Predicting Happiness Scores

2022-10-04

Abstract

This project will analyse the data to seek possible correlations between happiness scores and other criteria being test. I will look at all countries test, and then consider separately the happiest and least happy countries. Finally, I will try to predict the happiness score of 175 countries. I will do this using a variety of methods: by guessing, by using a correlation line, and finally by using the Root Mean Squared Error, RMSE.

The data scores 12 categories, one of which is the overall happiness score, over the course of seven years, from 2015 through 2022. The raw data is located at https://www.kaggle.com/datasets/mathurinache/world-happiness-report. I combined all years into one file which is downloaded and read by the script. It can also be viewed at https://github.com/Oanalkd/Happiness/blob/main/Happiness.csv.

What we will see is that on average happiness scores are increasing, although for some of the unhappiest ones, they are not. On average, not at the highest and lowest scores, the highest correlation occur between happiness and GDP, health, and freedom. A negative correlation occurs between happiness and generosity at both the top and bottom of the spectrum.

To predict the happiness levels, I train the model on years 2015 - 2021 and test it on year 2022 for an rmse of .41.

```r
#install required packages
if(!require(data.table)) install.packages("tinytex", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: data.table
```

```r
tinytex::install_tinytex(force = TRUE)
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::transpose() masks data.table::transpose()
```

```r
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caret
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(gridExtra)) install.packages("gridExtra", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: gridExtra
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
# remove all variables
rm (list=ls())

#Reading the data shows that there are 1230 records and 13 dimensions.
url<-"https://raw.githubusercontent.com/0analkd/Happiness/main/Happiness.csv"
Happiness<-read.csv(url)
dim(Happiness)
```

```
## [1] 1230   13
```

```r
names(Happiness)
```

```
##  [1] "Year"                          "Country"
##  [3] "Happiness.Rank"                "Happiness.Score"
##  [5] "Economy..GDP.per.Capita."      "Family"
##  [7] "Social.support"                "Health..Life.Expectancy."
##  [9] "Freedom"                       "Trust..Government.Corruption."
## [11] "Generosity"                    "Dystopia.Residual"
## [13] "Perceptions.of.corruption"
```

```r
#Create new fields that will be needed
Happiness$CtryYear<-paste(Happiness$Year, Happiness$Country, sep="_")
Happiness$GdpHealth<-(Happiness$Economy..GDP.per.Capita.+Happiness$Health..Life.Expectancy.)/2
Happiness$GdpHealthFree<-(Happiness$Economy..GDP.per.Capita.+Happiness$Health..Life.Expectancy.+Happines


#There are 175 distinct countries
Happiness%>%summarise(countries=n_distinct(Country))
```

```
##   countries
## 1       175
```

```
#Which country/year have the highest/lower scores
Happiness$CtryYear[which.max(Happiness$Happiness.Score)]
```

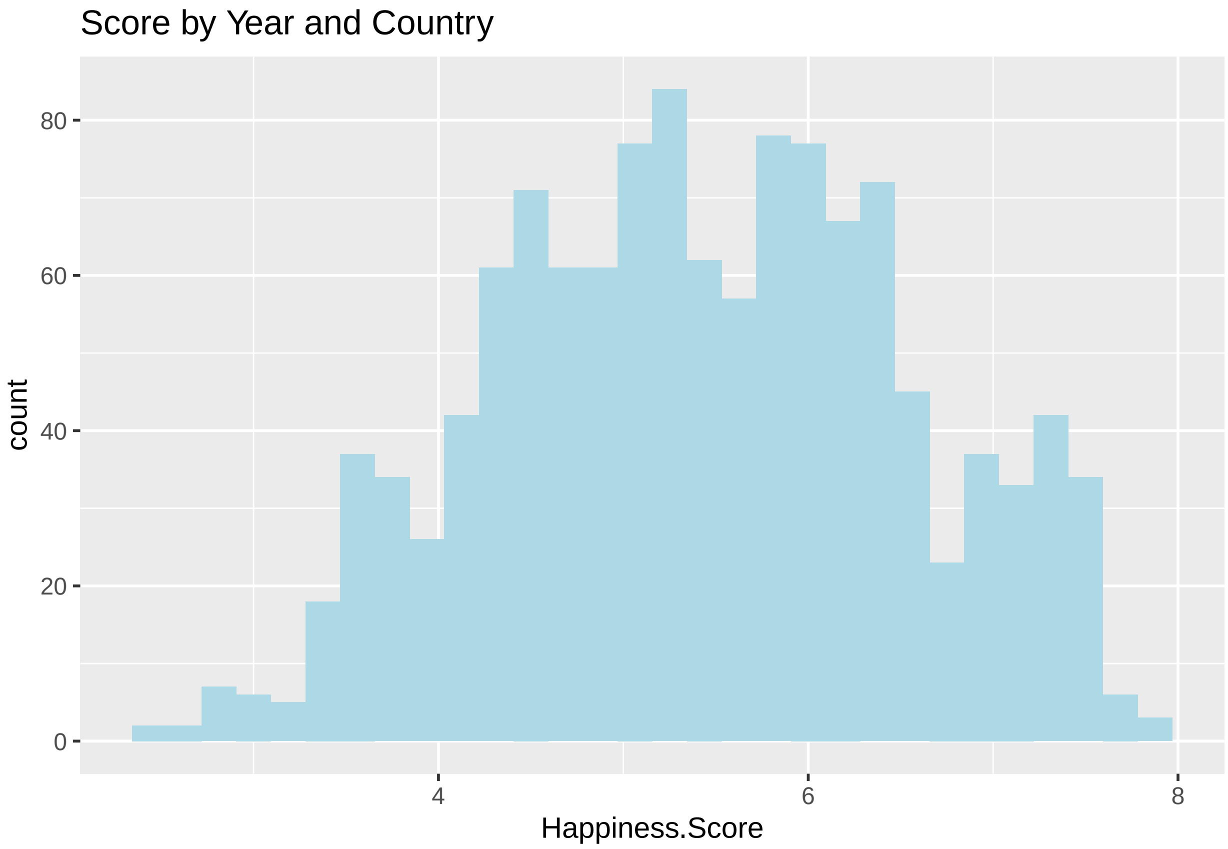
```
## [1] "2021_Finland"
```

```
Happiness$CtryYear[which.min(Happiness$Happiness.Score)]
```

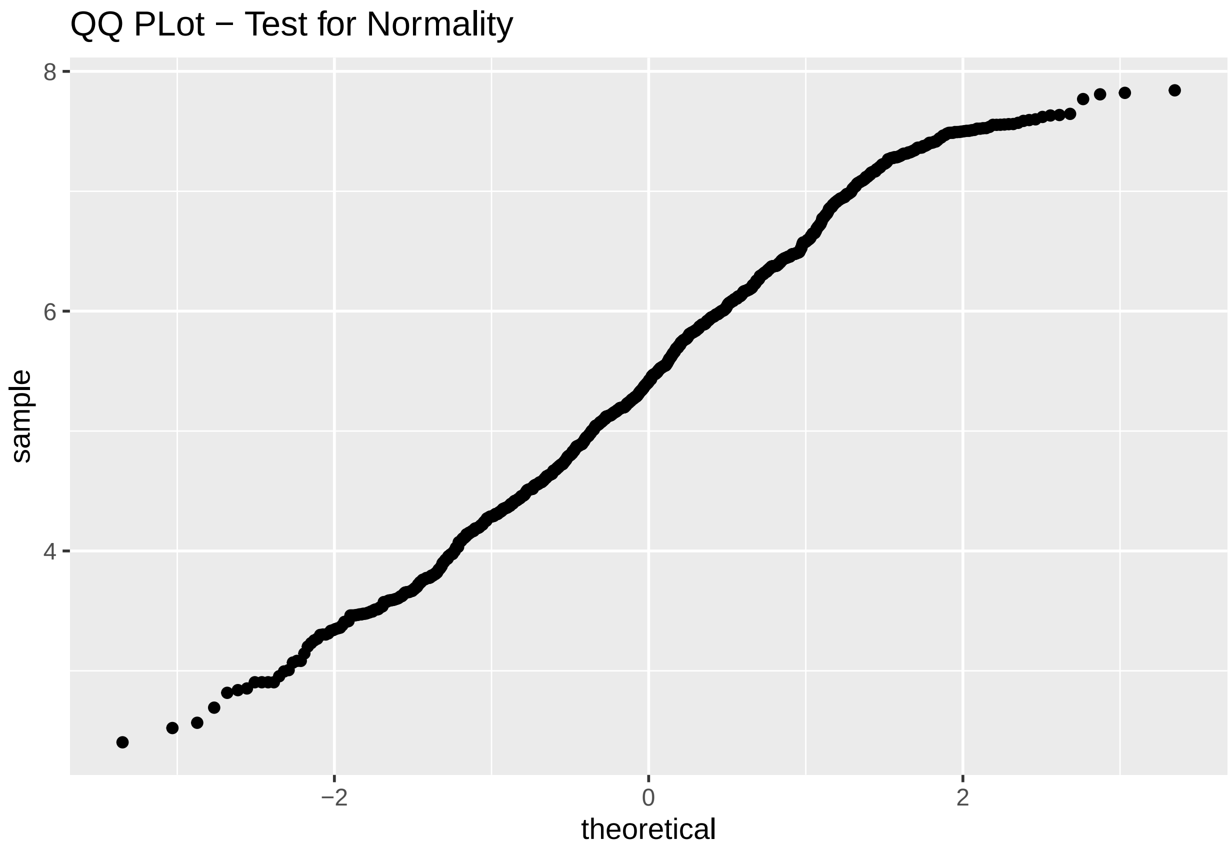
```
## [1] "2022_Afghanistan"
```

```
#Distribution of happiness rating by country AND year
plot1<-Happiness %>% group_by(Country)%>%
  ggplot(aes(Happiness.Score)) +
  geom_histogram(fill = "light blue")+
  ggtitle("Score by Year and Country")
plot1
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

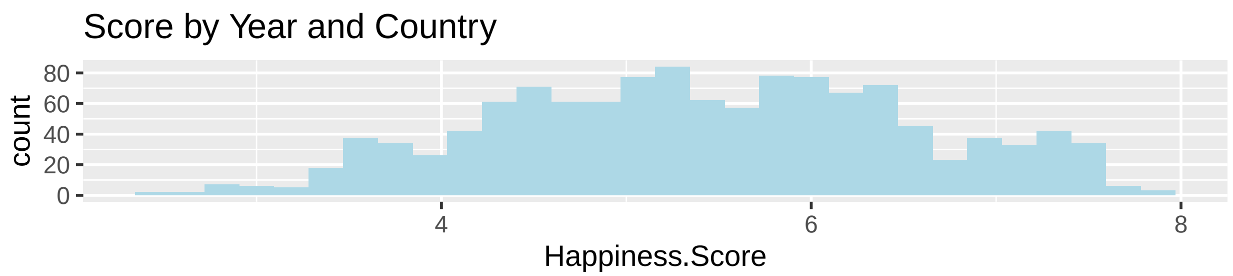


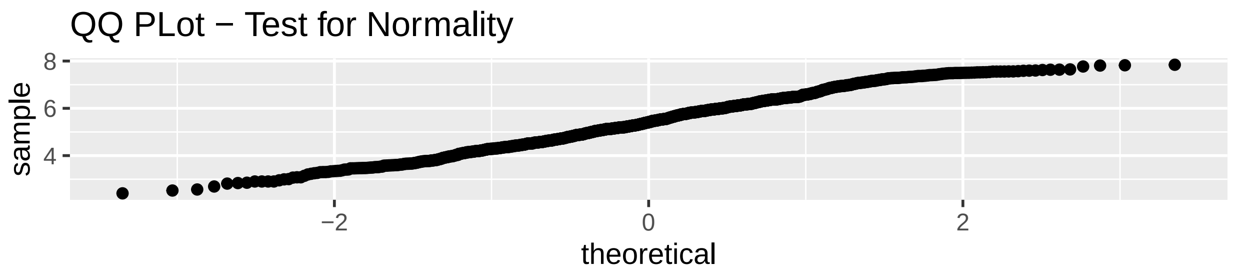
```
#We can see that the distribution is close to normal and therefore can be well described by the mean an
plot2<-ggplot(data=Happiness,aes(sample=Happiness.Score))+stat_qq()+ggtitle("QQ PLot - Test for Normali
plot2
```

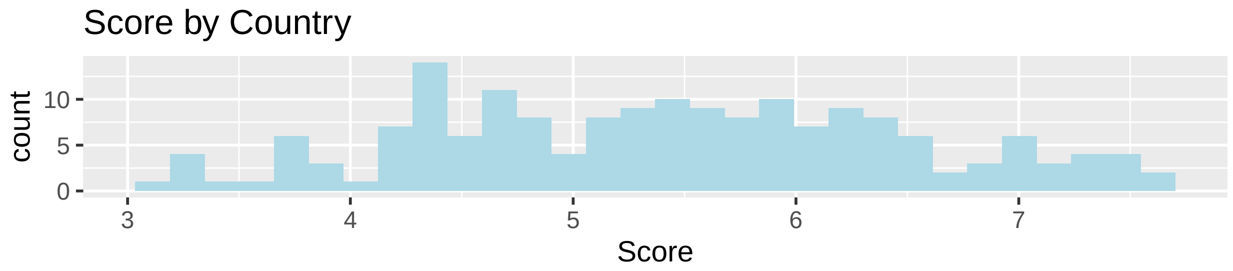


```
#Obtain the mean Happiness score by Country only
MeanHappy<-Happiness%>% group_by(Country)%>% summarise(mean(Happiness.Score))
#Rename columns
colnames(MeanHappy)<-c("Country","Score")
plot3<-MeanHappy %>%ggplot(aes(Score)) +   geom_histogram(fill = "light blue")+  ggtitle("Score by Coun
grid.arrange(plot1,plot2,plot3,nrow=3)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```







```
#Top most and bottom most  scores
MeanHappy%>%top_n(5, Score)
```

```
## # A tibble: 5 x 2
##   Country     Score
##   <chr>       <dbl>
## 1 Denmark      7.58
## 2 Finland      7.65
## 3 Iceland      7.52
## 4 Norway       7.49
## 5 Switzerland  7.52
```

```
MeanHappy%>%top_n(-5, Score)
```

```
## # A tibble: 5 x 2
##   Country                  Score
##   <chr>                    <dbl>
## 1 Afghanistan               3.13
## 2 Burundi                   3.28
## 3 Central African Republic  3.20
## 4 South Sudan               3.27
## 5 Syria                     3.29
```

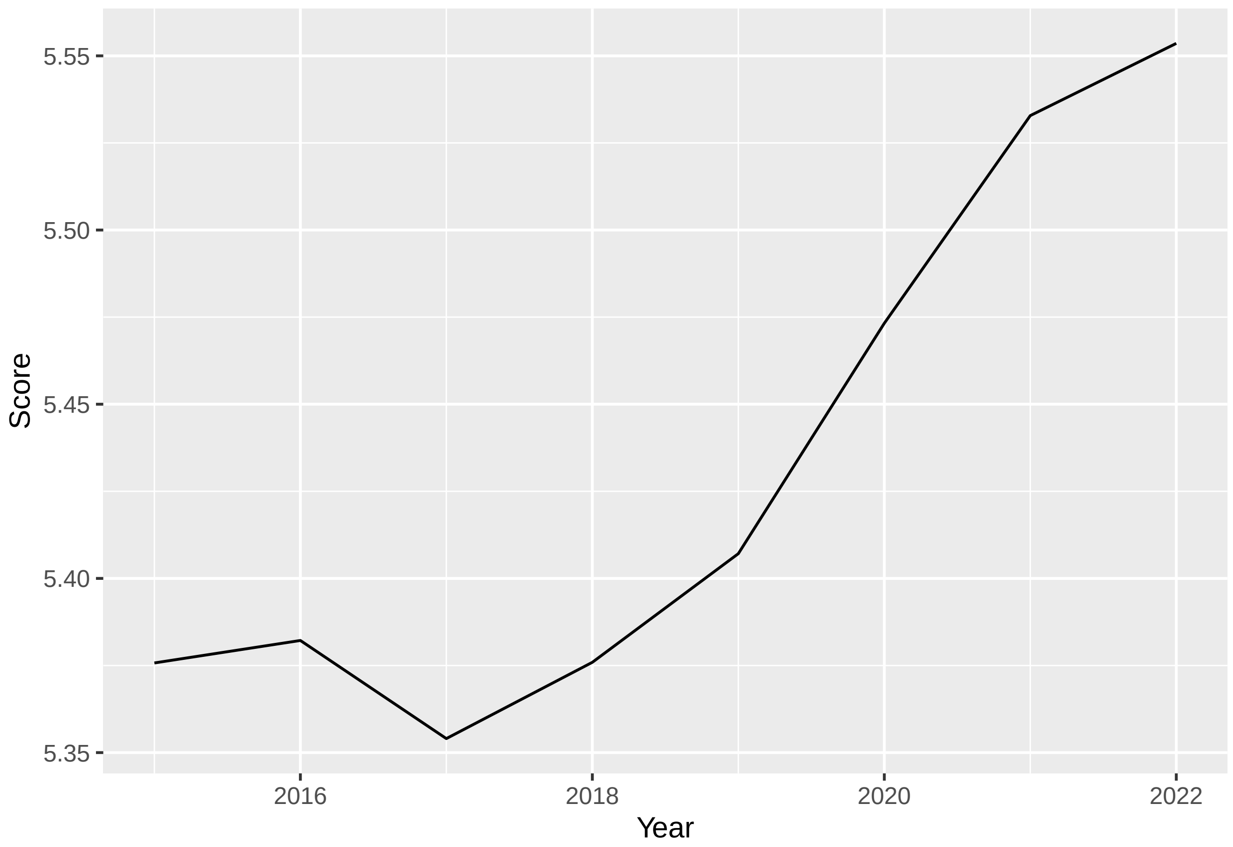
```
#Obtain mean and sd of happiness score
summary <- Happiness %>%summarize(mean = mean(Happiness$Happiness.Score), sd = sd(Happiness$Happiness.Sc
summary
```

```
##       mean       sd
## 1 5.430092 1.115361
```

```
#Now we look at the overall happiness levels across years.  They are increasing.
MeanHappyYear<-Happiness%>% group_by(Year)%>% summarise(mean(Happiness.Score))
#Rename columns
colnames(MeanHappyYear)<-c("Year","Score")
MeanHappyYear
```

```
## # A tibble: 8 x 2
##    Year Score
##   <int> <dbl>
## 1  2015  5.38
## 2  2016  5.38
## 3  2017  5.35
## 4  2018  5.38
## 5  2019  5.41
## 6  2020  5.47
## 7  2021  5.53
## 8  2022  5.55
```

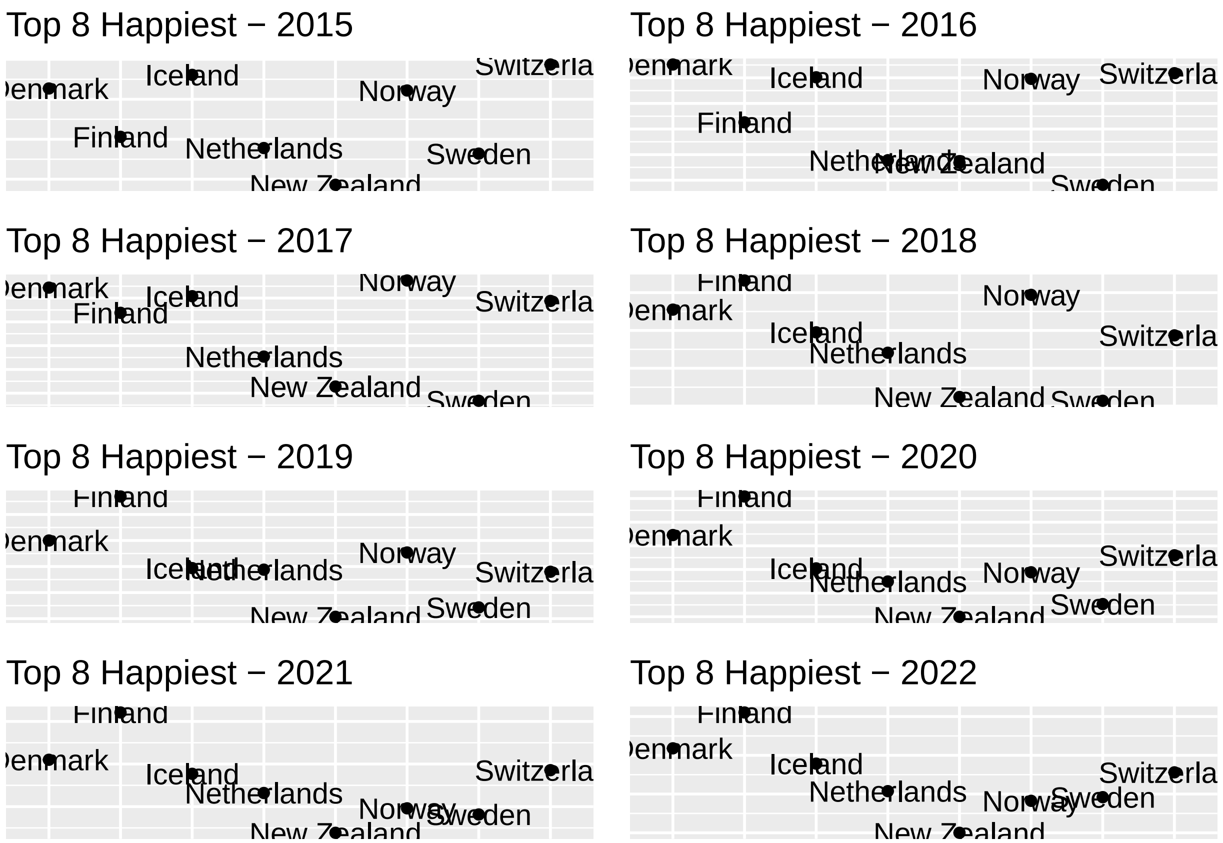
```
plot1<-MeanHappyYear%>%ggplot(aes(Year, Score)) + geom_line()
plot1
```



```
#We want to see which countries show up in the top 10 for each year.
top<-MeanHappy%>%arrange(desc(Score))%>%slice(1:10)
top_2015<-Happiness%>%filter(Year==2015)%>%arrange(desc(Happiness.Score))%>%slice(1:10)
top_2016<-Happiness%>%filter(Year==2016)%>%arrange(desc(Happiness.Score))%>%slice(1:10)
top_2017<-Happiness%>%filter(Year==2017)%>%arrange(desc(Happiness.Score))%>%slice(1:10)
top_2018<-Happiness%>%filter(Year==2018)%>%arrange(desc(Happiness.Score))%>%slice(1:10)
top_2019<-Happiness%>%filter(Year==2019)%>%arrange(desc(Happiness.Score))%>%slice(1:10)
top_2020<-Happiness%>%filter(Year==2020)%>%arrange(desc(Happiness.Score))%>%slice(1:10)
top_2021<-Happiness%>%filter(Year==2021)%>%arrange(desc(Happiness.Score))%>%slice(1:10)
top_2022<-Happiness%>%filter(Year==2022)%>%arrange(desc(Happiness.Score))%>%slice(1:10)
top_allyears<-rbind(top_2015,top_2016,top_2017,top_2018,top_2019,top_2020,top_2021,top_2022)
#Countries which show up in the top 10 happiest countries each year, will show up 8 times in this table
table(top_allyears$Country)
```

```
##
##   Australia     Austria      Canada     Denmark     Finland     Iceland
##           4           3           5           8           8           8
##      Israel  Luxembourg Netherlands New Zealand      Norway      Sweden
##           1           3           8           8           8           8
## Switzerland
##           8
```

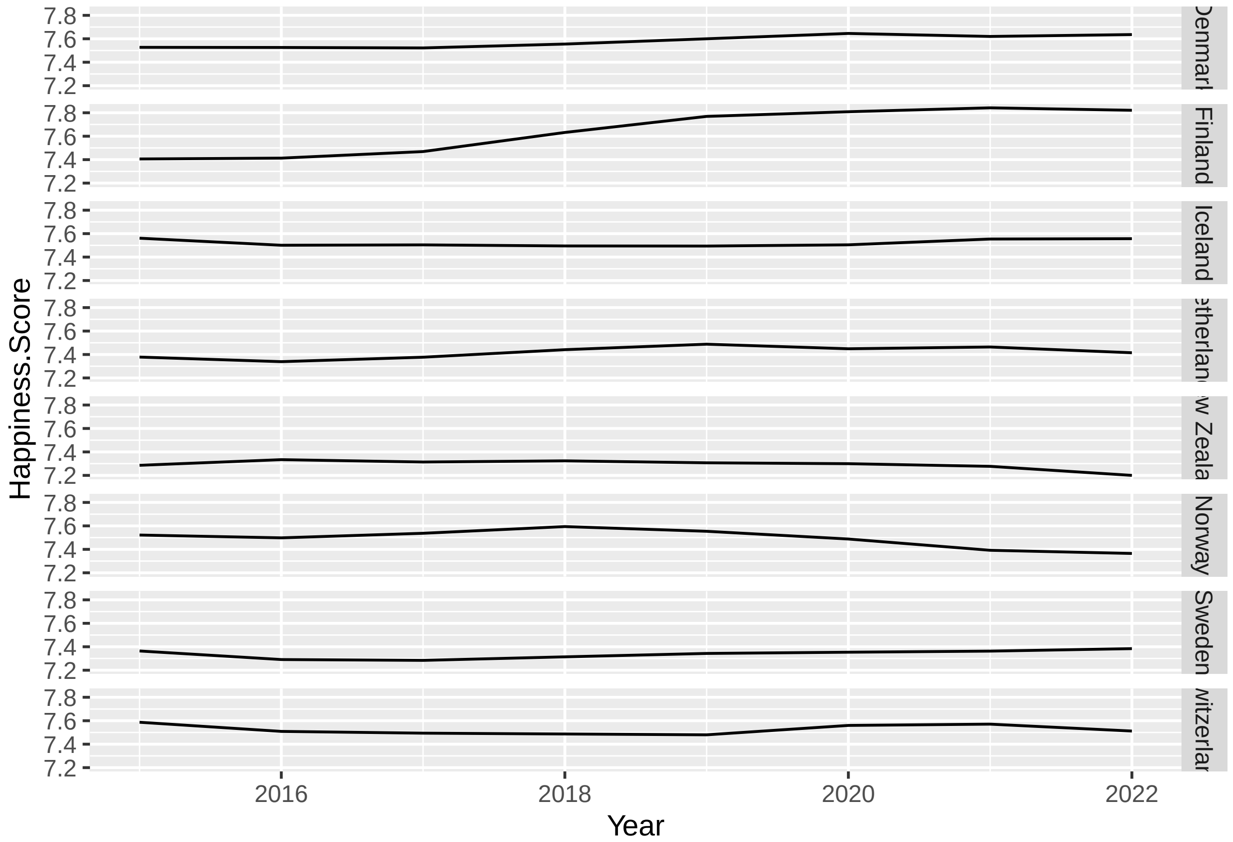
```
#We will select countries which were in top 8 across all years
top_8_all_years<-top_allyears%>%filter(Country %in% c("Denmark","Finland","Iceland","Netherlands","New 
plot1<-top_8_all_years %>%filter(Year==2015)%>%ggplot(aes(x=Country,y=Happiness.Score))+geom_point() + 
plot2<-top_8_all_years %>%filter(Year==2016)%>%ggplot(aes(x=Country,y=Happiness.Score))+geom_point() + g
plot3<-top_8_all_years %>%filter(Year==2017)%>%ggplot(aes(x=Country,y=Happiness.Score))+geom_point() + g
plot4<-top_8_all_years %>%filter(Year==2018)%>%ggplot(aes(x=Country,y=Happiness.Score))+geom_point() + g
plot5<-top_8_all_years %>%filter(Year==2019)%>%ggplot(aes(x=Country,y=Happiness.Score))+geom_point() + g
plot6<-top_8_all_years %>%filter(Year==2020)%>%ggplot(aes(x=Country,y=Happiness.Score))+geom_point() + g
plot7<-top_8_all_years %>%filter(Year==2021)%>%ggplot(aes(x=Country,y=Happiness.Score))+geom_point() + g
plot8<-top_8_all_years %>%filter(Year==2022)%>%ggplot(aes(x=Country,y=Happiness.Score))+geom_point() + g
# we can see the same countries in top 8 happiest in each year
grid.arrange(plot1,plot2,plot3,plot4,plot5,plot6,plot7,plot8,ncol=2)
```



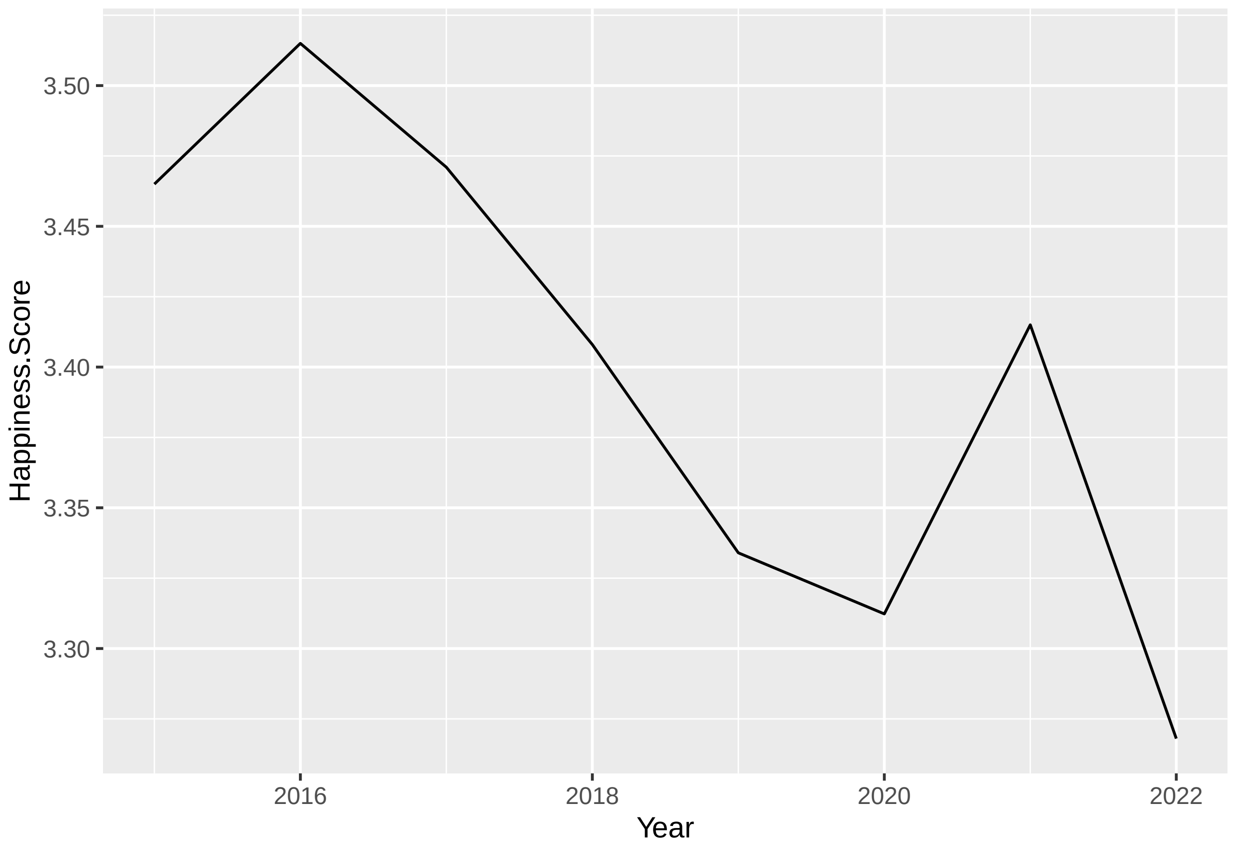
```
#Now we look at the unhappiest countries and see Rwanda is the only country that shows up in all years.
bottom<-MeanHappy%>%arrange(Score)%>%slice(1:10)
bottom_2015<-Happiness%>%filter(Year==2015)%>%arrange(Happiness.Score)%>%slice(1:10)
bottom_2016<-Happiness%>%filter(Year==2016)%>%arrange(Happiness.Score)%>%slice(1:10)
bottom_2017<-Happiness%>%filter(Year==2017)%>%arrange(Happiness.Score)%>%slice(1:10)
bottom_2018<-Happiness%>%filter(Year==2018)%>%arrange(Happiness.Score)%>%slice(1:10)
bottom_2019<-Happiness%>%filter(Year==2019)%>%arrange(Happiness.Score)%>%slice(1:10)
bottom_2020<-Happiness%>%filter(Year==2020)%>%arrange(Happiness.Score)%>%slice(1:10)
bottom_2021<-Happiness%>%filter(Year==2021)%>%arrange(Happiness.Score)%>%slice(1:10)
bottom_2022<-Happiness%>%filter(Year==2022)%>%arrange(Happiness.Score)%>%slice(1:10)
bottom_allyears<-rbind(bottom_2015,bottom_2016,bottom_2017,bottom_2018,bottom_2019,bottom_2020,bottom_20
table(bottom_allyears$Country)
```

```
##
##                 Afghanistan                        Benin                    Botswana
##                        6                        2                        4 
##             Burkina Faso                  Burundi Central African Republic 
##                        1                        5                        4 
##                     Chad                   Guinea                    Haiti 
##                        1                        3                        3 
##                    India              Ivory Coast                  Lebanon 
##                        1                        1                        1 
##                  Lesotho                  Liberia               Madagascar 
##                        2                        3                        1 
##                   Malawi                   Rwanda             Sierra Leone 
##                        5                        8                        1 
##              South Sudan                    Syria                 Tanzania 
##                        4                        5                        7 
##                     Togo                    Yemen                   Zambia 
##                        3                        5                        1 
##                 Zimbabwe 
##                        3
```

```
#Now we will look at scores by top countries by year, and Rwanda score by year.
p <- ggplot(data=top_8_all_years, aes(x = Year , y = Happiness.Score)) +
  geom_line() +facet_grid(Country~.)
p
```



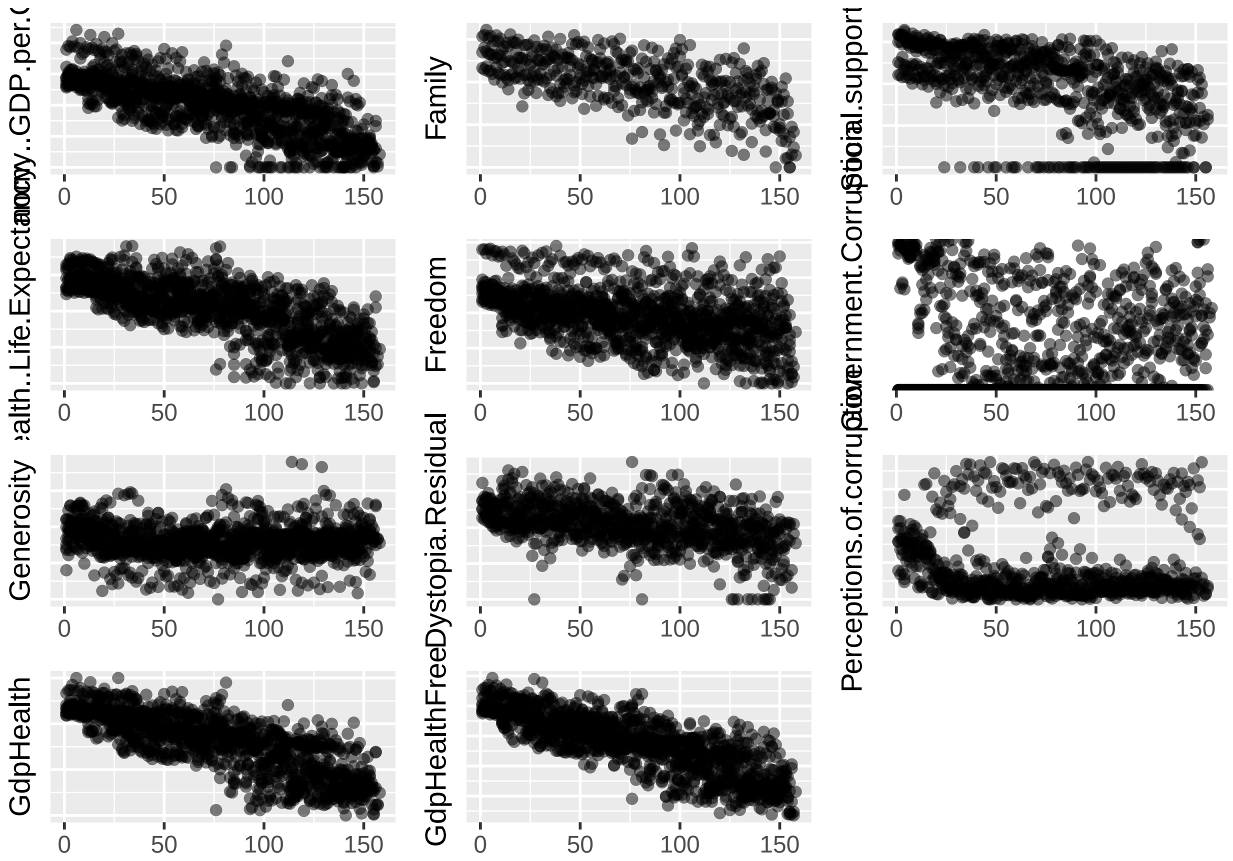
```
#Rwanda by year
p <-Happiness%>%filter(Country=='Rwanda')%>%ggplot(aes(x = Year , y = Happiness.Score)) + geom_line()
p
```



```r
#########################################################################
#CORRELATION BETWEEN FACTORS AND HAPPINESS SCORES ON ALL COUNTRIES
#We seek any relationship between happiness and various other dimensions
#########################################################################

#relationship between happiness and GDP
plot1<-Happiness%>%ggplot(aes(Happiness.Rank, Economy..GDP.per.Capita.)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

#relationship between happiness and family
plot2<-Happiness%>%ggplot(aes(Happiness.Rank, Family)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

#relationship between happiness and social support
plot3<-Happiness%>%ggplot(aes(Happiness.Rank, Social.support)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

# relationship between happiness and life expectancy
plot4<-Happiness%>%ggplot(aes(Happiness.Rank, Health..Life.Expectancy.)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

#relationship between happiness and freedom
plot5<-Happiness%>%ggplot(aes(Happiness.Rank, Freedom)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

# relationship between happiness and Governement corruption
```

```r
plot6<-Happiness%>%ggplot(aes(Happiness.Rank, Trust..Government.Corruption.)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

#relationship between happiness and generosity
plot7<-Happiness%>%ggplot(aes(Happiness.Rank, Generosity)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

#relationship between happiness and Dystopia
plot8<-Happiness%>%ggplot(aes(Happiness.Rank, Dystopia.Residual)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

#relationship between happiness and perception of corruption
plot9<-Happiness%>%ggplot(aes(Happiness.Rank, Perceptions.of.corruption)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

#happiness & GdpHealth
plot10<-Happiness%>%ggplot(aes(Happiness.Rank, GdpHealth)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

#happiness & GdpHealthFree
plot11<-Happiness%>%ggplot(aes(Happiness.Rank, GdpHealthFree)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=NULL)

#highest correlations
grid.arrange(plot1,plot2,plot3,plot4,plot5,plot6,plot7,plot8,plot9,plot10, plot11,ncol=3)
```

```
## Warning: Removed 760 rows containing missing values (geom_point).

## Warning: Removed 470 rows containing missing values (geom_point).

## Warning: Removed 312 rows containing missing values (geom_point).

## Warning: Removed 471 rows containing missing values (geom_point).
```



```
#We calculate correlation on top 5 observable correlations
corEcon<-round(cor(Happiness$Happiness.Score,Happiness$Economy..GDP.per.Capita.),digits=2)
corLifeEx<-round(cor(Happiness$Happiness.Score,Happiness$Health..Life.Expectancy.),digits=2)
corFree<-round(cor(Happiness$Happiness.Score,Happiness$Freedom),digits=2)
corGdpHealth<-round(cor(Happiness$Happiness.Score,Happiness$GdpHealth),digits=2)
corGdpHealthFree<-round(cor(Happiness$Happiness.Score,Happiness$GdpHealthFree),digits=2)
corEcon
```

```
## [1] 0.74
```

```
corLifeEx
```

```
## [1] 0.73
```

```
corFree
```

```
## [1] 0.46
```

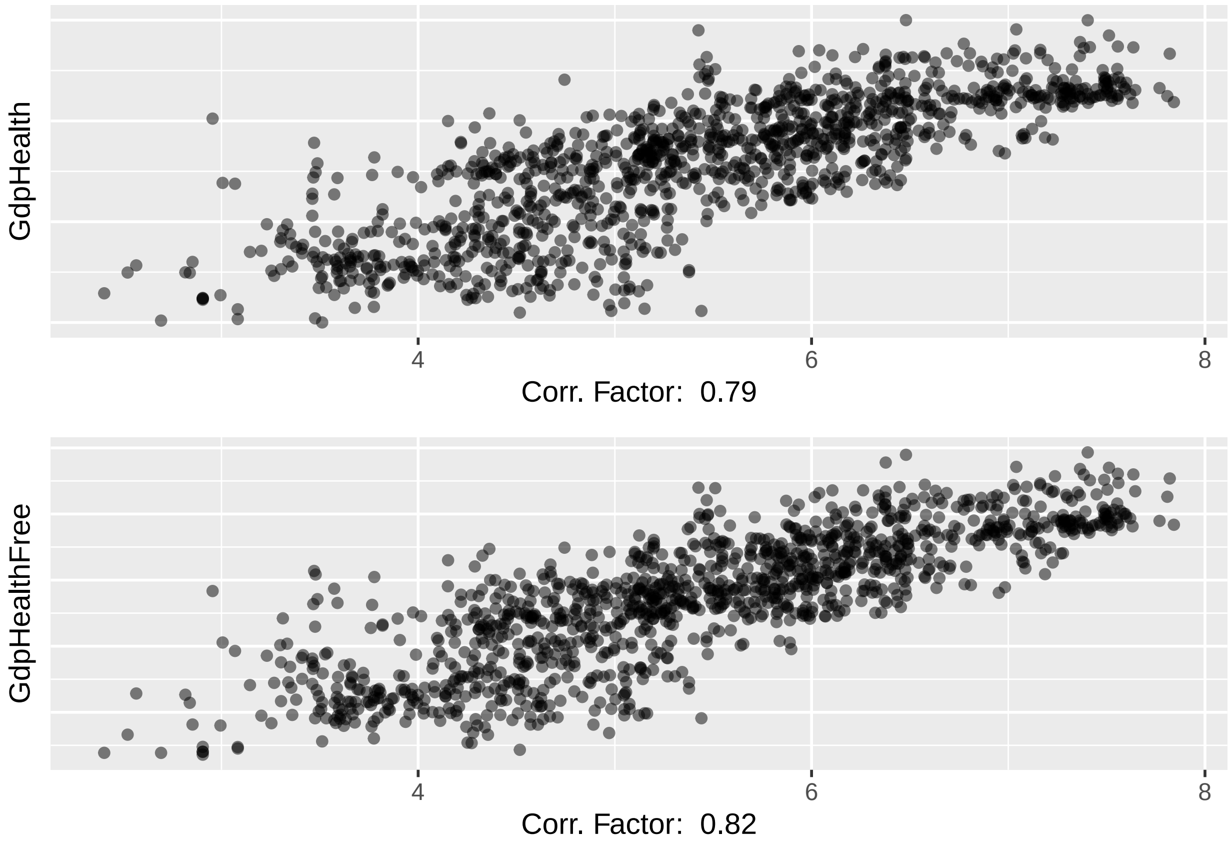
```
corGdpHealth
```

```
## [1] 0.79
```

```
corGdpHealthFree
```

```
## [1] 0.82
```

```
corGdpHealth2<-paste(c("Corr. Factor: ",corGdpHealth),collapse=" ")
corGdpHealthFree2<-paste(c("Corr. Factor: ",corGdpHealthFree),collapse=" ")

#Plot if top 2 correlation values
#relationship between happiness and GdpHealth
plot1<-Happiness%>%ggplot(aes(Happiness.Score, GdpHealth)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=corGdp
#Health and GdpHealthFree
plot2<-Happiness%>%ggplot(aes(Happiness.Score, GdpHealthFree)) +
  geom_point(alpha = 0.5)+theme(axis.text.y=element_blank(),axis.ticks.y=element_blank())+labs(x=corGdp

grid.arrange(plot1,plot2,nrow=2)
```



```
# For happiest countries, correlations are small.  The largest is a negative correlation with Generosit
corTopEcon<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$Economy..GDP.per.Capita.),digits=
corTopLifeEx<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$Health..Life.Expectancy.),digit
corTopFreedom<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$Freedom),digits=2)
corTopGenerosity<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$Generosity),digits=2)
corTopGdpHealth<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$GdpHealth),digits=2)
corTopGdpHealthFree<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$GdpHealthFree),digits=2)
corTopEcon
```

```
## [1] 0.07
```

```
corTopLifeEx
```

```
## [1] 0.04
```

```
corTopFreedom
```

```
## [1] 0.19
```

```
corTopGenerosity
```

```
## [1] -0.5
```

```
corTopGdpHealth
```

```
## [1] 0.11
```

```
corTopGdpHealthFree
```

```
## [1] 0.19
```

```
# For unhappiest countries, correlations are small.  The largest is a negative correlation with Generos
corBottomEcon<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$Economy..GDP.per.Capita.),digi
corBottomLifeEx<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$Health..Life.Expectancy.),dig
corBottomFreedom<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$Freedom),digits=2)
corBottomGenerosity<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$Generosity),digits=2)
corBottomGdpHealth<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$GdpHealth),digits=2)
corBottomGdpHealthFree<-round(cor(top_8_all_years$Happiness.Score,top_8_all_years$GdpHealthFree),digits=
corBottomEcon
```

```
## [1] 0.07
```

```
corBottomLifeEx
```

```
## [1] 0.04
```

```
corBottomFreedom
```

```
## [1] 0.19
```

```
corBottomGenerosity
```

```
## [1] -0.5
```

```
corBottomGdpHealth
```

```
## [1] 0.11
```

```
corBottomGdpHealthFree
```

```
## [1] 0.19
```

```
##########################################################################
#PREDICTING HAPPINESS SCORE
#We seek various ways of predicting happiness scores
##########################################################################


#We create train set to contain years 2015 - 2021, and test_set to contain year 2022
train_set <- Happiness %>% filter(Year!='2022')%>%select(Year,Country,CtryYear, Happiness.Score,Economy
max(train_set$Year)
```

```
## [1] 2021
```

```
test_set <- Happiness %>% filter(Year=='2022')%>%select(Year,Country,CtryYear,Happiness.Score,Economy..
min(test_set$Year)
```

```
## [1] 2022
```

```
# First model, we use the mean of train_set on test_set to get a squared loss of 1.19
HappyTrain<-mean(train_set$Happiness.Score)
HappyTrain
```

```
## [1] 5.41346
```

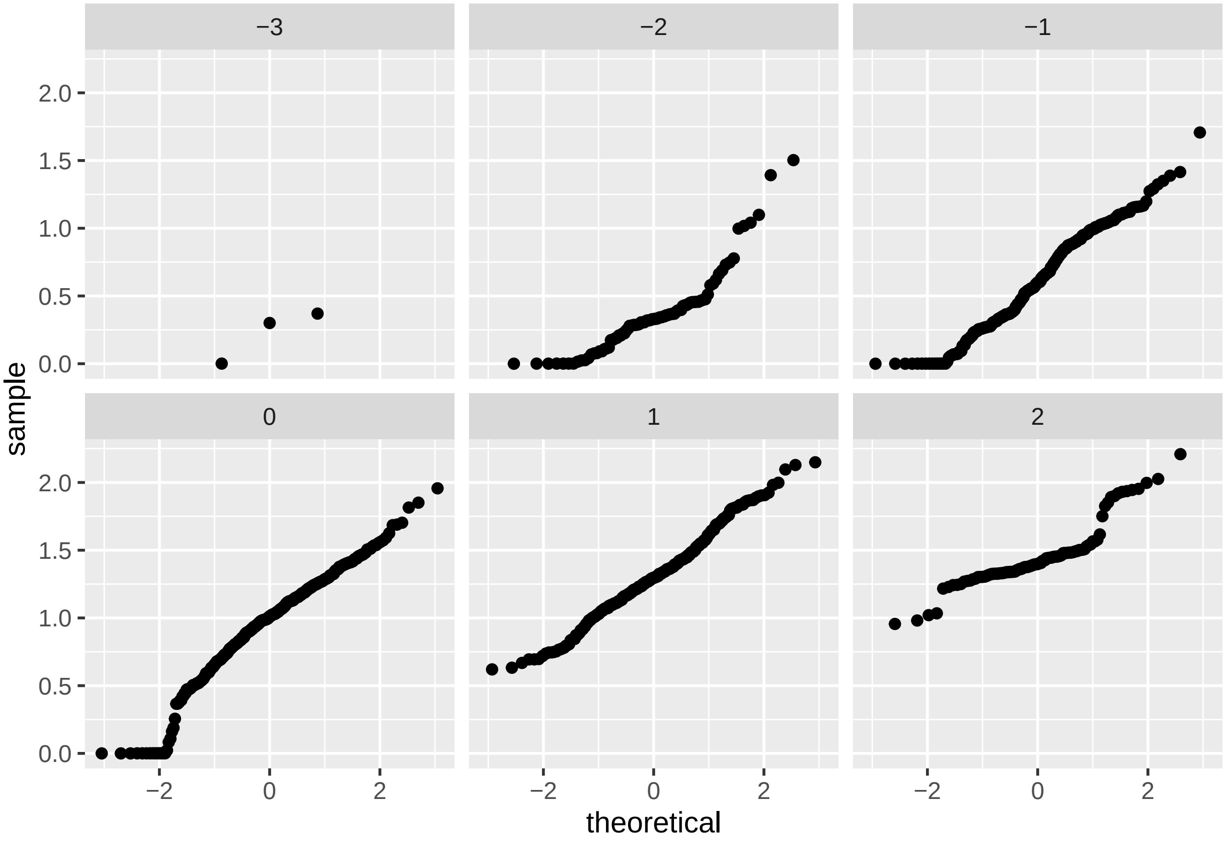
```
Sq_Loss<-mean((HappyTrain - test_set$Happiness.Score)^2)
Sq_Loss
```

```
## [1] 1.192769
```

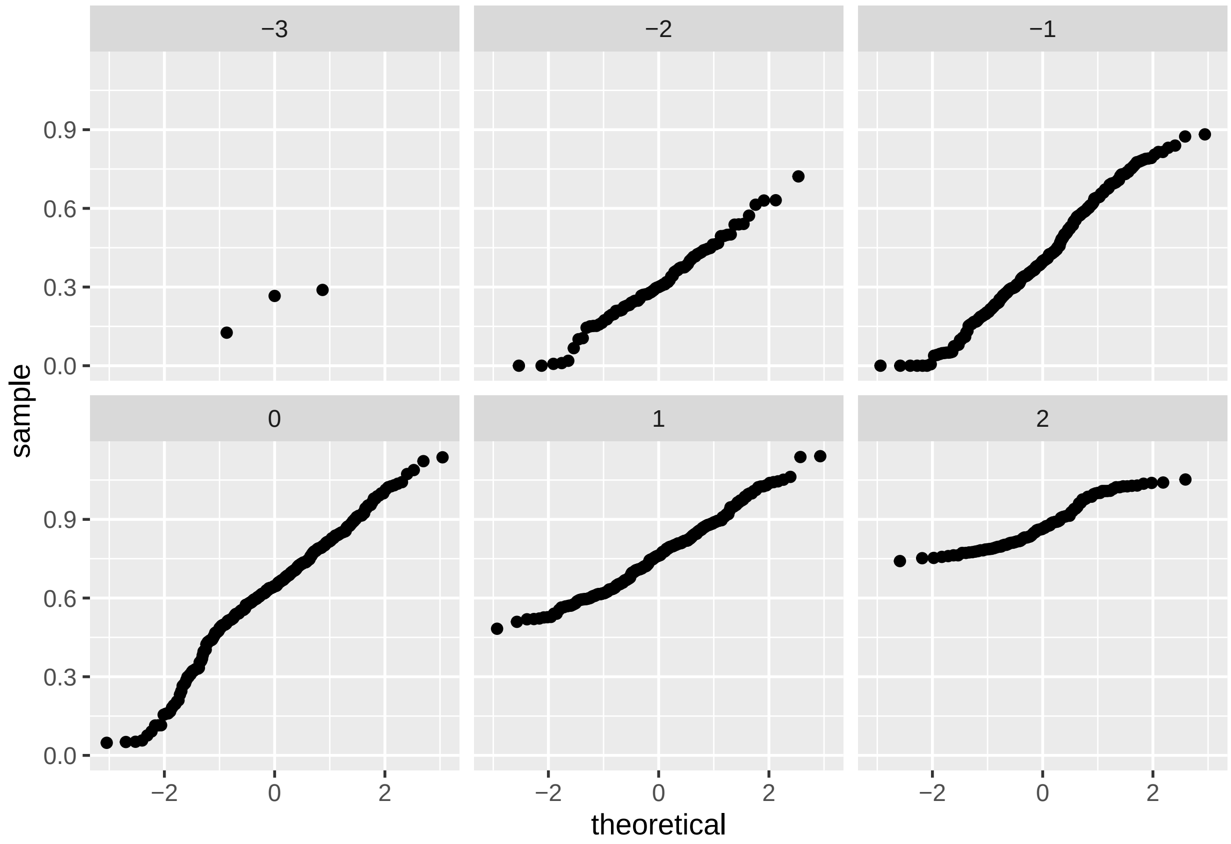
```
#create table to store results
Results<-data_frame(Method="Mean Happiness Score",SqLoss=Sq_Loss)
```

```
## Warning: ‘data_frame()‘ was deprecated in tibble 1.1.0.
## Please use ‘tibble()‘ instead.
```

```
#Data is approximately bivariate normal.
Happiness %>% mutate(z_happy = round((Happiness.Score - mean(Happiness.Score)) / sd(Happiness.Score)))
```



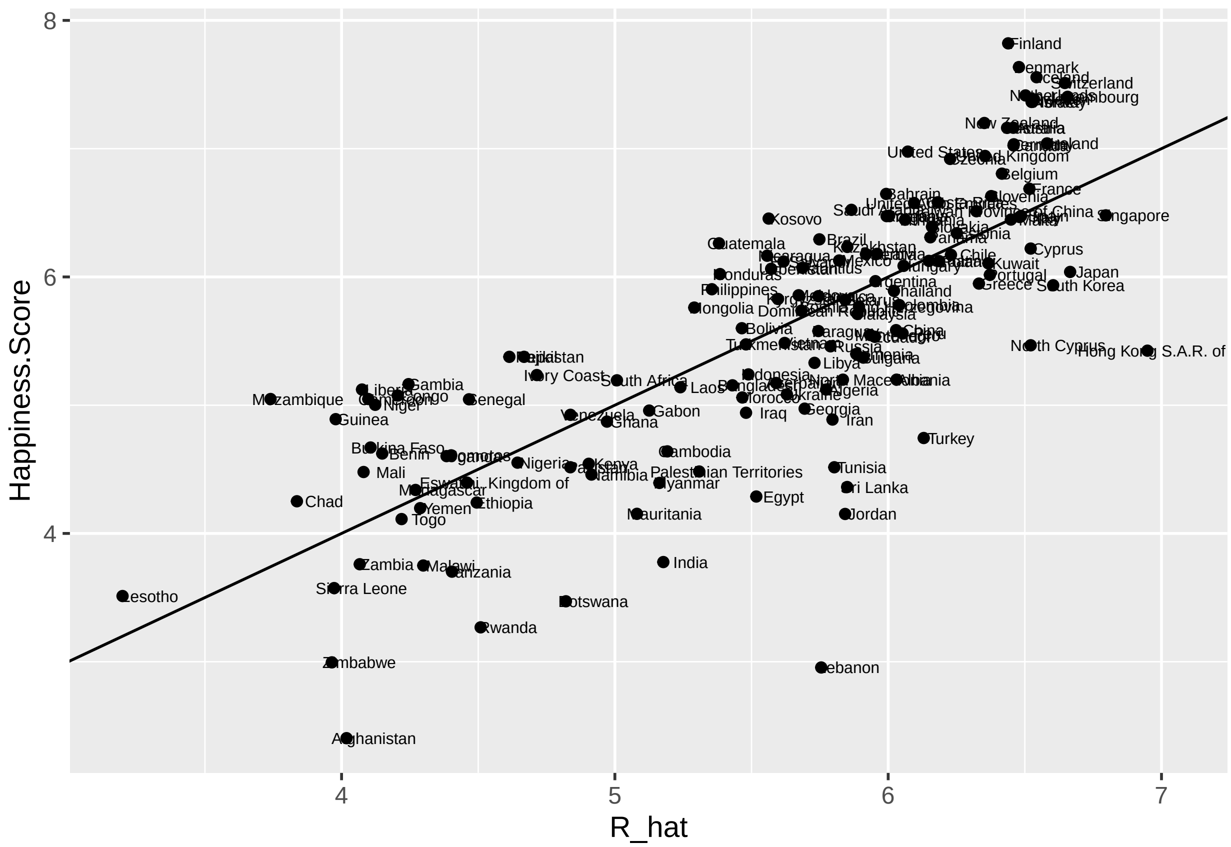
```
Happiness %>% mutate(z_happy = round((Happiness.Score - mean(Happiness.Score)) / sd(Happiness.Score))) %
```



```
#MODEL 2 uses the regression line based on GDP and Health to predict happiness score on the test set -
line_fit<-lm(Happiness.Score~Economy..GDP.per.Capita.+ Health..Life.Expectancy.,data=test_set)
line_fit
```

```
##
## Call:
## lm(formula = Happiness.Score ~ Economy..GDP.per.Capita. + Health..Life.Expectancy.,
##     data = test_set)
##
## Coefficients:
##              (Intercept)  Economy..GDP.per.Capita.  Health..Life.Expectancy.
##                   3.1981                    0.5513                    2.8362
```

```
prediction <- line_fit$coef[1] + line_fit$coef[2]*test_set$Happiness.Score
Sq_Loss<-mean((prediction - test_set$Happiness.Score)^2)
Results <- Results%>% add_row(Method="Correlation Line",SqLoss=Sq_Loss)

#Plotting the regression line against test set, year 2022.
Happiness %>%
  filter(Year %in% 2022) %>%
  mutate(R_hat = predict(line_fit, newdata = .)) %>%
  ggplot(aes(R_hat, Happiness.Score, label = Country)) +
  geom_point() +
  geom_text(nudge_x=0.1, cex = 2) +
  geom_abline()
```



```
#MODEL 3 - Using the Mean Squared errors method Naive model

#define the mean squared error function
RMSE <- function(true_score, predicted_score){
  sqrt(mean((true_score - predicted_score)^2))
}

#Mean score of all countries, 2015-2021
mu_hat <- mean(train_set$Happiness.Score)
mu_hat
```

```
## [1] 5.41346
```

```
naive_rmse <- RMSE(test_set$Happiness.Score, mu_hat)
naive_rmse
```

```
## [1] 1.092139
```

```
Sq_Loss<-mean((mu_hat - test_set$Happiness.Score)^2)
Sq_Loss
```

```
## [1] 1.192769
```

```
Results$RMSE<-NA
Results <- Results%>% add_row(Method="Naive RMSE - RMSE using Mean",SqLoss=Sq_Loss,RMSE=naive_rmse)
Results
```

```
## # A tibble: 3 x 3
##   Method                       SqLoss  RMSE
##   <chr>                         <dbl> <dbl>
## 1 Mean Happiness Score          1.19  NA
## 2 Correlation Line              0.735 NA
## 3 Naive RMSE - RMSE using Mean  1.19     1.09
```

```
#MODEL 4 - Using the Mean Squared errors adding Country bias
mu <- mean(train_set$Happiness.Score)
mu
```

```
## [1] 5.41346
```

```
country_avgs <- train_set %>%
  group_by(Country) %>%
  summarize(b_ctry = mean(Happiness.Score - mu))

predicted_ratings <- test_set %>%
 left_join(country_avgs, by='Country') %>%
  mutate( pred = mu + b_ctry)

nas<-subset(predicted_ratings,is.na(b_ctry))
nas
```

```
##     Year              Country                  CtryYear Happiness.Score
## 18  2022              Czechia              2022_Czechia           6.920
## 99  2022                Congo                2022_Congo           5.075
## 125 2022 Eswatini. Kingdom of 2022_Eswatini. Kingdom of           4.396
##     Economy..GDP.per.Capita. Health..Life.Expectancy. b_ctry pred
## 18                   1.81500                    0.715     NA   NA
## 99                   0.00095                    0.355     NA   NA
## 125                  1.27400                    0.197     NA   NA
```

```
#remove nas from test and train sets
test_set <- subset(test_set, CtryYear!='2022_Czechia')
test_set <-subset(test_set, CtryYear!='2022_Congo')
test_set <-subset(test_set, CtryYear!='2022_Eswatini. Kingdom of')

train_set <- subset(train_set, CtryYear!='2022_Czechia')
train_set <-subset(train_set, CtryYear!='2022_Congo')
train_set <-subset(train_set, CtryYear!='2022_Eswatini. Kingdom of')

# recalculate with only countries which are in both
mu <- mean(train_set$Happiness.Score)
mu
```

```
## [1] 5.41346
```

```
country_avgs <- train_set %>%
  group_by(Country) %>%
  summarize(b_ctry = mean(Happiness.Score - mu))



predicted_ratings <- test_set %>%
  left_join(country_avgs, by='Country') %>%
  mutate(pred = mu + b_ctry) %>%
  pull(pred)




model_1_rmse <- RMSE(test_set$Happiness.Score,predicted_ratings)
model_1_rmse
```

```
## [1] 0.4204794
```

```
Sq_Loss<-mean((mu - test_set$Happiness.Score)^2)
Sq_Loss
```

```
## [1] 1.193879
```

```
Results <- Results%>% add_row(Method="RMSE - with Country Bias",SqLoss=Sq_Loss,RMSE=model_1_rmse)

#It looks as if using the correlation line minimizes the squared loss and RMSE - with Country Bias
#minimizes the root mean squared error.
Results
```

```
## # A tibble: 4 x 3
##   Method                       SqLoss   RMSE
##   <chr>                         <dbl>  <dbl>
## 1 Mean Happiness Score          1.19  NA
## 2 Correlation Line              0.735 NA
## 3 Naive RMSE - RMSE using Mean  1.19   1.09
## 4 RMSE - with Country Bias      1.19   0.420
```