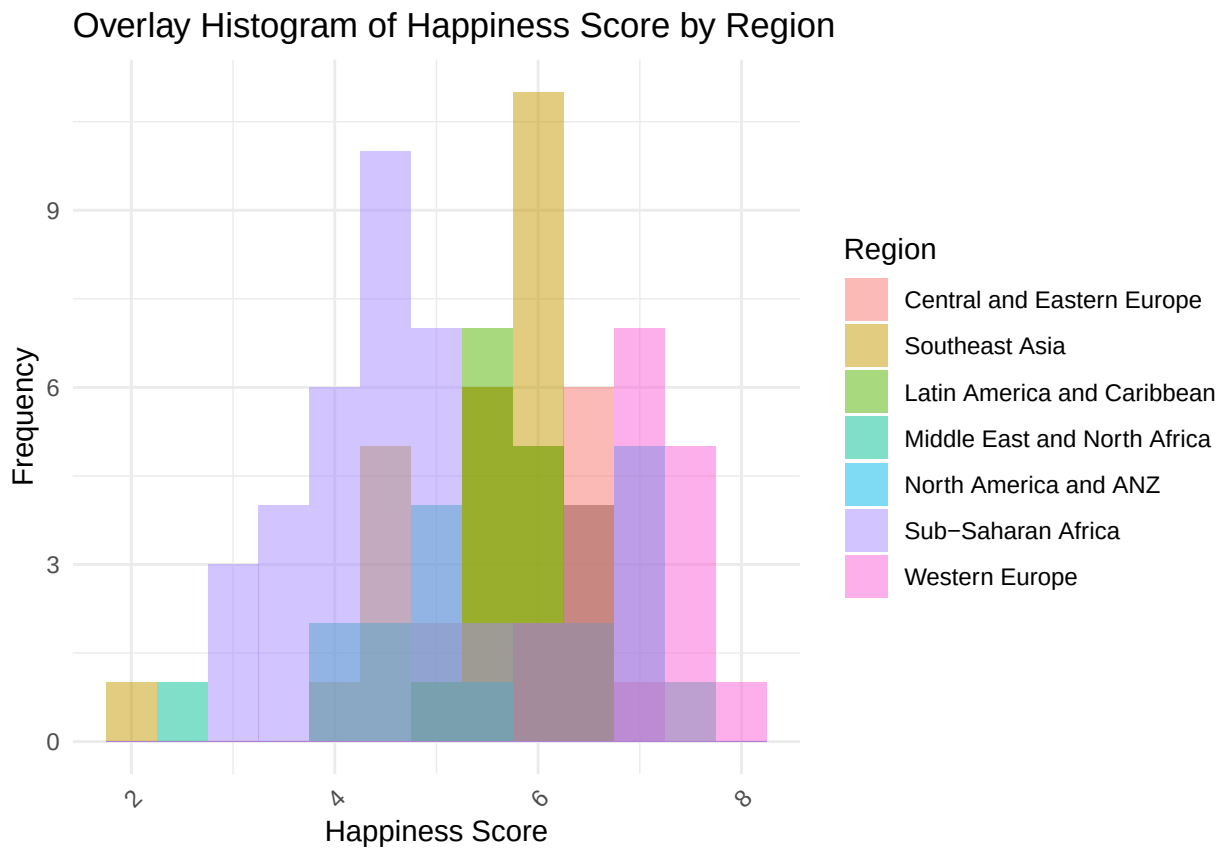
```
data_means <- aggregate(dataRegion$Happiness_score, list(dataRegion$Region), mean)

data_means %>% arrange(desc(x)) %>% ggplot(aes(x = x, y = reorder(Group.1, x),  fill = Group.1)) +
  geom_bar(stat = "identity") +
  labs(title = "Regionwise Happiness") +
  ylab("Region") +
  xlab("Happiness Score") +
  scale_fill_brewer(palette = "Set3") +
  geom_text(aes(label = x), position=position_stack(vjust=0.5),color="black",size=3)
```



The above barplot gives the region wise Happiness Score from highest to lowest.

3.b. Happiness score and Region overlay histogram

```
# Create the histogram using ggplot2
ggplot(dataRegion, aes(x = Happiness_score, fill = Region)) +
  geom_histogram(binwidth = 0.5, alpha = 0.5, position = "identity") +
  theme_minimal() +
  labs(title = "Overlay Histogram of Happiness Score by Region", x = "Happiness Score", y = "Frequency")
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



> We are using histogram to display the distribution of a continuous variable Happiness score broken down by a categorical variable Region. From the above boxplot, we can say that Western Europe and North America and ANZ has maximum happiness score and Sub-Saharan Africa has lowest happiness score.

5. Exploratory data analysis of Region with other variables.

Combine Regions to reduce number of categories

```
dataRegionCat <- dataRegion %>%
  mutate(Region = fct_recode(Region,
                             # new name         old name
                             "Africa" = "Sub-Saharan Africa",
                             "ME"     = "Middle East and North Africa",
                             "N AMR"  = "North America and ANZ",
                             "CIS"    = "Commonwealth of Independent States",
                             "S AMR"  = "Latin America and Caribbean",
                             "SE Asia"= "Southeast Asia",
                             "EUR"    = "Central and Eastern Europe",
                             "EUR"    = "Western Europe" ))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `Region = fct_recode(...)`.
## Caused by warning:
## ! Unknown levels in `f`: Commonwealth of Independent States
```

```
#Reviewing the cropped names of Region column
dataRegionCat
```

```
##               Country  Region Happiness_score    GDP Social_support
##                <char>  <fctr>           <num>  <num>          <num>
##   1:          Finland     EUR           7.804 10.792          0.969
##   2:          Denmark     EUR           7.586 10.962          0.954
##   3:          Iceland     EUR           7.530 10.896          0.983
##   4:           Israel      ME           7.473 10.639          0.943
##   5:      Netherlands     EUR           7.403 10.942          0.930
##  ---
## 133: Congo (Kinshasa)  Africa           3.207  7.007          0.652
## 134:         Zimbabwe  Africa           3.204  7.641          0.690
## 135:     Sierra Leone  Africa           3.138  7.394          0.555
## 136:          Lebanon      ME           2.392  9.478          0.530
## 137:      Afghanistan SE Asia           1.859  7.324          0.341
##      Life_expectancy Freedom Generosity Corruption Dystopia_residual
##                <num>   <num>      <num>      <num>             <num>
##   1:          71.150   0.961     -0.019      0.182             2.363
##   2:          71.250   0.934      0.134      0.196             2.084
##   3:          72.050   0.936      0.211      0.668             2.250
##   4:          72.697   0.809     -0.023      0.708             2.691
##   5:          71.550   0.887      0.213      0.379             2.110
##  ---
## 133:          55.375   0.664      0.086      0.834             1.162
## 134:          54.050   0.654     -0.046      0.766             0.905
## 135:          54.900   0.660      0.105      0.858             1.221
## 136:          66.149   0.474     -0.141      0.891            -0.110
## 137:          54.712   0.382     -0.081      0.847             0.976
```

Create a new factor variable called score

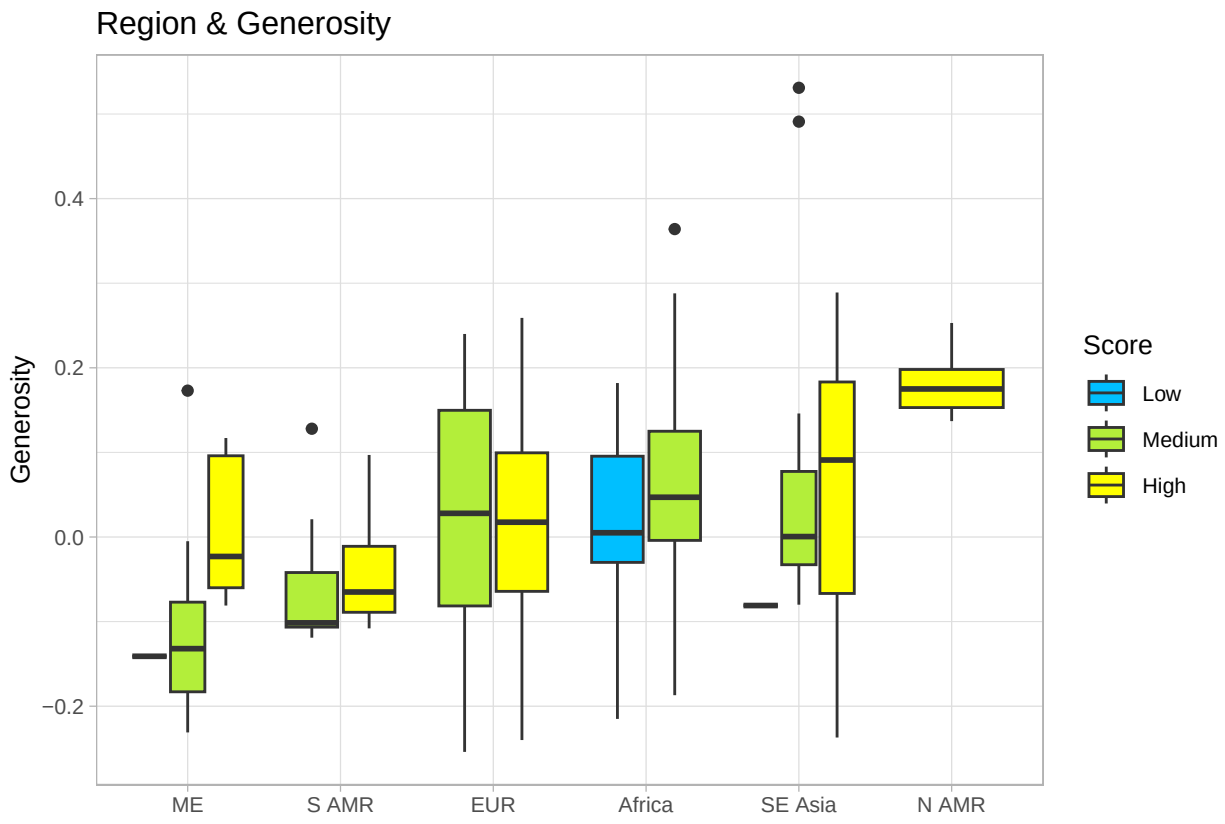
```
library(dplyr)
library(tidyr)
# Create a New Factor Variable for Happiness

mn = min(dataRegionCat$Happiness_score)
mx = max(dataRegionCat$Happiness_score)
b = (mx - mn)/3.0
brk = seq(from = (mn-0.001), to = mx, by = b)

# create new categorical variable
dataRegionCat$Score <- cut(dataRegionCat$Happiness_score, breaks = brk,labels = c("Low", "Medium", "High
dataRegionCat <- dataRegionCat %>% drop_na()

#Reviewing Score column in dataset
dataRegionCat
```

```
##               Country  Region Happiness_score    GDP Social_support
##                <char>  <fctr>           <num>  <num>          <num>
##   1:          Denmark     EUR           7.586 10.962          0.954
##   2:          Iceland     EUR           7.530 10.896          0.983
##   3:           Israel      ME           7.473 10.639          0.943
##   4:      Netherlands     EUR           7.403 10.942          0.930
```

```
##   5:           Sweden     EUR           7.395 10.883          0.939
##  ---
## 132: Congo (Kinshasa)  Africa           3.207  7.007          0.652
## 133:         Zimbabwe  Africa           3.204  7.641          0.690
## 134:     Sierra Leone  Africa           3.138  7.394          0.555
## 135:          Lebanon      ME           2.392  9.478          0.530
## 136:      Afghanistan SE Asia           1.859  7.324          0.341
##      Life_expectancy Freedom Generosity Corruption Dystopia_residual  Score
##                <num>   <num>      <num>      <num>             <num> <fctr>
##   1:          71.250   0.934      0.134      0.196             2.084   High
##   2:          72.050   0.936      0.211      0.668             2.250   High
##   3:          72.697   0.809     -0.023      0.708             2.691   High
##   4:          71.550   0.887      0.213      0.379             2.110   High
##   5:          72.150   0.948      0.165      0.202             1.903   High
##  ---
## 132:          55.375   0.664      0.086      0.834             1.162    Low
## 133:          54.050   0.654     -0.046      0.766             0.905    Low
## 134:          54.900   0.660      0.105      0.858             1.221    Low
## 135:          66.149   0.474     -0.141      0.891            -0.110    Low
## 136:          54.712   0.382     -0.081      0.847             0.976    Low
```

5.f. Region by Generosity

```
ggplot(data = dataRegionCat, aes(x = reorder(Region, Generosity),
                    y = Generosity, fill = Score)) +
  geom_boxplot(show.legend = T) +
  labs(x = "", fill = "Score", y = "Generosity",
       title = "Region & Generosity") +
  scale_fill_manual(values = c("deepskyblue","olivedrab2", "yellow")) +
  theme_light(base_size = 10)
```



Overall, this plot suggests that generosity varies significantly across regions and may be related to regional differences in happiness scores.

STEP 7 : HYPOTHESIS TEST :

The target variable Happiness score is slightly skewed as discussed above and follows nearly a normal distribution, so we will be using T-statistic instead of Z-statisitc.

1. Hypothesis test for one mean

Creating a separate dataset for South East Asia.

```
Southeast_Asia_Data <- subset(data, Region == "Southeast Asia")
Southeast_Asia_Data
```

```
##         Country         Region Happiness_score    GDP Social_support
##          <char>         <char>           <num>  <num>          <num>
##  1:   Singapore Southeast Asia           6.587 11.571          0.878
##  2:       Malta Southeast Asia           6.300 10.661          0.923
##  3:  Uzbekistan Southeast Asia           6.014  8.948          0.875
##  4:    Malaysia Southeast Asia           6.012 10.169          0.799
##  5:   Mauritius Southeast Asia           5.902  9.957          0.888
```

```
##  6:    Thailand Southeast Asia           5.843  9.751          0.874
##  7:  Kyrgyzstan Southeast Asia           5.825  8.486          0.911
##  8:     Moldova Southeast Asia           5.819  9.499          0.857
##  9:     Vietnam Southeast Asia           5.763  9.287          0.821
## 10: Philippines Southeast Asia           5.523  8.979          0.780
## 11:       Nepal Southeast Asia           5.360  8.256          0.748
## 12:  Tajikistan Southeast Asia           5.330  8.237          0.836
## 13:   Indonesia Southeast Asia           5.277  9.385          0.804
## 14:        Laos Southeast Asia           5.111  8.962          0.679
## 15:    Cambodia Southeast Asia           4.393  8.385          0.747
## 16:     Myanmar Southeast Asia           4.372  8.404          0.787
##     Life_expectancy Freedom Generosity Corruption Dystopia_residual
##               <num>   <num>      <num>      <num>             <num>
##  1:          73.800   0.878      0.063      0.146             1.067
##  2:          71.600   0.886      0.119      0.729             1.429
##  3:          65.301   0.938      0.230      0.638             1.856
##  4:          65.662   0.877      0.160      0.758             1.804
##  5:          63.850   0.813     -0.028      0.775             1.790
##  6:          68.450   0.850      0.289      0.910             1.594
##  7:          66.852   0.934      0.181      0.904             1.920
##  8:          65.299   0.840     -0.080      0.901             1.995
##  9:          65.502   0.939     -0.004      0.759             1.824
## 10:          62.038   0.919     -0.060      0.732             1.931
## 11:          61.847   0.808      0.146      0.783             2.187
## 12:          62.226   0.832     -0.060      0.522             1.823
## 13:          63.048   0.880      0.531      0.876             1.288
## 14:          60.946   0.919      0.091      0.703             1.707
## 15:          61.900   0.958      0.073      0.857             1.068
## 16:          61.388   0.727      0.491      0.658             0.893
```

Reviewing Summary of South East Asia

```
summary(Southeast_Asia_Data)
```

```
##    Country             Region          Happiness_score      GDP
##  Length:16          Length:16          Min.   :4.372   Min.   : 8.237
##  Class :character   Class :character   1st Qu.:5.317   1st Qu.: 8.466
##  Mode  :character   Mode  :character   Median :5.791   Median : 9.133
##                                        Mean   :5.589   Mean   : 9.309
##                                        3rd Qu.:5.929   3rd Qu.: 9.803
##                                        Max.   :6.587   Max.   :11.571
##  Social_support   Life_expectancy    Freedom         Generosity
##  Min.   :0.6790   Min.   :60.95   Min.   :0.7270   Min.   :-0.0800
##  1st Qu.:0.7853   1st Qu.:62.00   1st Qu.:0.8380   1st Qu.:-0.0100
##  Median :0.8285   Median :64.57   Median :0.8790   Median : 0.1050
##  Mean   :0.8254   Mean   :64.98   Mean   :0.8749   Mean   : 0.1339
##  3rd Qu.:0.8758   3rd Qu.:65.96   3rd Qu.:0.9227   3rd Qu.: 0.1933
##  Max.   :0.9230   Max.   :73.80   Max.   :0.9580   Max.   : 0.5310
##    Corruption     Dystopia_residual
##  Min.   :0.1460   Min.   :0.893
##  1st Qu.:0.6917   1st Qu.:1.394
##  Median :0.7585   Median :1.797
##  Mean   :0.7282   Mean   :1.636
```

```
##  3rd Qu.:0.8618   3rd Qu.:1.872
##  Max.   :0.9100   Max.   :2.187
```

Preliminary test to check one-sample t-test assumptions:

1. Is this a large sample? - No, because n < 30 while n = 9.

2. To check whether the data follow a normal distribution:

2.1 Shapiro-Wilk test:

H0: NULL Hypothesis : the data are normally distributed

H1: Alternative Hypothesis : The data is not normally distributed

```
shapiro.test(Southeast_Asia_Data$Happiness_score)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Southeast_Asia_Data$Happiness_score
## W = 0.93142, p-value = 0.2567
```

The output of this test will include a p-value. If the p-value is greater than 0.05, then we cannot reject the null hypothesis that the data is normally distributed. If the p-value is less than 0.05, then we can reject the null hypothesis and conclude that the data is not normally distributed. Here in South East Asia data, pvalue > 0.05 implying that the distribution of the data are not significantly different from normal distribution. In other words, we can assume the normality.

HYPOTHESIS TEST For one mean:

H0 : NULL Hypothesis : Average Happiness score of countires in SouthEast Asia is 5.

H1 : Alternative Hypothesis : Average Happiness score of countires in SouthEast Asia is not equal to 5.

mu : Mean of Happiness scores of South Asia

H0: mu = 5

H1: mu != 5

Manual calculations

```
mu <- 5 #Population mean as per Hypothesis
xbar <- mean(Southeast_Asia_Data$Happiness_score) #Sample mean
s <- sd(Southeast_Asia_Data$Happiness_score) #Sample Standard Deviation
```

```r
n <- nrow(Southeast_Asia_Data) #Sample size

stdError <- s/sqrt(n) #Standard error
df <- n-1 #Degree of freedom
ci <- 0.95 #Confidence interval
alpha <- 1-ci #Significance level
tstatistic <- (xbar - mu)/stdError #T-statistic
tvalue <- qt(1-alpha/2,df) #Critical value of t-distribution

lowerbound <- xbar - tvalue*stdError #lower bound of confidence interval
upperbound <- xbar + tvalue*stdError #upper bound of confidence interval

p_value <- 2 * (1 - pt(abs(tstatistic), df)) #p-value

cat("Sample size is:",n)
```

```
## Sample size is: 16
```

```r
cat("\nT-Statistic value is :",tstatistic)
```

```
##
## T-Statistic value is : 3.885354
```

```r
cat("\nDegree of freedom is :",df)
```

```
##
## Degree of freedom is : 15
```

```r
cat("\np-value is :",p_value)
```

```
##
## p-value is : 0.001464336
```

```r
cat("\nTrue mean value is :",mu)
```

```
##
## True mean value is : 5
```

```r
cat("\nConfidence Interval is :",ci*100,"%")
```

```
##
## Confidence Interval is : 95 %
```

```r
cat("\nLowerbound for confidence interval is :",lowerbound)
```

```
##
## Lowerbound for confidence interval is : 5.26608
```

```r
cat("\nUpperbound for confidence interval is :",upperbound)
```

```
##
## Upperbound for confidence interval is : 5.912795
```

```r
cat("\nMean of Sample is :",xbar)
```

```
##
## Mean of Sample is : 5.589437
```

One sample t-test : One-sample t-test is used to compare the mean of one sample to a known standard (or theoretical/hypothetical) mean.

```
t.test(Southeast_Asia_Data$Happiness_score, mu = 5, conf.level = .95)
```

```
##
##  One Sample t-test
##
## data:  Southeast_Asia_Data$Happiness_score
## t = 3.8854, df = 15, p-value = 0.001464
## alternative hypothesis: true mean is not equal to 5
## 95 percent confidence interval:
##  5.266080 5.912795
## sample estimates:
## mean of x
##  5.589437
```

The p-value being less than 0.05 indicates that we reject the null hypothesis. Additionally, the confidence interval does not include the null hypothesis value of 5. Therefore, we reject the null hypothesis and conclude that the mean happiness score of countries in Southeast Asia is not equal to 5.

2. Hypothesis test for 2 mean

Creating two independent dataset.

```
Cen_East_Europe_data <- subset(data, Region == "Central and Eastern Europe")
West_Europe_data <- subset(data, Region == "Western Europe")

Europe_data = rbind(Cen_East_Europe_data, West_Europe_data)
Europe_data
```

```
##                    Country                     Region Happiness_score    GDP
##                     <char>                     <char>           <num>  <num>
##  1:                Czechia Central and Eastern Europe           6.845 10.611
##  2:               Slovakia Central and Eastern Europe           6.469 10.353
##  3:                Estonia Central and Eastern Europe           6.455 10.541
##  4:                 Kosovo Central and Eastern Europe           6.368  9.359
##  5:                 Panama Central and Eastern Europe           6.265 10.305
##  6:                 Poland Central and Eastern Europe           6.260 10.453
##  7:              Nicaragua Central and Eastern Europe           6.259  8.618
##  8:                 Latvia Central and Eastern Europe           6.213 10.370
##  9:             Kazakhstan Central and Eastern Europe           6.144 10.166
## 10:                 Serbia Central and Eastern Europe           6.144  9.854
## 11:                Croatia Central and Eastern Europe           6.125 10.341
## 12:                Hungary Central and Eastern Europe           6.041 10.419
## 13:             Montenegro Central and Eastern Europe           5.722  9.813
## 14: Bosnia and Herzegovina Central and Eastern Europe           5.633  9.616
## 15:               Bulgaria Central and Eastern Europe           5.466 10.087
## 16:                Armenia Central and Eastern Europe           5.342  9.615
## 17:                Albania Central and Eastern Europe           5.277  9.567
```

```
## 18:        North Macedonia Central and Eastern Europe           5.254  9.703
## 19:                Georgia Central and Eastern Europe           5.109  9.646
## 20:                Ukraine Central and Eastern Europe           5.071  9.314
## 21:                Finland             Western Europe           7.804 10.792
## 22:                Denmark             Western Europe           7.586 10.962
## 23:                Iceland             Western Europe           7.530 10.896
## 24:            Netherlands             Western Europe           7.403 10.942
## 25:                 Sweden             Western Europe           7.395 10.883
## 26:                 Norway             Western Europe           7.315 11.088
## 27:            Switzerland             Western Europe           7.240 11.164
## 28:             Luxembourg             Western Europe           7.228 11.660
## 29:                Austria             Western Europe           7.097 10.899
## 30:                Ireland             Western Europe           6.911 11.527
## 31:                Germany             Western Europe           6.892 10.879
## 32:                Belgium             Western Europe           6.859 10.844
## 33:              Lithuania             Western Europe           6.763 10.568
## 34:                 France             Western Europe           6.661 10.701
## 35:               Slovenia             Western Europe           6.650 10.588
## 36:                  Spain             Western Europe           6.436 10.540
## 37:                  Italy             Western Europe           6.405 10.634
## 38:               Portugal             Western Europe           5.968 10.429
## 39:                 Greece             Western Europe           5.931 10.288
##                    Country                     Region Happiness_score    GDP
##     Social_support Life_expectancy Freedom Generosity Corruption
##              <num>           <num>   <num>      <num>      <num>
##  1:          0.953          69.050   0.903      0.040      0.859
##  2:          0.953          68.838   0.753     -0.016      0.898
##  3:          0.946          69.650   0.929      0.032      0.409
##  4:          0.844          65.195   0.861      0.259      0.866
##  5:          0.896          68.950   0.855     -0.133      0.878
##  6:          0.925          69.049   0.765     -0.031      0.736
##  7:          0.853          65.650   0.877      0.021      0.625
##  8:          0.937          66.400   0.818     -0.056      0.830
##  9:          0.931          65.802   0.853      0.000      0.721
## 10:          0.873          67.088   0.845      0.204      0.816
## 11:          0.917          68.950   0.757     -0.093      0.925
## 12:          0.943          67.500   0.758     -0.059      0.839
## 13:          0.890          67.100   0.805      0.063      0.844
## 14:          0.880          67.275   0.746      0.206      0.918
## 15:          0.918          66.500   0.801     -0.057      0.911
## 16:          0.790          67.789   0.796     -0.155      0.705
## 17:          0.718          69.150   0.794     -0.007      0.878
## 18:          0.805          66.500   0.769      0.131      0.902
## 19:          0.716          64.950   0.786     -0.254      0.649
## 20:          0.878          64.550   0.795      0.240      0.907
## 21:          0.969          71.150   0.961     -0.019      0.182
## 22:          0.954          71.250   0.934      0.134      0.196
## 23:          0.983          72.050   0.936      0.211      0.668
## 24:          0.930          71.550   0.887      0.213      0.379
## 25:          0.939          72.150   0.948      0.165      0.202
## 26:          0.943          71.500   0.947      0.141      0.283
## 27:          0.920          72.900   0.891      0.027      0.266
## 28:          0.879          71.675   0.915      0.024      0.345
## 29:          0.888          71.150   0.855      0.102      0.497
```

```
## 30:          0.905          71.300   0.874      0.092      0.358
## 31:          0.896          71.300   0.846      0.030      0.420
## 32:          0.915          70.899   0.825      0.001      0.549
## 33:          0.939          67.397   0.748     -0.145      0.805
## 34:          0.909          72.300   0.819     -0.100      0.553
## 35:          0.951          71.052   0.913      0.014      0.771
## 36:          0.932          72.350   0.782     -0.066      0.711
## 37:          0.882          72.050   0.711     -0.074      0.842
## 38:          0.878          71.250   0.902     -0.196      0.878
## 39:          0.835          71.150   0.568     -0.240      0.793
##     Social_support Life_expectancy Freedom Generosity Corruption
##     Dystopia_residual
##                 <num>
##  1:             2.099
##  2:             2.078
##  3:             1.383
##  4:             2.393
##  5:             1.943
##  6:             1.772
##  7:             2.448
##  8:             1.808
##  9:             1.688
## 10:             1.881
## 11:             1.880
## 12:             1.663
## 13:             1.581
## 14:             1.635
## 15:             1.289
## 16:             1.534
## 17:             1.678
## 18:             1.435
## 19:             1.580
## 20:             1.172
## 21:             2.363
## 22:             2.084
## 23:             2.250
## 24:             2.110
## 25:             1.903
## 26:             1.829
## 27:             1.870
## 28:             1.845
## 29:             2.124
## 30:             1.545
## 31:             1.898
## 32:             1.976
## 33:             2.377
## 34:             1.872
## 35:             1.799
## 36:             1.789
## 37:             2.052
## 38:             1.556
## 39:             2.089
##     Dystopia_residual
```

Reviewing Summary of Europe

```
#summary statistics by groups
group_by(Europe_data, Region) %>% summarise(count = n(),
                                            mean = mean(Happiness_score, na.rm = TRUE),
                                            sd = sd(Happiness_score, na.rm = TRUE))
```

```
## # A tibble: 2 x 4
##   Region                      count  mean    sd
##   <chr>                       <int> <dbl> <dbl>
## 1 Central and Eastern Europe    20  5.92 0.520
## 2 Western Europe                19  6.95 0.526
```

Countries in Western Europe exhibit higher mean happiness scores (6.95 compared to those in Central and Eastern Europe (5.92), with slightly greater variability in happiness scores observed in Western Europe.

1. Are the two samples independent?

Yes, since scores from Central and Eastern European countries and Western European countries are not related.

2. To check whether the data follow a normal distribution:

2.1 Shapiro-Wilk test:

H0: NULL Hypothesis : the data are normally distributed

H1: Alternative Hypothesis : The data is not normally distributed

```
shapiro.test(Cen_East_Europe_data$Happiness_score)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Cen_East_Europe_data$Happiness_score
## W = 0.9228, p-value = 0.1122
```

```
shapiro.test(West_Europe_data$Happiness_score)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  West_Europe_data$Happiness_score
## W = 0.96348, p-value = 0.6426
```

In conclusion, since the p-values (p=0.1122for Central and Eastern Europe and p=0.6426 for Western Europe) are greater than the chosen significance level of 0.05, we fail to reject the null hypothesis. Thus, we can assume that the happiness score data for both regions follow a normal distribution.

Hypothesis test For Difference in Variance :

A hypothesis test for two independent population variances can be used to determine whether or not the data provide statistical evidence of a difference in the variance of the Happiness score between Central and Eastern European countries and Western European countries.

H0 : NULL Hypothesis : There is no significant difference between the variance of Happiness scores of Central and Eastern European countries and Western European countries.

H1 : Alternative Hypothesis : There is significant difference between the variance of Happiness scores of Central and Eastern European countries and Western European countries.

(sigma1)2 : Mean of Happiness scores of Central and Eastern European countries

(sigma2)2 : Mean of Happiness scores of Western European countries

H0: (sigma1)2 minus (sigma2)2 = 0

H1: (sigma1)2 minus (sigma2)2 != 0

Manual calculations for variance

```
#Sample 1
s1 <- sd(Cen_East_Europe_data$Happiness_score) #Sample Standard Deviation of Sample1
n1 <- nrow(Cen_East_Europe_data) #Sample size of Sample1
var1 <- s1^2 #Variance of sample 1

#Sample 2
s2 <- sd(West_Europe_data$Happiness_score) #Sample Standard Deviation of Sample2
n2 <- nrow(West_Europe_data) #Sample size of Sample2
var2 <- s2^2 #Variance of sample 2

stdError <- s1^2/s2^2 #F-Statistic

df1 <- n1-1 #Degree of freedom for sample 1
df2 <- n2-1 #Degree of freedom for sample 2

ci <- 0.95 #Confidence intercal
alpha <- 1-ci #Significance level
```

```r
fvalue1 <- qf(1-alpha/2, df1, df2) #Critical value of F-distribution for lower bound
fvalue2 <- qf(1-alpha/2, df2, df1) #Critical value of F-distribution for upper bound

lowerbound <- stdError/fvalue1 #lower bound of confidence interval
upperbound <- stdError*fvalue2 #upper bound of confidence interval

p_value <- 2*pf(stdError,df1,df2) #p-value


cat("\nF-Statistic value is :",stdError)
```

```
##
## F-Statistic value is : 0.9775079
```

```r
cat("\nDegree of freedom for sample 1 is :",df1)
```

```
##
## Degree of freedom for sample 1 is : 19
```

```r
cat("\nDegree of freedom for sample 2 is :",df2)
```

```
##
## Degree of freedom for sample 2 is : 18
```

```r
cat("\np-value is :",p_value)
```

```
##
## p-value is : 0.9581494
```

```r
cat("\nConfidence Interval is :",ci*100,"%")
```

```
##
## Confidence Interval is : 95 %
```

```r
cat("\nLowerbound for confidence interval is :",lowerbound)
```

```
##
## Lowerbound for confidence interval is : 0.3794048
```

```r
cat("\nUpperbound for confidence interval is :",upperbound)
```

```
##
## Upperbound for confidence interval is : 2.488449
```

```r
cat("\nRatio of variances is :",var1/var2)
```

```
##
## Ratio of variances is : 0.9775079
```

The hypothesis test suggests that there is no significant difference in the variance of Happiness scores between Central and Eastern European countries and Western European countries ($p > 0.05$). Therefore, we fail to reject the null hypothesis, indicating similar variability in happiness scores between the two regions.

Using F-test to compare 2 variances

```r
var.test(Cen_East_Europe_data$Happiness_score,West_Europe_data$Happiness_score,alternative = "two.sided
```

```
##
```

```
##  F test to compare two variances
##
## data:  Cen_East_Europe_data$Happiness_score and West_Europe_data$Happiness_score
## F = 0.97751, num df = 19, denom df = 18, p-value = 0.9581
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3794048 2.4884492
## sample estimates:
## ratio of variances
##          0.9775079
```

The F-test yields a p-value of p = 0.9581, exceeding the significance level alpha = 0.05. Hence, we fail to reject the null hypothesis, indicating no significant difference in variances between the two datasets. Consequently, we can proceed with the classic t-test, which assumes equality of variances.

Independent two-sample t.test with equal variance :The Independent two-samples t-test is used to compare the mean of two independent groups.

```
t.test(Happiness_score ~ Region, data = Europe_data, var.equal = TRUE, conf.level = .95)
```

```
##
##  Two Sample t-test
##
## data:  Happiness_score by Region
## t = -6.1401, df = 37, p-value = 4.058e-07
## alternative hypothesis: true difference in means between group Central and Eastern Europe and group W
## 95 percent confidence interval:
##  -1.367450 -0.688876
## sample estimates:
## mean in group Central and Eastern Europe
##                                 5.923100
##             mean in group Western Europe
##                                 6.951263
```

The obtained p-value (4.058e-07) is much less than the significance level of 0.05, leading to the rejection of the null hypothesis. Thus, we can conclude that there is a significant difference in mean Happiness scores between Central and Eastern Europe and Western Europe.

STEP 8 : ASSOCIATION BETWEEN VARIABLES

Checking coorelation between numerical variable.

Pearson Correlation Matrix

Creating a data frame for numerical variables only

```
dataNumerical <- subset(data, select=c("Happiness_score", "GDP", "Social_support", "Life_expectancy", "
```

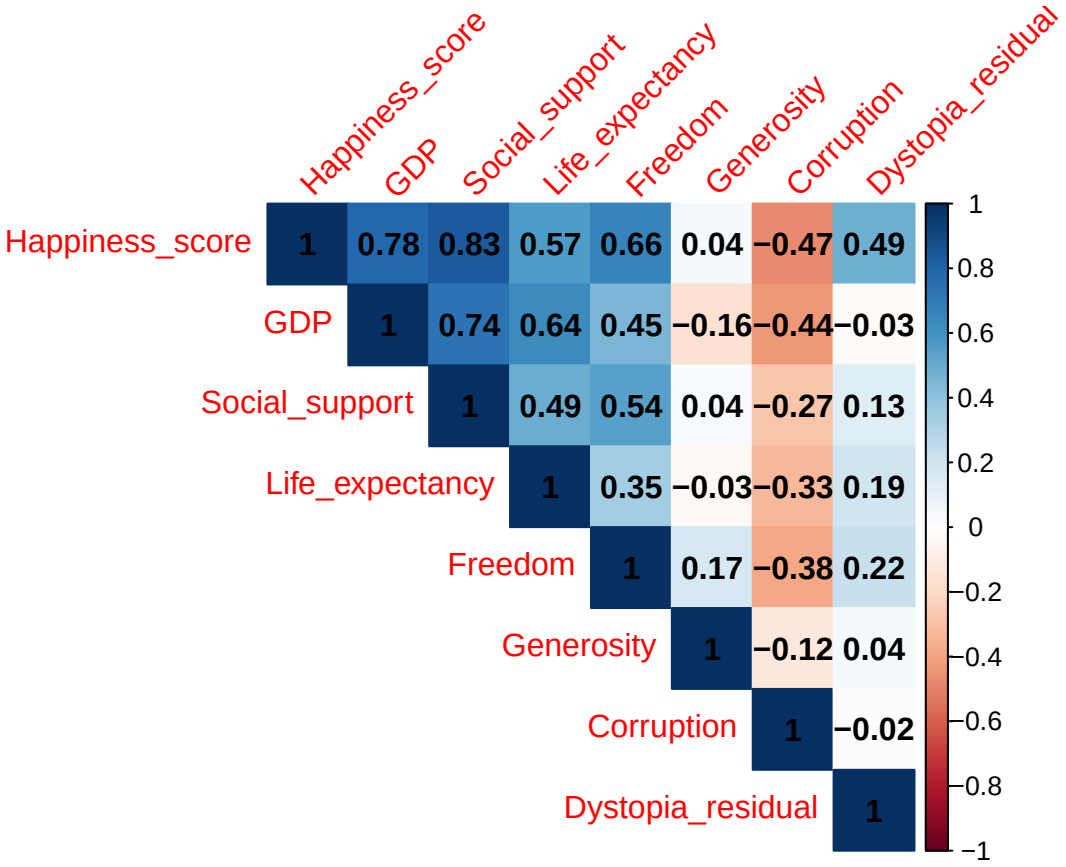
Checking coorelation between numerical variable.

```
corData <- data.frame(cor(dataNumerical))
corData
```

```
##                    Happiness_score         GDP Social_support Life_expectancy
## Happiness_score         1.00000000  0.78436731     0.83453165      0.56903586
## GDP                     0.78436731  1.00000000     0.73806877      0.63710410
## Social_support          0.83453165  0.73806877     1.00000000      0.49289581
## Life_expectancy         0.56903586  0.63710410     0.49289581      1.00000000
## Freedom                 0.66292435  0.45143893     0.54163014      0.34713387
## Generosity              0.04408178 -0.15645556     0.03657376     -0.03129528
## Corruption             -0.47191055 -0.43696084    -0.27249045     -0.32745207
## Dystopia_residual       0.48656654 -0.02746914     0.13070859      0.19140415
##                          Freedom  Generosity  Corruption Dystopia_residual
## Happiness_score        0.6629244  0.04408178 -0.47191055        0.48656654
## GDP                    0.4514389 -0.15645556 -0.43696084       -0.02746914
## Social_support         0.5416301  0.03657376 -0.27249045        0.13070859
## Life_expectancy        0.3471339 -0.03129528 -0.32745207        0.19140415
## Freedom                1.0000000  0.17022947 -0.38378630        0.22310185
## Generosity             0.1702295  1.00000000 -0.12265326        0.04274654
## Corruption            -0.3837863 -0.12265326  1.00000000       -0.01983249
## Dystopia_residual      0.2231019  0.04274654 -0.01983249        1.00000000
```

Coorelation graph

```
# Coorelation graph
library(corrplot)

# Select the columns of interest from dataNumerical
selected_columns <- dataNumerical[, c("Happiness_score", "GDP", "Social_support", "Life_expectancy", "F

# Compute the correlation matrix
correlation_matrix <- cor(selected_columns)

# Plot the correlation matrix
corrplot(correlation_matrix,
         method = "color",
         sig.level = 0.01,
         insig = "blank",
         addCoef.col = "black",
         tl.srt = 45,
         type = "upper"
)
```



> The correlation matrix indicates the strength and direction of linear relationships between variables. Strong positive correlations are observed between Happiness Score and factors like GDP and Social Support. Conversely, Corruption shows a moderate negative correlation with Happiness Score. Overall, this analysis provides insights into potential associations among these variables, aiding further exploration and modeling efforts More accurately, it has a very slight negative effect implying that if people of a country are generous the country is slightly unhappier, score wise.

Highly coorelated variable

```
library(lattice)
library(ggplot2)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
#Finding highly correlated variables
highly_correlated <- findCorrelation(cor(dataNumerical), cutoff = 0.7)
cat("Highly correlated variables are:", colnames(dataNumerical)[highly_correlated])
```

```
## Highly correlated variables are: Happiness_score GDP
```

Happiness_score and GDP is highly correlated with Happiness score.

STEP 9 : CREATING TRAINING AND TEST DATASET

Spliting the data into training and testing data

```
data1 <- subset(data, select=c("Happiness_score", "GDP", "Social_support", "Life_expectancy", "Freedom"

i <- sample(2, size=nrow(data1), replace=TRUE, prob=c(0.8, 0.2))
dataTraining <- data1[i==1,]
dataTest <- data1[i==2,]

cat("Total observations in Training data set is ",nrow(dataTraining))
```

```
## Total observations in Training data set is  112
```

```
cat("\nTotal observations in Testing data set is ",nrow(dataTest))
```

```
##
## Total observations in Testing data set is  25
```

STEP 10 : CREATING LINEAR REGRESSION MODELS

1. Simple Linear Regression Model

Constructing a simple linear regression model of Happiness Score by GDP to carry out regression on the data.

```
Simple_Linear_Model <- lm(Happiness_score~GDP, data=dataTraining)
Simple_Linear_Model_Summary <- summary(Simple_Linear_Model)
Simple_Linear_Model_Summary
```

```
##
## Call:
## lm(formula = Happiness_score ~ GDP, data = dataTraining)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.25480 -0.34486  0.05504  0.38472  2.50435
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.31283    0.50019  -2.625  0.00991 **
## GDP          0.72724    0.05212  13.954  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6667 on 110 degrees of freedom
## Multiple R-squared:  0.639,  Adjusted R-squared:  0.6357
## F-statistic: 194.7 on 1 and 110 DF,  p-value: < 2.2e-16
```

How strong is the relationship between the predictor and the response? p-value is close to zero, thus relationship is strong

Is the relationship between the predictor and the response positive or negative? The coefficient is positive and hence there is a positive relationship

Using Simple Linear Regression Model to predict Happiness Score in dataTest.

```
y_Simple_Linear_Model_Pred <-predict(object = Simple_Linear_Model, newdata = dataTest)
summary(y_Simple_Linear_Model_Pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.783   4.663   5.317   5.328   6.023   6.539
```

Predicting Test Set Results for Simple Linear Regression Model

```
Simple_Linear_Model_Pred_DF <- as.data.frame(cbind(Prediction = y_Simple_Linear_Model_Pred, Actual = da

Simple_Linear_Model_Pred_DF
```

```
##    Prediction Actual
## 1    6.535589  7.804
## 2    6.424321  7.473
## 3    6.539225  6.535
## 4    6.420684  6.405
## 5    5.316727  6.150
## 6    5.853434  6.144
## 7    6.082516  6.012
## 8    6.169058  5.931
## 9    5.441086  5.763
## 10   5.221458  5.684
## 11   5.822162  5.569
## 12   6.022882  5.466
## 13   4.574211  5.267
## 14   5.204732  5.111
## 15   4.662935  4.973
## 16   5.025830  4.908
## 17   5.212731  4.903
## 18   3.844057  4.501
## 19   4.249133  4.279
## 20   5.499266  4.170
## 21   4.267314  4.137
## 22   5.057101  4.036
## 23   4.401127  3.694
## 24   3.782969  3.207
## 25   5.579990  2.392
```

We can see that the predicted interval is varying compared to the average happiness score indicating that the prediction interval is wider than the confidence interval.

Finding RSS, R^2, MAE, MSE, RSE values for simple linear regression model.

```
library(MLmetrics)
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following objects are masked from 'package:caret':
##
##     MAE, RMSE
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

```
#MAE
Simple_Linear_Model_MAE <- MAE(y_pred = y_Simple_Linear_Model_Pred, y_true = dataTest$Happiness_score)

#MSE
Simple_Linear_Model_MSE <- MSE(y_pred = y_Simple_Linear_Model_Pred, y_true = dataTest$Happiness_score)

#RSS
Simple_Linear_Model_Residual <- resid(Simple_Linear_Model)
Simple_Linear_Model_RSS <- sum(Simple_Linear_Model_Residual^2)

#$R^2$
Simple_Linear_Model_RSquare <- Simple_Linear_Model_Summary$r.squared

#RSE
Simple_Linear_Model_RSE <- Simple_Linear_Model_Summary$sigma

cat("RSS For Simple Linear Regression Model is:",Simple_Linear_Model_RSS)
```

```
## RSS For Simple Linear Regression Model is: 48.8961
```

```
cat("\nR Squared For Simple Linear Regression Model is:",Simple_Linear_Model_RSquare)
```

```
##
## R Squared For Simple Linear Regression Model is: 0.6389981
```

```
cat("\nMAE For Simple Linear Regression Model is:",Simple_Linear_Model_MAE)
```

```
##
## MAE For Simple Linear Regression Model is: 0.5810654
```

```
cat("\nMSE For Simple Linear Regression Model is:",Simple_Linear_Model_MSE)
```

```
##
## MSE For Simple Linear Regression Model is: 0.7675939
```

```
cat("\nRSE For Simple Linear Regression Model is:",Simple_Linear_Model_RSE)
```

```
##
## RSE For Simple Linear Regression Model is: 0.6667158
```

Mean Absolute Error is the mean of summation of absolute value of actual minus predicted response. Mean Square Error is the mean of summation of square of the actual minus predicted

response value. Residual Sum of Square is the summation of square of the actual minus predicted response value. The minimum the above three value, the better the model.

Residual Standard Error is a measure of lack of fit of the model to the data. If predicted value for one or more observations is far from actual value then RSE will be large indicating a model that does not fit the data well. What constitutes a good RSE is not well defined.

R2 lies between zero and 1. Higher the value, the better. But the decision is practically made on the application.

2. Multiple Linear Regression Model

Constructing a multiple linear regression model of Happiness Score by all features to carry out regression on the data.

```
Multiple_Regression_Model <- lm(Happiness_score~. , data=dataTraining)
Multiple_Regression_Model_Summary <- summary(Multiple_Regression_Model)
Multiple_Regression_Model_Summary
```

```
##
## Call:
## lm(formula = Happiness_score ~ ., data = dataTraining)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.065e-03 -6.129e-04 -2.867e-05  6.514e-04  2.002e-03
##
## Coefficients:
##                                   Estimate Std. Error   t value Pr(>|t|)
## (Intercept)                     -3.951e+00  2.961e-03 -1334.263   <2e-16 ***
## GDP                              3.586e-01  1.758e-04  2039.809   <2e-16 ***
## Social_support                   2.525e+00  1.464e-03  1724.181   <2e-16 ***
## Life_expectancy                  2.722e-02  4.134e-05   658.520   <2e-16 ***
## Freedom                          1.331e+00  1.252e-03  1063.581   <2e-16 ***
## Generosity                       5.362e-01  8.003e-04   669.997   <2e-16 ***
## Corruption                      -7.155e-01  7.281e-04  -982.653   <2e-16 ***
## Dystopia_residual                9.997e-01  2.493e-04  4010.775   <2e-16 ***
## RegionEast Asia                 -7.176e-05  4.574e-04    -0.157   0.8757
## RegionLatin America and Caribbean  1.947e-04  3.839e-04     0.507   0.6132
## RegionMiddle East and North Africa -2.350e-04  4.243e-04    -0.554   0.5809
## RegionNorth America and ANZ      1.196e-03  5.806e-04     2.060   0.0421 *
## RegionSouth Asia                -9.447e-04  7.122e-04    -1.326   0.1878
## RegionSoutheast Asia            -1.018e-03  4.037e-04    -2.520   0.0134 *
## RegionSub-Saharan Africa        -4.369e-04  5.346e-04    -0.817   0.4158
## RegionWestern Europe             5.905e-04  4.245e-04     1.391   0.1675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0009952 on 96 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 9.117e+06 on 15 and 96 DF,  p-value: < 2.2e-16
```

Which predictors appear to have a statistically significant relationship to the response? All the predictors except Region appear to have a statistically signiticant relationship to the response.

What does one unit increase in corruption imply? A one unit increase in corruption implies that Happiness Score would reduce by 0.8335.

Now we will again construct multiple linear regression without Region variable.

```
Multiple_Regression_Model <- lm(Happiness_score~. -Region , data=dataTraining)
Multiple_Regression_Model_Summary <- summary(Multiple_Regression_Model)
Multiple_Regression_Model_Summary
```

```
##
## Call:
## lm(formula = Happiness_score ~ . - Region, data = dataTraining)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.119e-03 -6.422e-04  9.671e-05  5.836e-04  2.282e-03
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -3.954e+00  1.711e-03 -2311.0   <2e-16 ***
## GDP                3.587e-01  1.640e-04  2187.2   <2e-16 ***
## Social_support     2.526e+00  1.371e-03  1841.7   <2e-16 ***
## Life_expectancy    2.724e-02  3.134e-05   869.2   <2e-16 ***
## Freedom            1.330e+00  1.173e-03  1134.1   <2e-16 ***
## Generosity         5.361e-01  7.657e-04   700.2   <2e-16 ***
## Corruption        -7.162e-01  6.593e-04 -1086.4   <2e-16 ***
## Dystopia_residual  1.000e+00  2.298e-04  4351.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00104 on 104 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.789e+07 on 7 and 104 DF,  p-value: < 2.2e-16
```

Using Multiple Linear Regression Model to predict Happiness Score in dataTest.

```
y_Multiple_Regression_Model_Pred <-predict(object = Multiple_Regression_Model, newdata = dataTest)
summary(y_Multiple_Regression_Model_Pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.596   4.170   5.268   5.088   6.014   7.804
```

Predicting Test Set Results for Multiple Linear Regression Model

```
Multiple_Regression_Model_Pred_DF <- as.data.frame(cbind(Prediction = y_Multiple_Regression_Model_Pred,

Multiple_Regression_Model_Pred_DF
```

```
##    Prediction Actual
```

```
## 1    7.803657  7.804
## 2    7.472208  7.473
## 3    6.534300  6.535
## 4    6.405989  6.405
## 5    6.151120  6.150
## 6    6.143240  6.144
## 7    6.014002  6.012
## 8    5.931531  5.931
## 9    5.762694  5.763
## 10   5.682911  5.684
## 11   5.568443  5.569
## 12   5.465886  5.466
## 13   5.268323  5.267
## 14   5.110813  5.111
## 15   4.973311  4.973
## 16   1.595692  4.908
## 17   4.902488  4.903
## 18   4.501157  4.501
## 19   4.278395  4.279
## 20   4.169764  4.170
## 21   4.136125  4.137
## 22   4.035716  4.036
## 23   3.694384  3.694
## 24   3.208868  3.207
## 25   2.393127  2.392
```

We can see that the predicted interval is varying compared to the average happiness score indicating that the prediction interval is wider than the confidence interval.

Finding RSS, R^2, MAE and MSE values for Multiple linear regression model.

```
library(MLmetrics)

#MAE
Multiple_Regression_Model_MAE <- MAE(y_pred = y_Multiple_Regression_Model_Pred, y_true = dataTest$Happi

#MSE
Multiple_Regression_Model_MSE <- MSE(y_pred = y_Multiple_Regression_Model_Pred, y_true = dataTest$Happi

#RSS
Multiple_Regression_Model_Residual <- resid(Multiple_Regression_Model)
Multiple_Regression_Model_RSS <- sum(Multiple_Regression_Model_Residual^2)

#$R^2$
Multiple_Regression_Model_RSquare <- Multiple_Regression_Model_Summary$r.squared

#RSE
Multiple_Regression_Model_RSE <- Multiple_Regression_Model_Summary$sigma


cat("RSS For Multiple Linear Regression Model is:",Multiple_Regression_Model_RSS)
```

```
## RSS For Multiple Linear Regression Model is: 0.0001124967
```

```
cat("\nR Squared For Multiple Linear Regression Model is:",Multiple_Regression_Model_RSquare)
```

```
##
## R Squared For Multiple Linear Regression Model is: 0.9999992
```

```
cat("\nMAE For Multiple Linear Regression Model is:",Multiple_Regression_Model_MAE)
```

```
##
## MAE For Multiple Linear Regression Model is: 0.1331792
```

```
cat("\nMSE For Multiple Linear Regression Model is:",Multiple_Regression_Model_MSE)
```

```
##
## MSE For Multiple Linear Regression Model is: 0.4388561
```

```
cat("\nRSE For Multiple Linear Regression Model is:",Multiple_Regression_Model_RSE)
```

```
##
## RSE For Multiple Linear Regression Model is: 0.001040048
```

The MSE, MAE, RSE are too small. Suggesting a good model or an overfit. Since they were calculated on testing data, we do not think it is overfitting and conclude it is a good model.

3. Forward Stepwise Subset Selection Linear Regression Model

Begins with Null Model Fit p simple linear regressions and add to the null model the variable that results in the lowest RSS. Add to that model the variable that results in the lowest RSS amongst all two-variable models. Continue until some stopping rule is satisfied, for example when all remaining variables have a p-value above some threshold.

```
library(MASS)
# Create a null model
forward_intercept_only <- lm(Happiness_score ~ 1, data=dataTraining)
# Create a full model
forward_all <- lm(Happiness_score~., data=dataTraining)
# perform forward step-wise regression
Forward_Regression_Model <- stepAIC (forward_intercept_only, direction='forward',scope = formula(forwar
```

```
## Start:  AIC=23.29
## Happiness_score ~ 1
##
##                     Df Sum of Sq     RSS      AIC
## + Social_support     1    93.990  41.456 -107.314
## + Region             8    98.061  37.384 -104.892
## + GDP                1    86.549  48.896  -88.826
## + Life_expectancy    1    78.702  56.744  -72.155
## + Freedom            1    66.046  69.400  -49.605
## + Corruption         1    33.344 102.102   -6.363
## + Dystopia_residual  1    25.535 109.911    1.891
## <none>                           135.446   23.288
## + Generosity         1     0.696 134.749   24.711
##
## Step:  AIC=-107.31
## Happiness_score ~ Social_support
##
##                     Df Sum of Sq    RSS     AIC
```

```
## + Dystopia_residual  1   16.4983 24.957 -162.15
## + Region             8   15.2432 26.213 -142.65
## + Corruption         1   10.8586 30.597 -139.33
## + GDP                1   10.7372 30.719 -138.89
## + Freedom            1    9.7599 31.696 -135.38
## + Life_expectancy    1    7.6542 33.802 -128.18
## <none>                           41.456 -107.31
## + Generosity         1    0.0277 41.428 -105.39
##
## Step:  AIC=-162.15
## Happiness_score ~ Social_support + Dystopia_residual
##
##                   Df Sum of Sq     RSS     AIC
## + GDP              1   18.6459  6.3115 -314.13
## + Life_expectancy  1   11.9795 12.9779 -233.39
## + Corruption       1   11.9247 13.0327 -232.92
## + Region           8   11.1161 13.8413 -212.17
## + Freedom          1    6.0054 18.9520 -190.98
## <none>                         24.9574 -162.15
## + Generosity       1    0.0171 24.9403 -160.22
##
## Step:  AIC=-314.13
## Happiness_score ~ Social_support + Dystopia_residual + GDP
##
##                   Df Sum of Sq    RSS     AIC
## + Corruption       1    3.3770 2.9345 -397.90
## + Freedom          1    2.8929 3.4186 -380.80
## + Generosity       1    1.6838 4.6277 -346.88
## + Region           8    2.1511 4.1604 -344.80
## + Life_expectancy  1    1.1804 5.1311 -335.32
## <none>                         6.3115 -314.13
##
## Step:  AIC=-397.9
## Happiness_score ~ Social_support + Dystopia_residual + GDP +
##     Corruption
##
##                   Df Sum of Sq    RSS     AIC
## + Freedom          1   1.71246 1.2220 -494.01
## + Generosity       1   0.69887 2.2356 -426.37
## + Life_expectancy  1   0.68202 2.2525 -425.52
## + Region           8   0.87453 2.0600 -421.53
## <none>                         2.9345 -397.90
##
## Step:  AIC=-494.01
## Happiness_score ~ Social_support + Dystopia_residual + GDP +
##     Corruption + Freedom
##
##                   Df Sum of Sq     RSS     AIC
## + Life_expectancy  1   0.69160 0.53042 -585.49
## + Generosity       1   0.40475 0.81727 -537.07
## + Region           8   0.36483 0.85720 -517.73
## <none>                         1.22203 -494.01
##
## Step:  AIC=-585.49
```

```
## Happiness_score ~ Social_support + Dystopia_residual + GDP +
##     Corruption + Freedom + Life_expectancy
##
##              Df Sum of Sq     RSS      AIC
## + Generosity  1   0.53031 0.00011 -1530.84
## + Region      8   0.08572 0.44471  -589.23
## <none>                    0.53042  -585.49
##
## Step:  AIC=-1530.84
## Happiness_score ~ Social_support + Dystopia_residual + GDP +
##     Corruption + Freedom + Life_expectancy + Generosity
##
##          Df  Sum of Sq        RSS     AIC
## + Region  8 1.7412e-05 9.5084e-05 -1533.7
## <none>                1.1250e-04 -1530.8
##
## Step:  AIC=-1533.68
## Happiness_score ~ Social_support + Dystopia_residual + GDP +
##     Corruption + Freedom + Life_expectancy + Generosity + Region
```

Viewing Results of Forward Stepwise Subset Selection Linear Regression Model

```
# view results of forward stepwise regression
Forward_Regression_Model$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## Happiness_score ~ 1
##
## Final Model:
## Happiness_score ~ Social_support + Dystopia_residual + GDP +
##     Corruption + Freedom + Life_expectancy + Generosity + Region
##
##
##                  Step Df     Deviance Resid. Df   Resid. Dev         AIC
## 1                                          111 1.354455e+02    23.28793
## 2    + Social_support  1 9.398979e+01       110 4.145575e+01  -107.31369
## 3 + Dystopia_residual  1 1.649835e+01       109 2.495740e+01  -162.14877
## 4               + GDP  1 1.864594e+01       108 6.311465e+00  -314.12667
## 5        + Corruption  1 3.376982e+00       107 2.934484e+00  -397.90034
## 6           + Freedom  1 1.712456e+00       106 1.222027e+00  -494.01462
## 7   + Life_expectancy  1 6.916023e-01       105 5.304250e-01  -585.48847
## 8        + Generosity  1 5.303125e-01       104 1.124967e-04 -1530.84154
## 9            + Region  8 1.741239e-05        96 9.508435e-05 -1533.67544
```

Viewing summary for Forward Stepwise Subset Selection Linear Regression Model

```
# view final model
Forward_Regression_Model_Summary <- summary(Forward_Regression_Model)
Forward_Regression_Model_Summary
```

```
##
## Call:
## lm(formula = Happiness_score ~ Social_support + Dystopia_residual +
##     GDP + Corruption + Freedom + Life_expectancy + Generosity +
##     Region, data = dataTraining)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.065e-03 -6.129e-04 -2.867e-05  6.514e-04  2.002e-03
##
## Coefficients:
##                                    Estimate Std. Error   t value Pr(>|t|)
## (Intercept)                       -3.951e+00  2.961e-03 -1334.263   <2e-16 ***
## Social_support                     2.525e+00  1.464e-03  1724.181   <2e-16 ***
## Dystopia_residual                  9.997e-01  2.493e-04  4010.775   <2e-16 ***
## GDP                                3.586e-01  1.758e-04  2039.809   <2e-16 ***
## Corruption                        -7.155e-01  7.281e-04  -982.653   <2e-16 ***
## Freedom                            1.331e+00  1.252e-03  1063.581   <2e-16 ***
## Life_expectancy                    2.722e-02  4.134e-05   658.520   <2e-16 ***
## Generosity                         5.362e-01  8.003e-04   669.997   <2e-16 ***
## RegionEast Asia                   -7.176e-05  4.574e-04    -0.157   0.8757
## RegionLatin America and Caribbean  1.947e-04  3.839e-04     0.507   0.6132
## RegionMiddle East and North Africa -2.350e-04  4.243e-04    -0.554   0.5809
## RegionNorth America and ANZ        1.196e-03  5.806e-04     2.060   0.0421 *
## RegionSouth Asia                  -9.447e-04  7.122e-04    -1.326   0.1878
## RegionSoutheast Asia              -1.018e-03  4.037e-04    -2.520   0.0134 *
## RegionSub-Saharan Africa          -4.369e-04  5.346e-04    -0.817   0.4158
## RegionWestern Europe               5.905e-04  4.245e-04     1.391   0.1675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0009952 on 96 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 9.117e+06 on 15 and 96 DF,  p-value: < 2.2e-16
```

Using Forward Stepwise Subset Selection Linear Regression Model to predict Happiness Score in dataTest.

```
y_Forward_Regression_Model_Pred <-predict(object = Forward_Regression_Model, newdata = dataTest)
summary(y_Forward_Regression_Model_Pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.598   4.170   5.268   5.088   6.013   7.803
```

Predicting Test Set Results for Forward Stepwise Subset Selection Linear Regression Model

```
Forward_Regression_Model_Pred_DF <- as.data.frame(cbind(Prediction = y_Forward_Regression_Model_Pred, A

Forward_Regression_Model_Pred_DF
```

```
##    Prediction Actual
## 1    7.803499  7.804
## 2    7.471322  7.473
## 3    6.533931  6.535
## 4    6.406250  6.405
## 5    6.151337  6.150
## 6    6.143322  6.144
## 7    6.013138  6.012
## 8    5.931676  5.931
## 9    5.761964  5.763
## 10   5.683335  5.684
## 11   5.568815  5.569
## 12   5.466159  5.466
## 13   5.268023  5.267
## 14   5.110324  5.111
## 15   4.973142  4.973
## 16   1.597545  4.908
## 17   4.902557  4.903
## 18   4.501250  4.501
## 19   4.278444  4.279
## 20   4.170014  4.170
## 21   4.136170  4.137
## 22   4.035718  4.036
## 23   3.694876  3.694
## 24   3.209378  3.207
## 25   2.393794  2.392
```

The prediction interval is substantially wider than the confidence interval, reflecting the increased uncertainty about Happiness Score in comparison to the average Happiness score.

Finding RSS, R^2, MAE and MSE values for Forward Stepwise Subset Selection Linear Regression Model

```
library(MLmetrics)

#MAE
Forward_Regression_Model_MAE <- MAE(y_pred = y_Forward_Regression_Model_Pred, y_true = dataTest$Happines

#MSE
Forward_Regression_Model_MSE <- MSE(y_pred = y_Forward_Regression_Model_Pred, y_true = dataTest$Happines

#RSS
Forward_Regression_Model_Residual <- resid(Forward_Regression_Model)
Forward_Regression_Model_RSS <- sum(Forward_Regression_Model_Residual^2)
```

```r
#$R^2$
Forward_Regression_Model_RSquare <- Forward_Regression_Model_Summary$r.squared

#RSE
Forward_Regression_Model_RSE <- Forward_Regression_Model_Summary$sigma

cat("RSS For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_Model_RSS
```

```
## RSS For Forward Stepwise Subset Selection Linear Regression Model is: 9.508435e-05
```

```r
cat("\nR Squared For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_
```

```
##
## R Squared For Forward Stepwise Subset Selection Linear Regression Model is: 0.9999993
```

```r
cat("\nMAE For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_Model_
```

```
##
## MAE For Forward Stepwise Subset Selection Linear Regression Model is: 0.1332036
```

```r
cat("\nMSE For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_Model_
```

```
##
## MSE For Forward Stepwise Subset Selection Linear Regression Model is: 0.4383655
```

```r
cat("\nRSE For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_Model_
```

```
##
## RSE For Forward Stepwise Subset Selection Linear Regression Model is: 0.0009952196
```

4. Backward Stepwise Subset Selection Linear Regression Model

> Start with all variables in the model. Remove the variable with the largest p value that is, the variable that is the least statistically significant. The new (p1) variable model is fit, and the variable with the largest p-value is removed. Continue until a stopping rule is reached.

```r
# Create a full model
backward_all <- lm(Happiness_score~., data=dataTraining)
# perform Backward step-wise regression
Backward_Regression_Model <- stepAIC (backward_all, direction='backward')
```

```
## Start:  AIC=-1533.68
## Happiness_score ~ GDP + Social_support + Life_expectancy + Freedom +
##     Generosity + Corruption + Dystopia_residual + Region
##
##                     Df Sum of Sq     RSS      AIC
## <none>                           0.0001 -1533.68
## - Region             8    0.0000  0.0001 -1530.84
## - Life_expectancy    1    0.4295  0.4296  -593.10
## - Generosity         1    0.4446  0.4447  -589.23
## - Corruption         1    0.9564  0.9565  -503.45
## - Freedom            1    1.1204  1.1205  -485.73
## - Social_support     1    2.9444  2.9445  -377.52
## - GDP                1    4.1211  4.1212  -339.86
## - Dystopia_residual  1   15.9329 15.9330  -188.41
```

Viewing Results of Backward Stepwise Subset Selection Linear Regression Model

```
# view results of backward stepwise regression
Backward_Regression_Model$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## Happiness_score ~ GDP + Social_support + Life_expectancy + Freedom +
##     Generosity + Corruption + Dystopia_residual + Region
##
## Final Model:
## Happiness_score ~ GDP + Social_support + Life_expectancy + Freedom +
##     Generosity + Corruption + Dystopia_residual + Region
##
##
##   Step Df Deviance Resid. Df   Resid. Dev       AIC
## 1                        96 9.508435e-05 -1533.675
```

Viewing summary for Backward Stepwise Subset Selection Linear Regression Model

```
# view final model
Backward_Regression_Model_Summary <- summary(Backward_Regression_Model)
Backward_Regression_Model_Summary
```

```
##
## Call:
## lm(formula = Happiness_score ~ GDP + Social_support + Life_expectancy +
##     Freedom + Generosity + Corruption + Dystopia_residual + Region,
##     data = dataTraining)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -2.065e-03 -6.129e-04 -2.867e-05  6.514e-04  2.002e-03
##
## Coefficients:
##                                          Estimate Std. Error   t value Pr(>|t|)
## (Intercept)                            -3.951e+00  2.961e-03 -1334.263   <2e-16 ***
## GDP                                     3.586e-01  1.758e-04  2039.809   <2e-16 ***
## Social_support                          2.525e+00  1.464e-03  1724.181   <2e-16 ***
## Life_expectancy                         2.722e-02  4.134e-05   658.520   <2e-16 ***
## Freedom                                 1.331e+00  1.252e-03  1063.581   <2e-16 ***
## Generosity                              5.362e-01  8.003e-04   669.997   <2e-16 ***
## Corruption                             -7.155e-01  7.281e-04  -982.653   <2e-16 ***
## Dystopia_residual                       9.997e-01  2.493e-04  4010.775   <2e-16 ***
## RegionEast Asia                        -7.176e-05  4.574e-04    -0.157   0.8757
## RegionLatin America and Caribbean       1.947e-04  3.839e-04     0.507   0.6132
## RegionMiddle East and North Africa     -2.350e-04  4.243e-04    -0.554   0.5809
## RegionNorth America and ANZ             1.196e-03  5.806e-04     2.060   0.0421 *
```

```
## RegionSouth Asia                   -9.447e-04  7.122e-04    -1.326   0.1878
## RegionSoutheast Asia               -1.018e-03  4.037e-04    -2.520   0.0134 *
## RegionSub-Saharan Africa           -4.369e-04  5.346e-04    -0.817   0.4158
## RegionWestern Europe                5.905e-04  4.245e-04     1.391   0.1675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0009952 on 96 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 9.117e+06 on 15 and 96 DF,  p-value: < 2.2e-16
```

Using Backward Stepwise Subset Selection Linear Regression Model to predict Happiness Score in dataTest.

```
y_Backward_Regression_Model_Pred <-predict(object = Backward_Regression_Model, newdata = dataTest)
summary(y_Backward_Regression_Model_Pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.598   4.170   5.268   5.088   6.013   7.803
```

Using Backward Stepwise Subset Selection Linear Regression Model to predict Confidence Interval on Happiness Score in dataTest.

```
y_Backward_Regression_Model_Pred_conf <-predict(object = Backward_Regression_Model, newdata = dataTest,
summary(y_Backward_Regression_Model_Pred_conf)
```

```
##       fit             lwr             upr
##  Min.   :1.598   Min.   :1.592   Min.   :1.603
##  1st Qu.:4.170   1st Qu.:4.169   1st Qu.:4.171
##  Median :5.268   Median :5.267   Median :5.269
##  Mean   :5.088   Mean   :5.087   Mean   :5.089
##  3rd Qu.:6.013   3rd Qu.:6.012   3rd Qu.:6.014
##  Max.   :7.803   Max.   :7.803   Max.   :7.804
```

Predicting Test Set Results for Backward Stepwise Subset Selection Linear Regression Model

```
Backward_Regression_Model_Pred_DF <- as.data.frame(cbind(Prediction = y_Backward_Regression_Model_Pred,

Backward_Regression_Model_Pred_DF
```

```
##    Prediction Actual
## 1    7.803499  7.804
## 2    7.471322  7.473
## 3    6.533931  6.535
## 4    6.406250  6.405
## 5    6.151337  6.150
## 6    6.143322  6.144
## 7    6.013138  6.012
## 8    5.931676  5.931
```

```
## 9    5.761964  5.763
## 10   5.683335  5.684
## 11   5.568815  5.569
## 12   5.466159  5.466
## 13   5.268023  5.267
## 14   5.110324  5.111
## 15   4.973142  4.973
## 16   1.597545  4.908
## 17   4.902557  4.903
## 18   4.501250  4.501
## 19   4.278444  4.279
## 20   4.170014  4.170
## 21   4.136170  4.137
## 22   4.035718  4.036
## 23   3.694876  3.694
## 24   3.209378  3.207
## 25   2.393794  2.392
```

The prediction interval is substantially wider than the confidence interval, reflecting the increased uncertainty about Happiness Score in comparison to the average Happiness score.

Finding RSS, R^2, MAE and MSE values for Backward Stepwise Subset Selection Linear Regression Model

```
library(MLmetrics)

#MAE
Backward_Regression_Model_MAE <- MAE(y_pred = y_Backward_Regression_Model_Pred, y_true = dataTest$Happi

#MSE
Backward_Regression_Model_MSE <- MSE(y_pred = y_Backward_Regression_Model_Pred, y_true = dataTest$Happi

#RSS
Backward_Regression_Model_Residual <- resid(Backward_Regression_Model)
Backward_Regression_Model_RSS <- sum(Backward_Regression_Model_Residual^2)

#$R^2$
Backward_Regression_Model_RSquare <- Backward_Regression_Model_Summary$r.squared

#RSE
Backward_Regression_Model_RSE <- Backward_Regression_Model_Summary$sigma


cat("RSS For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regression_Model_
```

```
## RSS For Backward Stepwise Subset Selection Linear Regression Model is: 9.508435e-05
```

```
cat("\nR Squared For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regressio
```

```
##
## R Squared For Backward Stepwise Subset Selection Linear Regression Model is: 0.9999993
```

```
cat("\nMAE For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regression_Model
```

```
##
```

```
## MAE For Backward Stepwise Subset Selection Linear Regression Model is: 0.1332036
```

```
cat("\nMSE For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regression_Model
```

```
##
## MSE For Backward Stepwise Subset Selection Linear Regression Model is: 0.4383655
```

```
cat("\nRSE For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regression_Model
```

```
##
## RSE For Backward Stepwise Subset Selection Linear Regression Model is: 0.0009952196
```

STEP 11 : MODEL ASSESSMENT

1. Simple Linear Regression Model

```
cat("RSS For Simple Linear Regression Model is:",Simple_Linear_Model_RSS)
```

```
## RSS For Simple Linear Regression Model is: 48.8961
```

```
cat("\nR Squared For Simple Linear Regression Model is:",Simple_Linear_Model_RSquare)
```

```
##
## R Squared For Simple Linear Regression Model is: 0.6389981
```

```
cat("\nMAE For Simple Linear Regression Model is:",Simple_Linear_Model_MAE)
```

```
##
## MAE For Simple Linear Regression Model is: 0.5810654
```

```
cat("\nMSE For Simple Linear Regression Model is:",Simple_Linear_Model_MSE)
```

```
##
## MSE For Simple Linear Regression Model is: 0.7675939
```

```
cat("\nRSE For Simple Linear Regression Model is:",Simple_Linear_Model_RSE)
```

```
##
## RSE For Simple Linear Regression Model is: 0.6667158
```

2. Multiple Linear Regression Model

```
cat("RSS For Multiple Linear Regression Model is:",Multiple_Regression_Model_RSS)
```

```
## RSS For Multiple Linear Regression Model is: 0.0001124967
```

```
cat("\nR Squared For Multiple Linear Regression Model is:",Multiple_Regression_Model_RSquare)
```

```
##
## R Squared For Multiple Linear Regression Model is: 0.9999992
```

```
cat("\nMAE For Multiple Linear Regression Model is:",Multiple_Regression_Model_MAE)
```

```
##
## MAE For Multiple Linear Regression Model is: 0.1331792
```

```
cat("\nMSE For Multiple Linear Regression Model is:",Multiple_Regression_Model_MSE)
```

```
##
## MSE For Multiple Linear Regression Model is: 0.4388561
cat("\nRSE For Multiple Linear Regression Model is:",Multiple_Regression_Model_RSE)

##
## RSE For Multiple Linear Regression Model is: 0.001040048
```

3. Forward Stepwise subset selection Linear Regression Model

```
cat("RSS For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_Model_RSS

## RSS For Forward Stepwise Subset Selection Linear Regression Model is: 9.508435e-05
cat("\nR Squared For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_M

##
## R Squared For Forward Stepwise Subset Selection Linear Regression Model is: 0.9999993
cat("\nMAE For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_Model_M

##
## MAE For Forward Stepwise Subset Selection Linear Regression Model is: 0.1332036
cat("\nMSE For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_Model_M

##
## MSE For Forward Stepwise Subset Selection Linear Regression Model is: 0.4383655
cat("\nRSE For Forward Stepwise Subset Selection Linear Regression Model is:",Forward_Regression_Model_M

##
## RSE For Forward Stepwise Subset Selection Linear Regression Model is: 0.0009952196
```

4. Backward Stepwise subset selection Linear Regression Model

```
cat("RSS For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regression_Model_M

## RSS For Backward Stepwise Subset Selection Linear Regression Model is: 9.508435e-05
cat("\nR Squared For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regressio

##
## R Squared For Backward Stepwise Subset Selection Linear Regression Model is: 0.9999993
cat("\nMAE For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regression_Model

##
## MAE For Backward Stepwise Subset Selection Linear Regression Model is: 0.1332036
cat("\nMSE For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regression_Model

##
## MSE For Backward Stepwise Subset Selection Linear Regression Model is: 0.4383655
cat("\nRSE For Backward Stepwise Subset Selection Linear Regression Model is:",Backward_Regression_Model

##
## RSE For Backward Stepwise Subset Selection Linear Regression Model is: 0.0009952196
```

Based on the given metrics, it seems that all four models perform well, but they are designed to answer different questions.

The Simple Linear Regression model uses only one predictor variable, and it appears to have the lowest performance compared to the other models, with a higher RSS, lower R-squared, higher MAE, and higher MSE, higher RSE. This model is useful when we want to study the relationship between two variables and see how a change in one variable affects the other.

The Multiple Linear Regression model uses multiple predictor variables, and it has a lower RSS, higher R-squared, lower MAE, lower MSE and lower RSE than the Simple Linear Regression model. This model is helpful when we want to study the relationship between a response variable and multiple predictor variables and see how these predictors affect the response variable.

The Forward Stepwise and Backward Stepwise Subset Selection Linear Regression models both use a subset of predictor variables, and they have the same performance metrics as the Multiple Linear Regression model. These models are useful when we want to identify a subset of predictors that best explain the variability in the response variable.

Given that all four models have similar performance, the choice of which model to use depends on the research question and the available data. If the research question involves only one predictor variable, then the Simple Linear Regression model is appropriate. If the research question involves multiple predictors, then the Multiple Linear Regression model or one of the Subset Selection models may be more appropriate. If we are interested in identifying the most important predictors, then we should use one of the Subset Selection models