Happiness 2017

Original Author: Javad Zabihi

Fernando Huilca

31/01/2025

Contents

Introduction

The dataset that we have chosen is happiness 2017 dataset, one of Kaggle's dataset. This dataset gives the happiness rank and happiness score of 155 countries around the world based on seven factors including family, life expectancy, economy, generosity, trust in government, freedom, and dystopia residual. Sum of the value of these seven factors gives us the happiness score and the higher the happiness score, the lower the happiness rank. So, it is evident that the higher value of each of these seven factors means the level of happiness is higher. We can define the meaning of these factors as the extent to which these factors lead to happiness. Dystopia is the opposite of utopia and has the lowest happiness level. Dystopia will be considered as a reference for other countries to show how far they are from being the poorest country regarding happiness level.

There are three parts to my report as follows:

- Cleaning
- Visualization
- Prediction

Mission

The purpose of choosing this work is to find out which factors are more important to live a happier life. As a result, people and countries can focus on the more significant factors to achieve a higher happiness level. We also will implement several machine learning algorithms to predict the happiness score and compare the result to discover which algorithm works better for this specific dataset.

Cleaning

Now we can load our dataset and see the structure of happiness variables. Our dataset is pretty clean, and we will implement a few adjustments to make it looks better.

```
library(plyr)
library(dplyr)
library(tidyverse)
library(lubridate)
library(caTools)
library(ggplot2)
library(ggthemes)
library(reshape2)
library(data.table)
library(tidyr)
library(corrgram)
library(corrplot)
library(formattable)
library(cowplot)
library(ggpubr)
library(plot3D)
```

```
# World happiness report 2017
Happiness <- read.csv("C:\\Users\\Fernando_Huilca\\Desktop\\EPN-FernandoHuilca\\Cuarto Semestre\\Sistema

str(Happiness)
```

```
## 'data.frame':    155 obs. of  12 variables:
##  $ Country                      : chr  "Norway" "Denmark" "Iceland" "Switzerland" ...
##  $ Happiness.Rank               : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Happiness.Score              : num  7.54 7.52 7.5 7.49 7.47 ...
##  $ Whisker.high                 : num  7.59 7.58 7.62 7.56 7.53 ...
##  $ Whisker.low                  : num  7.48 7.46 7.39 7.43 7.41 ...
##  $ Economy..GDP.per.Capita.     : num  1.62 1.48 1.48 1.56 1.44 ...
##  $ Family                       : num  1.53 1.55 1.61 1.52 1.54 ...
##  $ Health..Life.Expectancy.     : num  0.797 0.793 0.834 0.858 0.809 ...
##  $ Freedom                      : num  0.635 0.626 0.627 0.62 0.618 ...
##  $ Generosity                   : num  0.362 0.355 0.476 0.291 0.245 ...
##  $ Trust..Government.Corruption.: num  0.316 0.401 0.154 0.367 0.383 ...
##  $ Dystopia.Residual            : num  2.28 2.31 2.32 2.28 2.43 ...
```

We have observed the variables inside our dataset, their class, and the first few observations of each. In fact, the dataset has 155 observations and 12 variables. I believe some of the variable names are not clear enough and I decided to change the name of several of them a little bit. Also, I will remove whisker low and whisker

high variables from my dataset because these variables give only the lower and upper confidence interval of happiness score and there is no need to use them for visualization and prediction.

```
# Changing the name of columns
colnames (Happiness) <- c("Country", "Happiness.Rank", "Happiness.Score",
                          "Whisker.High", "Whisker.Low", "Economy", "Family",
                          "Life.Expectancy", "Freedom", "Generosity",
                          "Trust", "Dystopia.Residual")


# Country: Name of countries
# Happiness.Rank: Rank of the country based on the Happiness Score
# Happiness.Score: Happiness measurement on a scale of 0 to 10
# Whisker.High: Upper confidence interval of happiness score
# Whisker.Low: Lower confidence interval of happiness score
# Economy: The value of all final goods and services produced within a nation in a given year
# Per capita GDP is a measure of the total output of a country that takes the gross domestic product (G
# Family: Importance of having a family
# Life.Expectancy: Importance of health and amount of time prople expect to live
# Freedom: Importance of freedom in each country
# Generosity: The quality of being kind and generous
# Trust: Perception of corruption in a government
# Dystopia.Residual: Plays as a reference

# Deleting unnecessary columns (Whisker.high and Whisker.low)

Happiness <- Happiness[, -c(4,5)]
```

The next step is adding another column to the dataset which is continent. I want to work on different continents to discover whether there are different trends for them regarding which factors play a significant role in gaining higher happiness score. Asia, Africa, North America, South America, Europe, and Australia are our six continents in this dataset. Then I moved the position of the continent column to the second column because I think with this position arrange, dataset looks better. Finally, I changed the type of continent variable to factor to be able to work with it easily for visualization. Now we can see the final structure of our dataset which consisted of 155 observations and 11 variables. Country and continent are factor variables, Happiness rank is an integer, and the remaining variables are in numeric type.

```
# Creating a new column for continents

Happiness$Continent <- NA

Happiness$Continent[which(Happiness$Country %in% c("Israel", "United Arab Emirates", "Singapore", "Thail
                                    "Qatar", "Saudi Arabia", "Kuwait", "Bahrain", "Malaysia", "Uzbekistar
                                    "South Korea", "Turkmenistan", "Kazakhstan", "Turkey", "Hong Kong S.A
                                    "Jordan", "China", "Pakistan", "Indonesia", "Azerbaijan", "Lebanon",
                                    "Tajikistan", "Bhutan", "Kyrgyzstan", "Nepal", "Mongolia", "Palestini
                                    "Iran", "Bangladesh", "Myanmar", "Iraq", "Sri Lanka", "Armenia", "Ind
                                    "Cambodia", "Afghanistan", "Yemen", "Syria"))] <- "Asia"
Happiness$Continent[which(Happiness$Country %in% c("Norway", "Denmark", "Iceland", "Switzerland", "Finla
                                    "Netherlands", "Sweden", "Austria", "Ireland", "Germany",
                                    "Belgium", "Luxembourg", "United Kingdom", "Czech Republic",
                                    "Malta", "France", "Spain", "Slovakia", "Poland", "Italy",
                                    "Russia", "Lithuania", "Latvia", "Moldova", "Romania",
                                    "Slovenia", "North Cyprus", "Cyprus", "Estonia", "Belarus",
```

```
                                    "Serbia", "Hungary", "Croatia", "Kosovo", "Montenegro",
                                    "Greece", "Portugal", "Bosnia and Herzegovina", "Macedonia",
                                    "Bulgaria", "Albania", "Ukraine"))] <- "Europe"
Happiness$Continent[which(Happiness$Country %in% c("Canada", "Costa Rica", "United States", "Mexico",
                                    "Panama","Trinidad and Tobago", "El Salvador", "Belize", "Guatemala"
                                    "Jamaica", "Nicaragua", "Dominican Republic", "Honduras",
                                    "Haiti"))] <- "North America"
Happiness$Continent[which(Happiness$Country %in% c("Chile", "Brazil", "Argentina", "Uruguay",
                                    "Colombia", "Ecuador", "Bolivia", "Peru",
                                    "Paraguay", "Venezuela"))] <- "South America"
Happiness$Continent[which(Happiness$Country %in% c("New Zealand", "Australia"))] <- "Australia"
Happiness$Continent[which(is.na(Happiness$Continent))] <- "Africa"


# Moving the continent column's position in the dataset to the second column

Happiness <- Happiness %>% select(Country,Continent, everything())

# Changing Continent column to factor

Happiness$Continent <- as.factor(Happiness$Continent)

str(Happiness)
```

```
## 'data.frame':    155 obs. of  11 variables:
##  $ Country          : chr  "Norway" "Denmark" "Iceland" "Switzerland" ...
##  $ Continent        : Factor w/ 6 levels "Africa","Asia",..: 4 4 4 4 4 4 5 3 4 3 ...
##  $ Happiness.Rank   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Happiness.Score  : num  7.54 7.52 7.5 7.49 7.47 ...
##  $ Economy          : num  1.62 1.48 1.48 1.56 1.44 ...
##  $ Family           : num  1.53 1.55 1.61 1.52 1.54 ...
##  $ Life.Expectancy  : num  0.797 0.793 0.834 0.858 0.809 ...
##  $ Freedom          : num  0.635 0.626 0.627 0.62 0.618 ...
##  $ Generosity       : num  0.362 0.355 0.476 0.291 0.245 ...
##  $ Trust            : num  0.316 0.401 0.154 0.367 0.383 ...
##  $ Dystopia.Residual: num  2.28 2.31 2.32 2.28 2.43 ...
```

Visualization

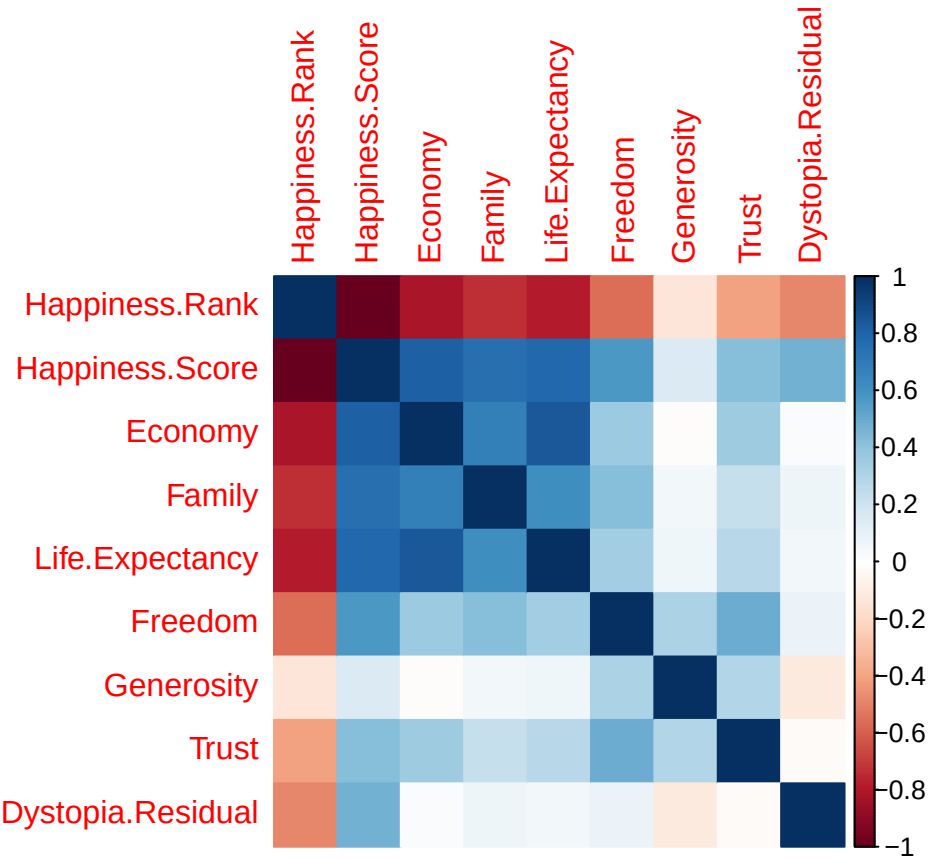
In this section, we will play with different variables to find out how they correlate with each other.

Correlation plot

Let's see the correlation between numerical variables in our dataset.

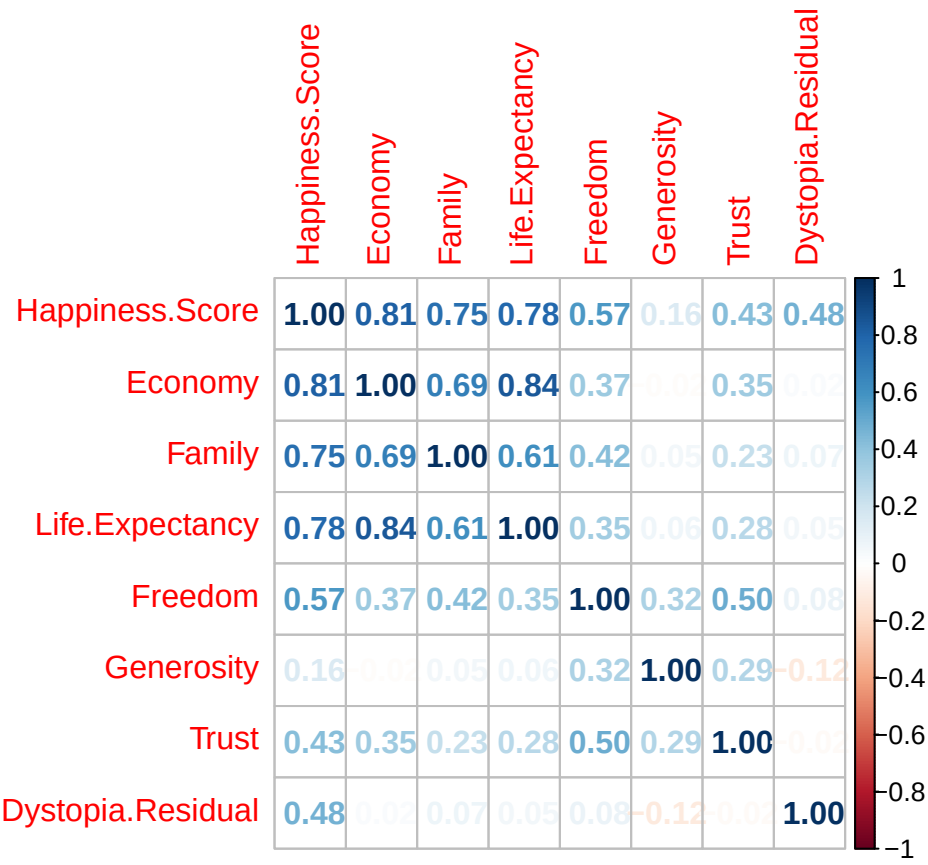
```
########## Correlation between variables

# Finding the correlation between numerical columns
Num.cols <- sapply(Happiness, is.numeric)
Cor.data <- cor(Happiness[, Num.cols])
```

```
corrplot(Cor.data, method = 'color')
```



Obviously, there is an inverse correlation between "Happiness Rank" and all the other numerical variables. In other words, the lower the happiness rank, the higher the happiness score, and the higher the other seven factors that contribute to happiness. So let's remove the happiness rank, and see the correlation again.

```
# Create a correlation plot
newdatacor = cor(Happiness[c(4:11)])
corrplot(newdatacor, method = "number")
```



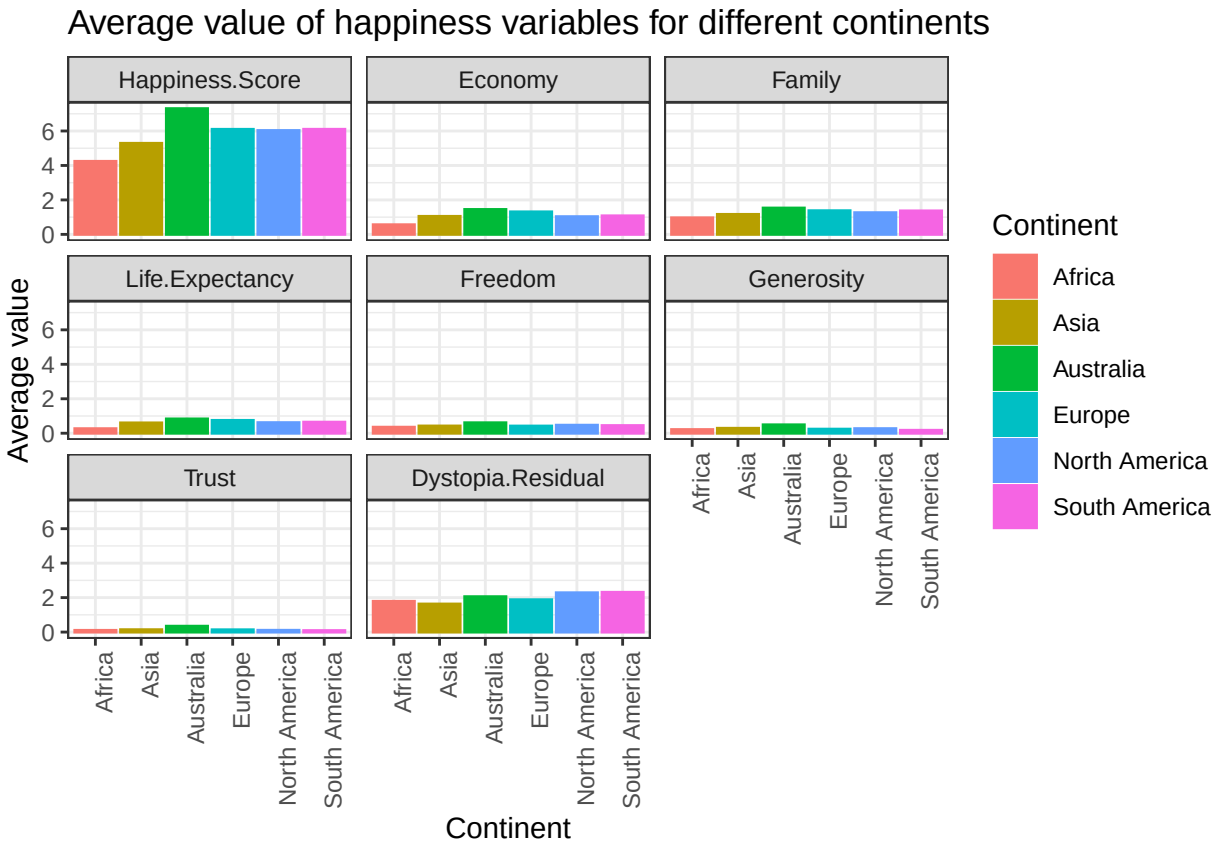
According to the above cor plot, Economy, life expectancy, and family play the most significant role in contributing to happiness. Trust and generosity have the lowest impact on the happiness score.

Comparing different continents regarding their happiness variables

Let's calculate the average happiness score and the average of the other seven variables for each continent. Then melt it to have variables and values in separate columns. Finally, using ggplot to show the difference between continents.

```
Happiness.Continent <- Happiness %>%
                          select(-3) %>%
                          group_by(Continent) %>%
                          summarise_at(vars(-Country), funs(mean(., na.rm=TRUE)))

# Or we can use aggregate
# aggregate(Happiness[, 4:11], list(Happiness$Continent), mean)

# Melting the "Happiness.Continent" dataset
Happiness.Continent.melt <- melt(Happiness.Continent)


# Faceting
ggplot(Happiness.Continent.melt, aes(y=value, x=Continent, color=Continent, fill=Continent)) +
  geom_bar( stat="identity") +
  facet_wrap(~variable) + theme_bw() +
```

```
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Average value of happiness variables for different continents",
       y = "Average value")
```

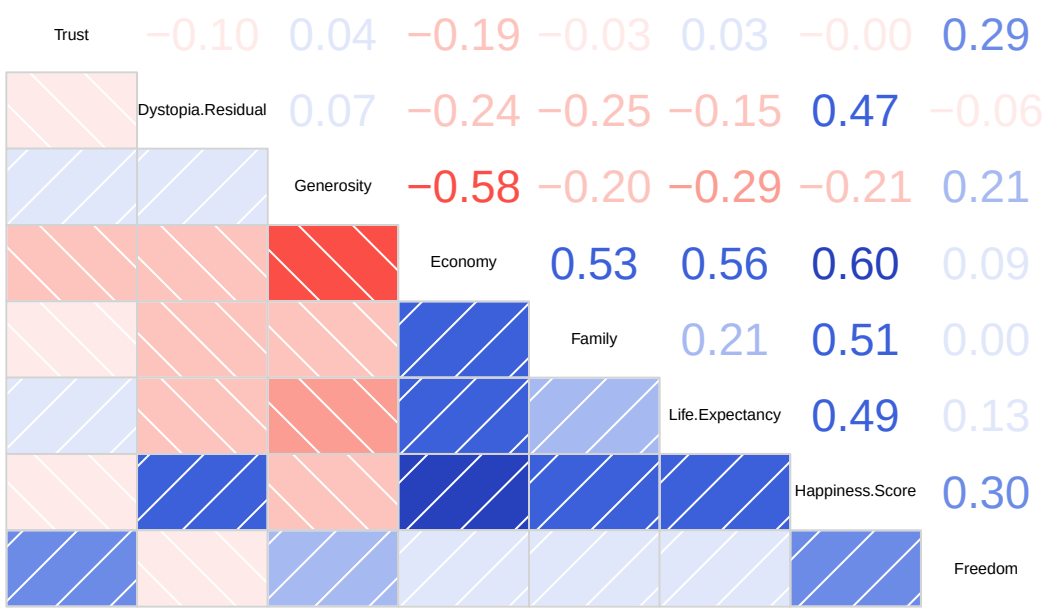


We can see that Australia has approximately the highest average in all fields except dystopia residual, after that Europe, North America, and South America are roughly the same regarding happiness score and the other seven factors. Finally, Asia and Africa have the lowest scores in all fields.

Correlation plot for each continent

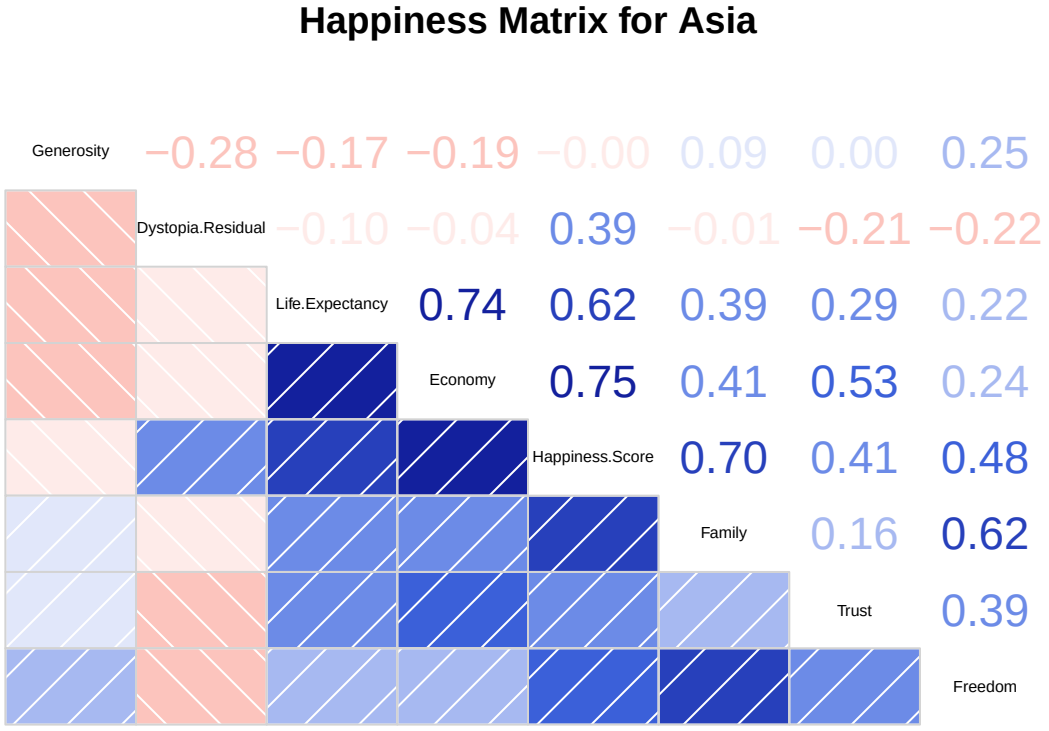
Let's see the correlation between variables for each continent.

```
corrgram(Happiness %>% select(-3) %>% filter(Continent == "Africa"), order=TRUE,
         upper.panel=panel.cor, main="Happiness Matrix for Africa")
```



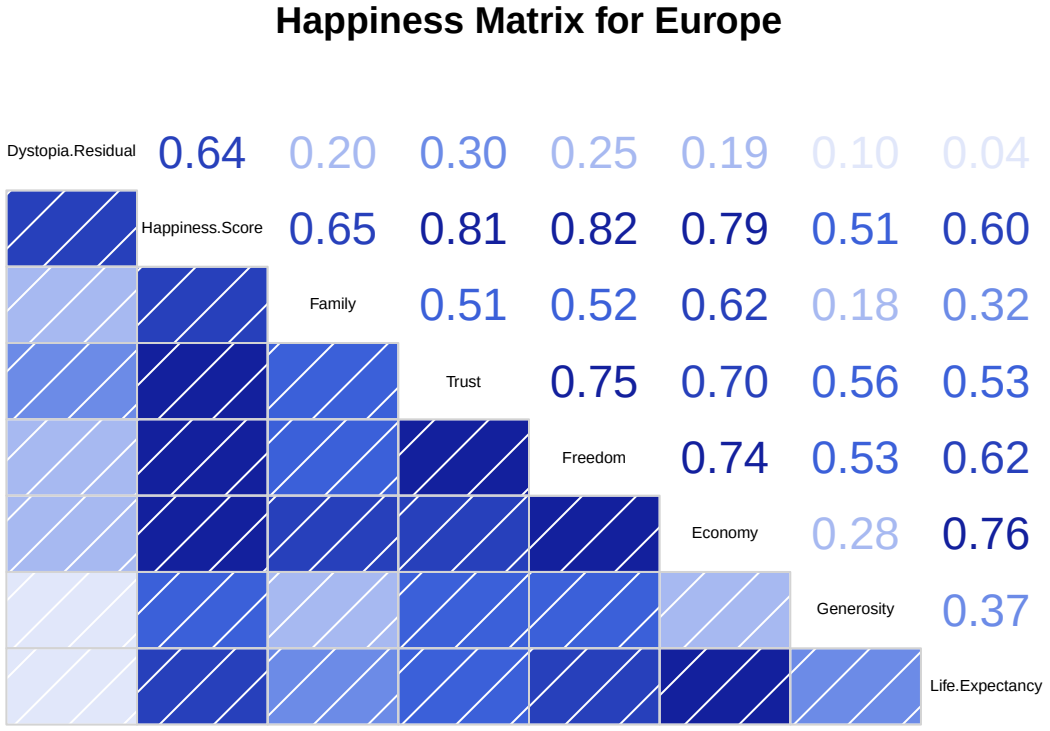
Correlation between "Happiness Score" and the other variables in Africa:
Economy > Family > Life.Expectancy > Dystopia.Residual > Freedom
There is no correlation between happiness score and trust.
There is an inverse correlation between happiness score and generosity.

```
corrgram(Happiness %>% select(-3) %>% filter(Continent == "Asia"), order=TRUE,
         upper.panel=panel.cor, main="Happiness Matrix for Asia")
```



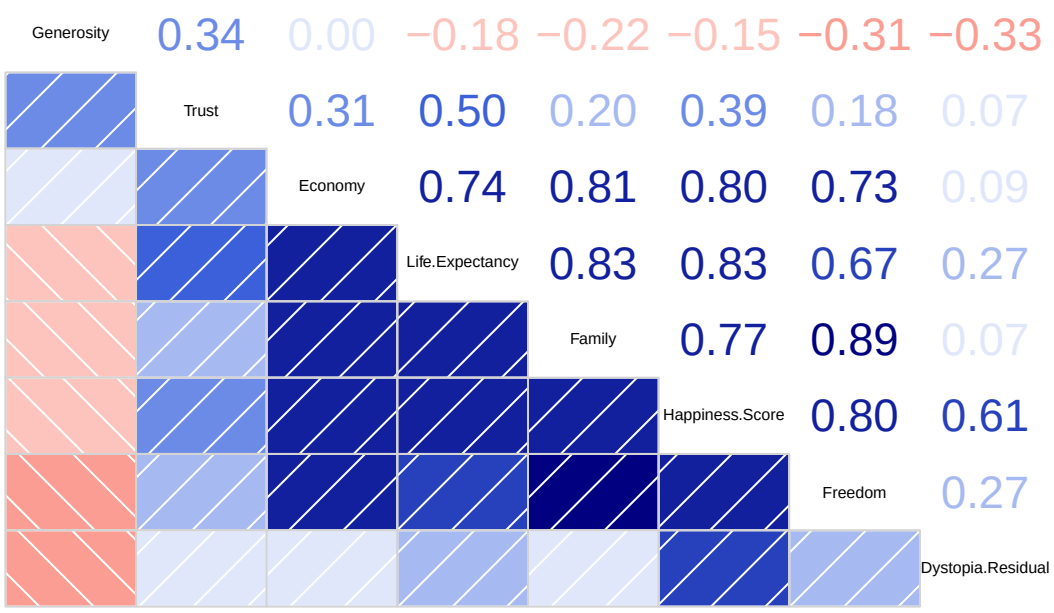
Correlation between "Happiness Score" and the other variables in Asia:
Economy > Family > Life.Expectancy > Freedom > Trust > Dystopia.Residual
There is no correlation between happiness score and generosity.

```
corrgram(Happiness %>% select(-3) %>% filter(Continent == "Europe"), order=TRUE,
         upper.panel=panel.cor, main="Happiness Matrix for Europe")
```



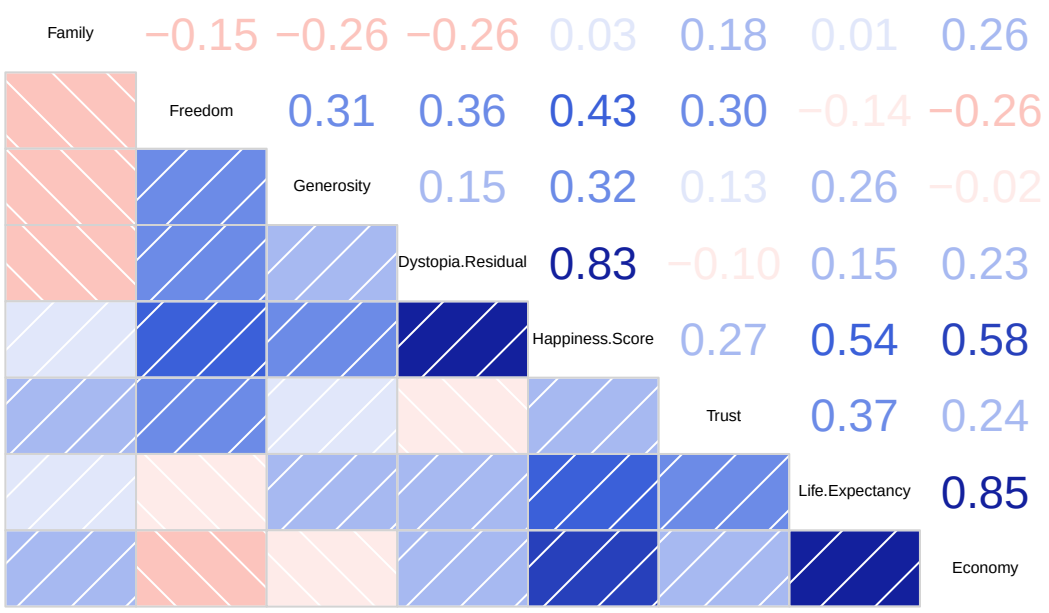
Correlation between "Happiness Score" and the other variables in Europe:
Freedom > Trust > Economy > Family > Dystopia.Residual > Life.Expectancy > Generosity
The highest correlation between generosity and happiness score took place in Europe.

```
corrgram(Happiness %>% select(-3) %>% filter(Continent == "North America"), order=TRUE,
         upper.panel=panel.cor, main="Happiness Matrix for North America")
```



Correlation between "Happiness Score" and the other variables in North America:
Life.Expectancy > Economy > Freedom > Family > Dystopia.Residual > Trust
There is an inverse correlation between happiness score and generosity.

```
corrgram(Happiness %>% select(-3) %>% filter(Continent == "South America"), order=TRUE,
         upper.panel=panel.cor, main="Happiness Matrix for South America")
```



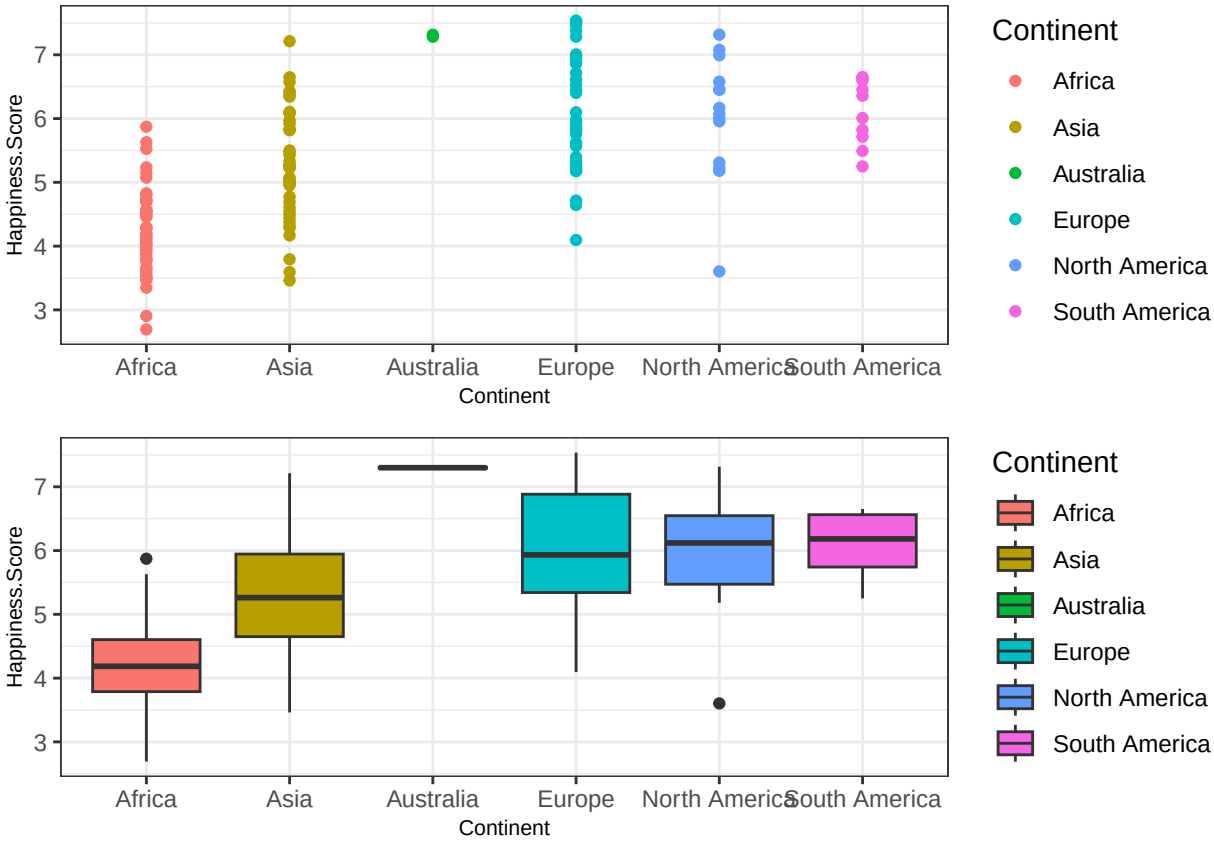
Correlation between "Happiness Score" and the other variables in South America:
Dystopia.Residual > Economy > Life.Expectancy > Freedom > Generosity > Trust > Family
The family is the least significant factor in South America.

Happiness score comparison on different continents

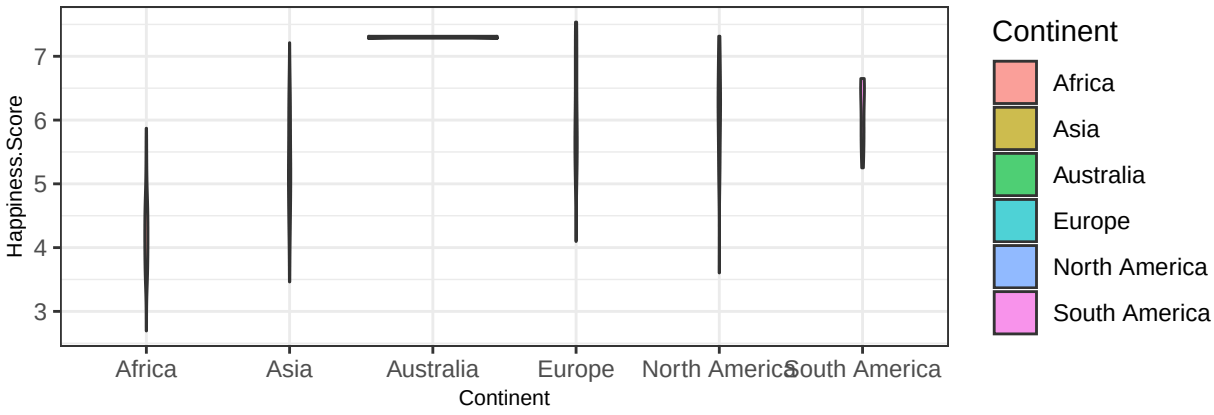
We will use scatter plot, box plot, and violin plot to see the happiness score distribution in different countries, how this score is populated in these continents and also will calculate the mean and median of happiness score for each of these continents.

```
####### Happiness score for each continent

gg1 <- ggplot(Happiness,
              aes(x=Continent,
                  y=Happiness.Score,
                  color=Continent))+
  geom_point() + theme_bw() +
  theme(axis.title = element_text(family = "Helvetica", size = (8)))

gg2 <- ggplot(Happiness , aes(x = Continent, y = Happiness.Score)) +
  geom_boxplot(aes(fill=Continent)) + theme_bw() +
  theme(axis.title = element_text(family = "Helvetica", size = (8)))

gg3 <- ggplot(Happiness,aes(x=Continent,y=Happiness.Score))+
  geom_violin(aes(fill=Continent),alpha=0.7)+ theme_bw() +
```

```
  theme(axis.title = element_text(family = "Helvetica", size = (8)))

# Compute descriptive statistics by groups
stable <- desc_statby(Happiness, measure.var = "Happiness.Score",
                      grps = "Continent")
stable <- stable[, c("Continent","mean","median")]
names(stable) <- c("Continent", "Mean of happiness score","Median of happiness score")
# Summary table plot
stable.p <- ggtexttable(stable,rows = NULL,
                        theme = ttheme("classic"))


ggarrange(gg1, gg2, ncol = 1, nrow = 2)
```




```
ggarrange(gg3, stable.p, ncol = 1, nrow = 2)
```



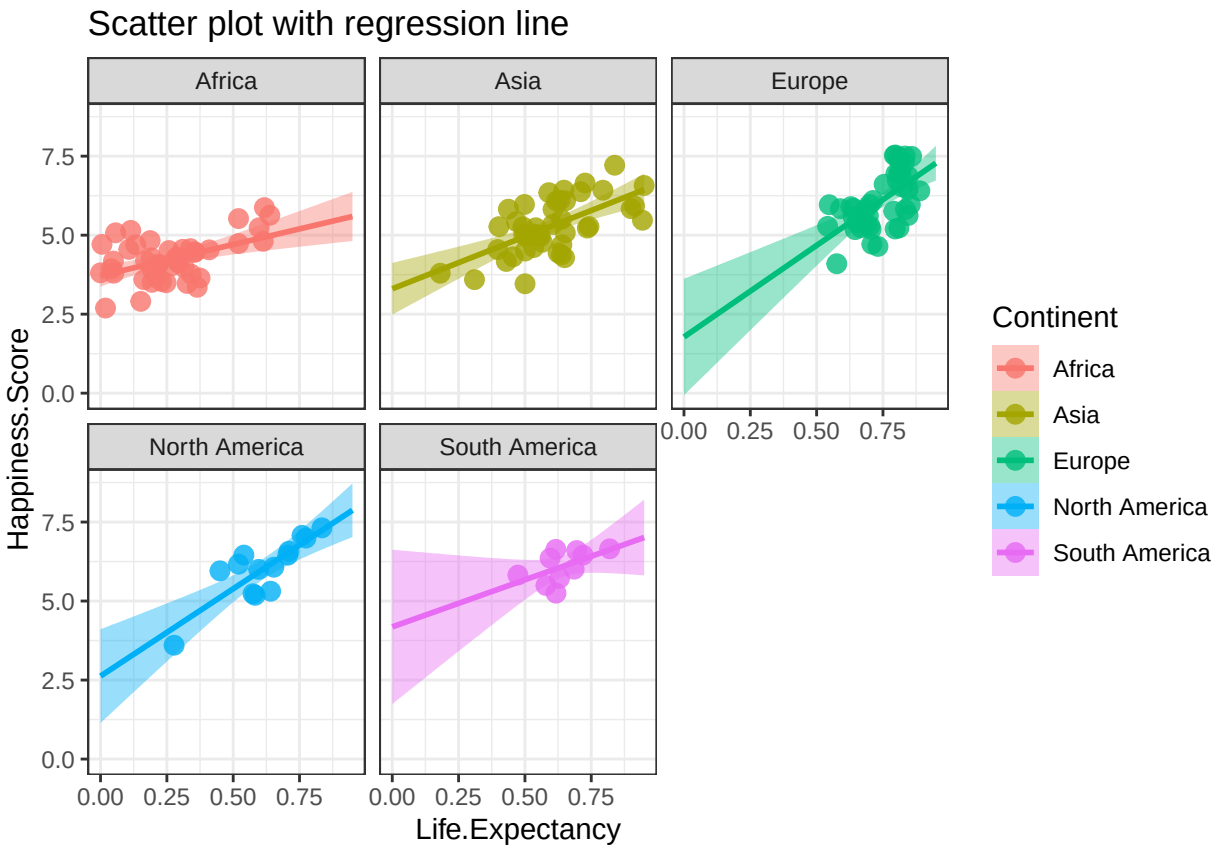
Continent	Mean of happiness score	Median of happiness score
Africa	4.239500	4.1850
Asia	5.284721	5.2620
Australia	7.299000	7.2990
Europe	6.097929	5.9325
North America	6.028214	6.1195
South America	6.098600	6.1825

As we have seen before, Australia has the highest median happiness score. Europe, South America, and North America are in the second place regarding median happiness score. Asia has the lowest median after Africa. We can see the range of happiness score for different continents, and also the concentration of happiness score.

Scatter plot with regression line

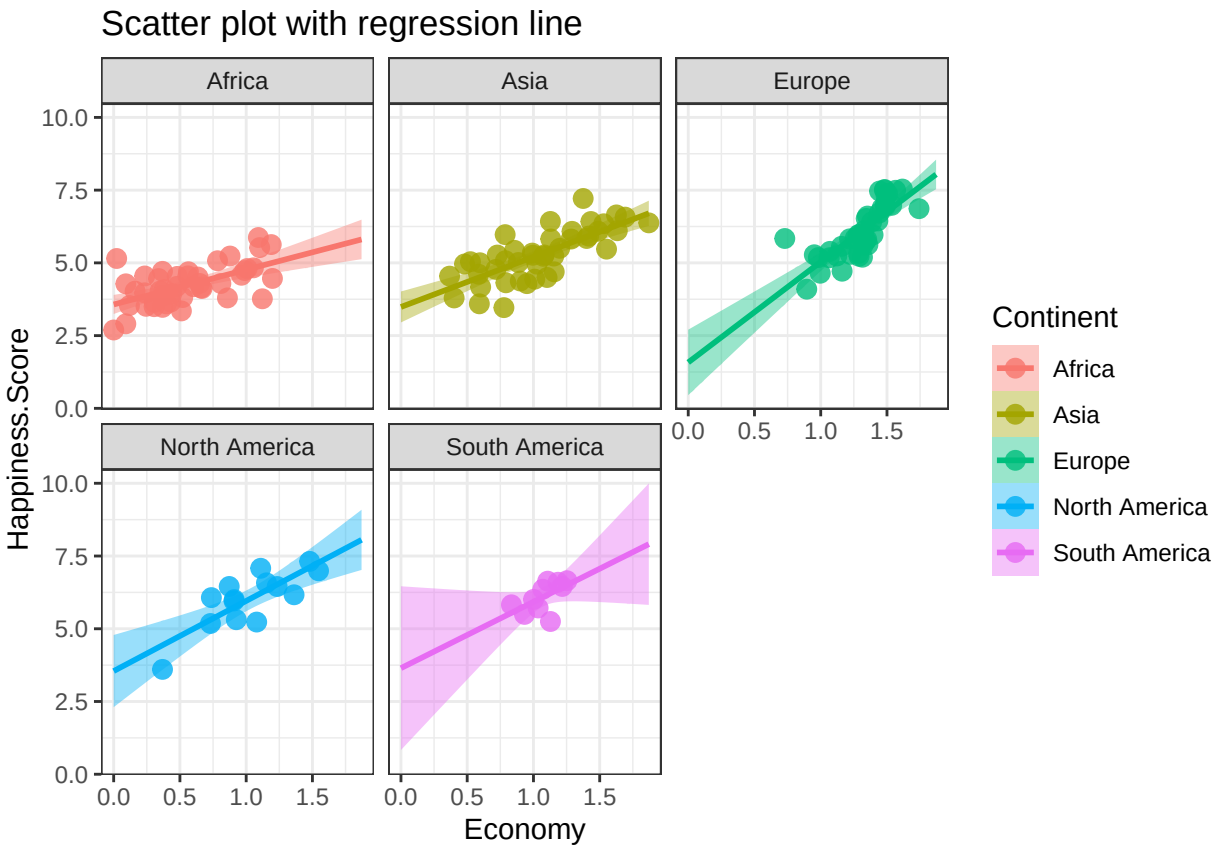
Let's see the correlation between happiness score and the other seven factors in the happiness dataset for different continents by creating a scatter plot.

```
ggplot(subset(Happiness, Happiness$Continent != "Australia"), aes(x = Life.Expectancy, y = Happiness.Sc
  geom_point(aes(color=Continent), size = 3, alpha = 0.8) +
  geom_smooth(aes(color = Continent, fill = Continent),
              method = "lm", fullrange = TRUE) +
  facet_wrap(~Continent) +
  theme_bw() + labs(title = "Scatter plot with regression line")
```



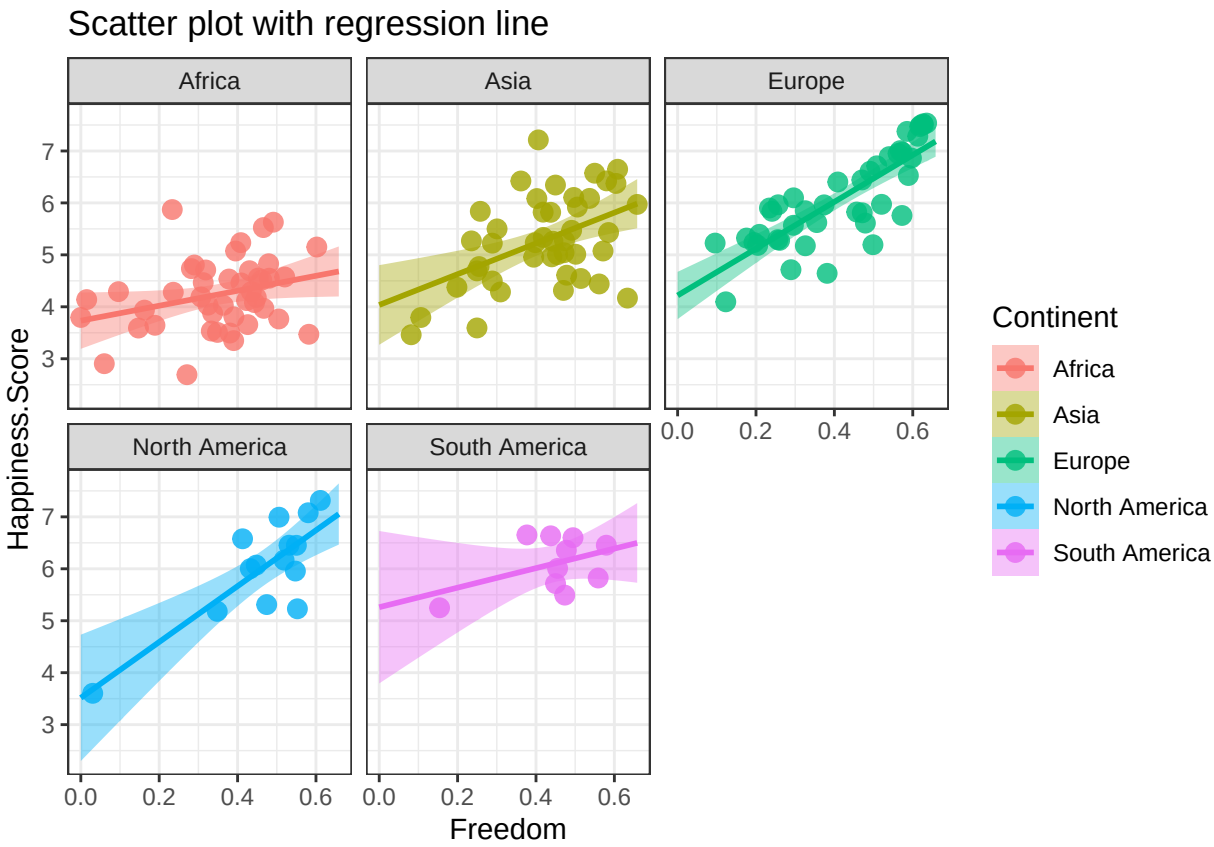
The correlation between life expectancy and happiness score in Europe, North America, and Asia is more significant than the other continents. Worth mentioning that we will not take Australia into account because there are just two countries in Australia and creating scatter plot with the regression line for this continent will not give us any insight.

```
ggplot(subset(Happiness, Happiness$Continent != "Australia"), aes(x = Economy, y = Happiness.Score)) +
  geom_point(aes(color=Continent), size = 3, alpha = 0.8) +
  geom_smooth(aes(color = Continent, fill = Continent),
              method = "lm", fullrange = TRUE) +
  facet_wrap(~Continent) +
  theme_bw() + labs(title = "Scatter plot with regression line")
```



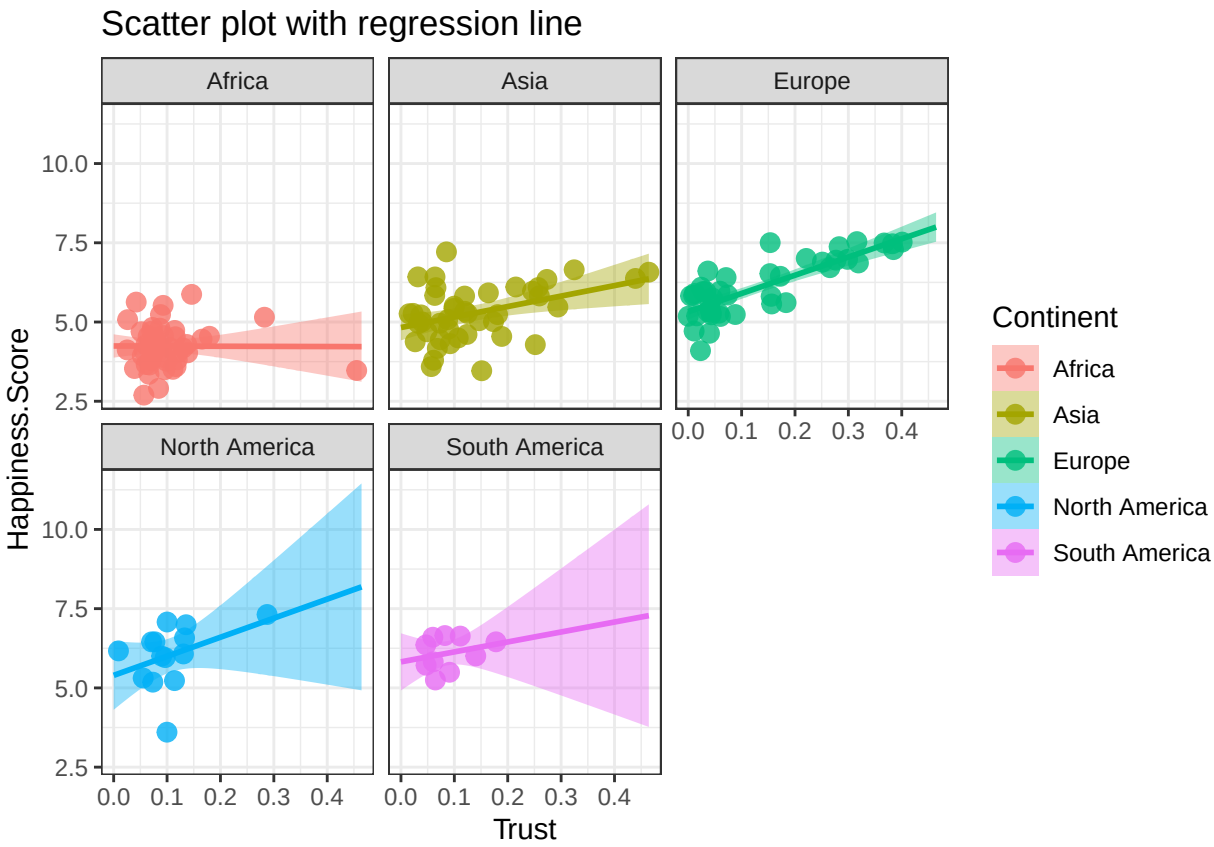
We can see pretty the same result here for the correlation between happiness score and economy. Africa has the lowest relationship in this regard.

```
ggplot(subset(Happiness, Happiness$Continent != "Australia"), aes(x = Freedom, y = Happiness.Score)) +
  geom_point(aes(color=Continent), size = 3, alpha = 0.8) +
  geom_smooth(aes(color = Continent, fill = Continent),
              method = "lm", fullrange = TRUE) +
  facet_wrap(~Continent) +
  theme_bw() + labs(title = "Scatter plot with regression line")
```



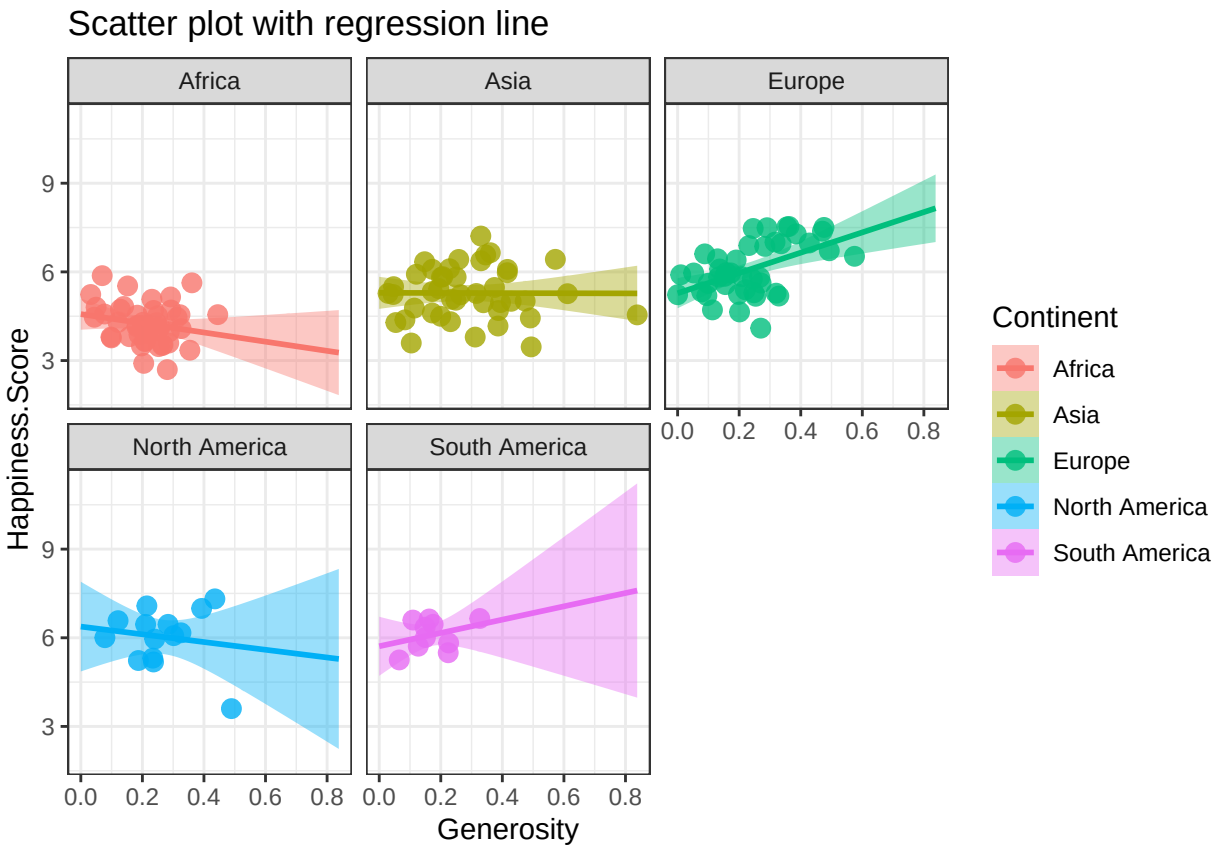
Freedom in Europe and North America is more correlated to happiness score than any other continents.

```
ggplot(subset(Happiness, Happiness$Continent != "Australia"), aes(x = Trust, y = Happiness.Score)) +
  geom_point(aes(color=Continent), size = 3, alpha = 0.8) +
  geom_smooth(aes(color = Continent, fill = Continent),
              method = "lm", fullrange = TRUE) +
  facet_wrap(~Continent) +
  theme_bw() + labs(title = "Scatter plot with regression line")
```



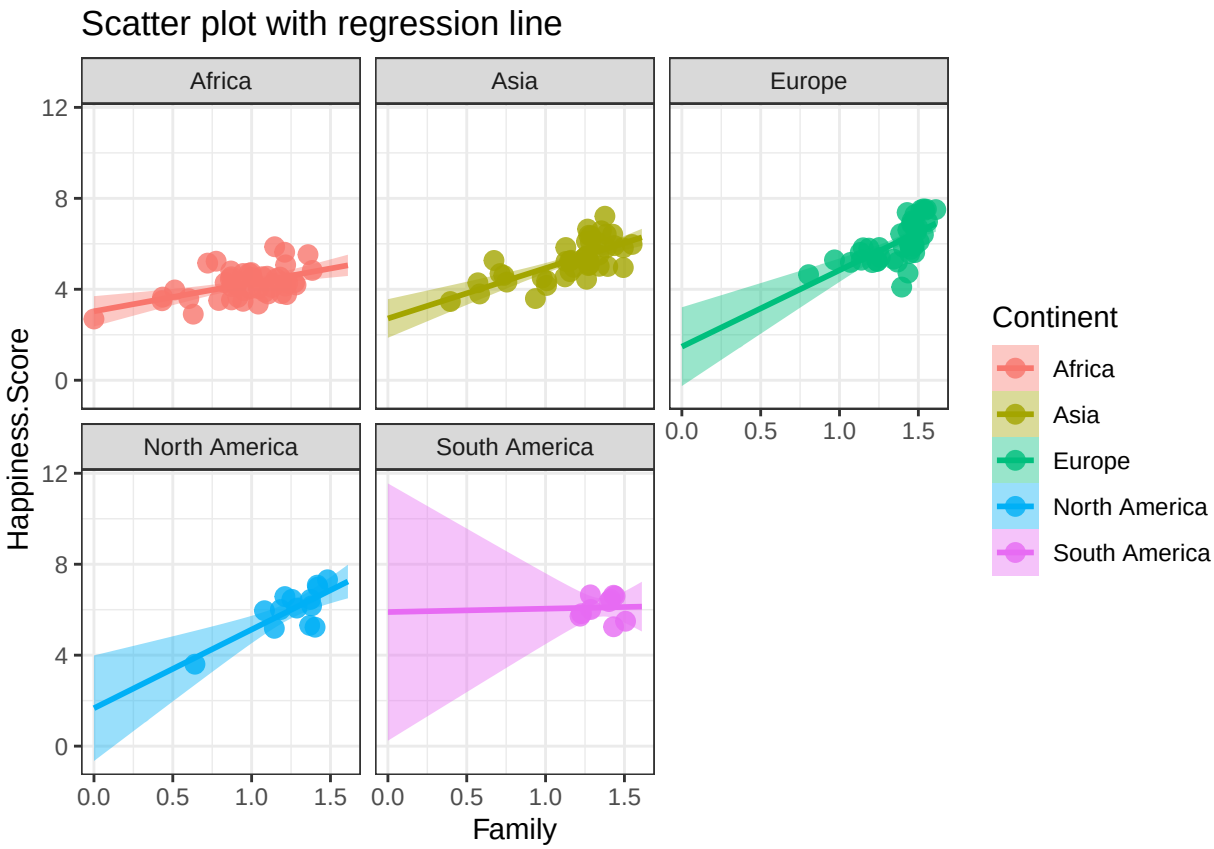
Approximately there is no correlation between trust and happiness score in Africa.

```
ggplot(subset(Happiness, Happiness$Continent != "Australia"), aes(x = Generosity, y = Happiness.Score))
  geom_point(aes(color=Continent), size = 3, alpha = 0.8) +
  geom_smooth(aes(color = Continent, fill = Continent),
              method = "lm", fullrange = TRUE) +
  facet_wrap(~Continent) +
  theme_bw() + labs(title = "Scatter plot with regression line")
```



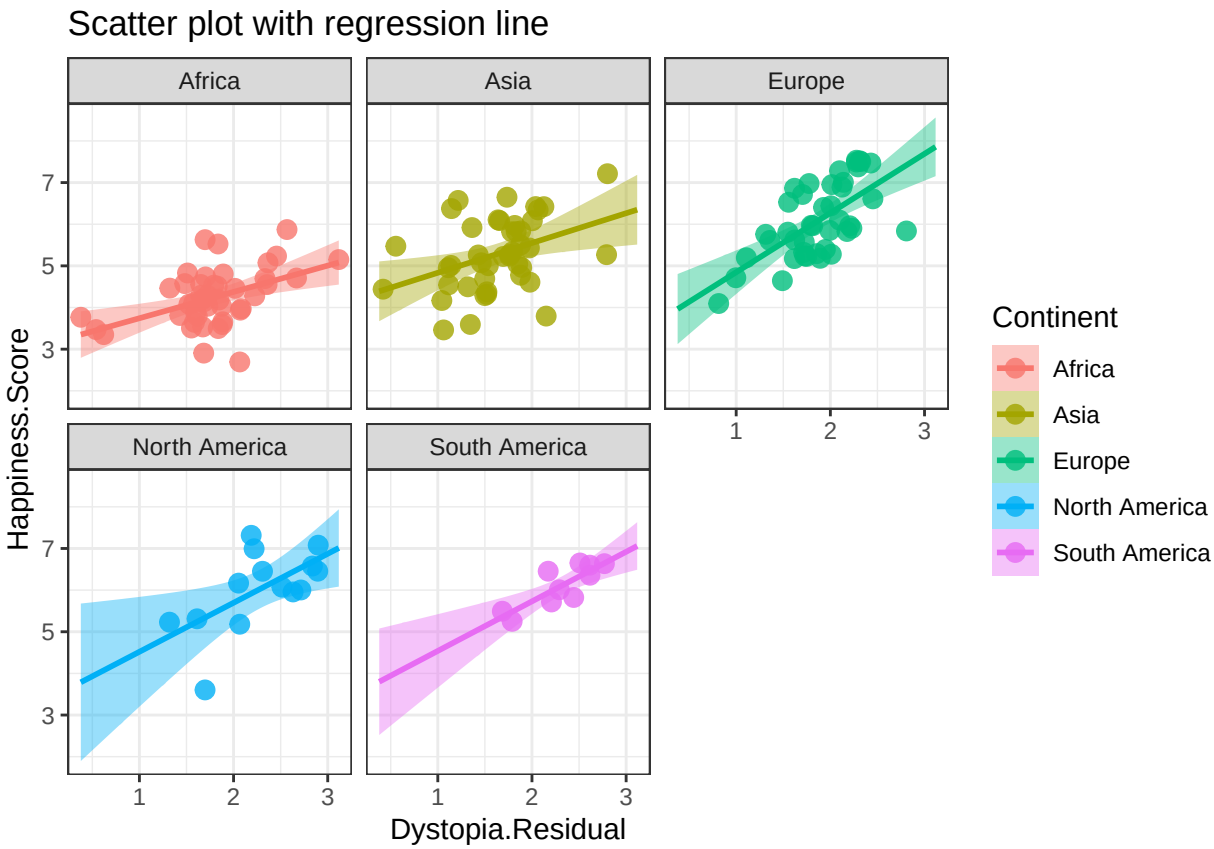
The regression line has a positive slope only for Europe and South America. For Asia the line is horizontal, and for Africa and North America the slope is negative.

```
ggplot(subset(Happiness, Happiness$Continent != "Australia"), aes(x = Family, y = Happiness.Score)) +
  geom_point(aes(color=Continent), size = 3, alpha = 0.8) +
  geom_smooth(aes(color = Continent, fill = Continent),
              method = "lm", fullrange = TRUE) +
  facet_wrap(~Continent) +
  theme_bw() + labs(title = "Scatter plot with regression line")
```



In South America with increase in the family score, the happiness score remains constant.

```
ggplot(subset(Happiness, Happiness$Continent != "Australia"), aes(x = Dystopia.Residual, y = Happiness.S
  geom_point(aes(color=Continent), size = 3, alpha = 0.8) +
  geom_smooth(aes(color = Continent, fill = Continent),
              method = "lm", fullrange = TRUE) +
  facet_wrap(~Continent) +
  theme_bw() + labs(title = "Scatter plot with regression line")
```



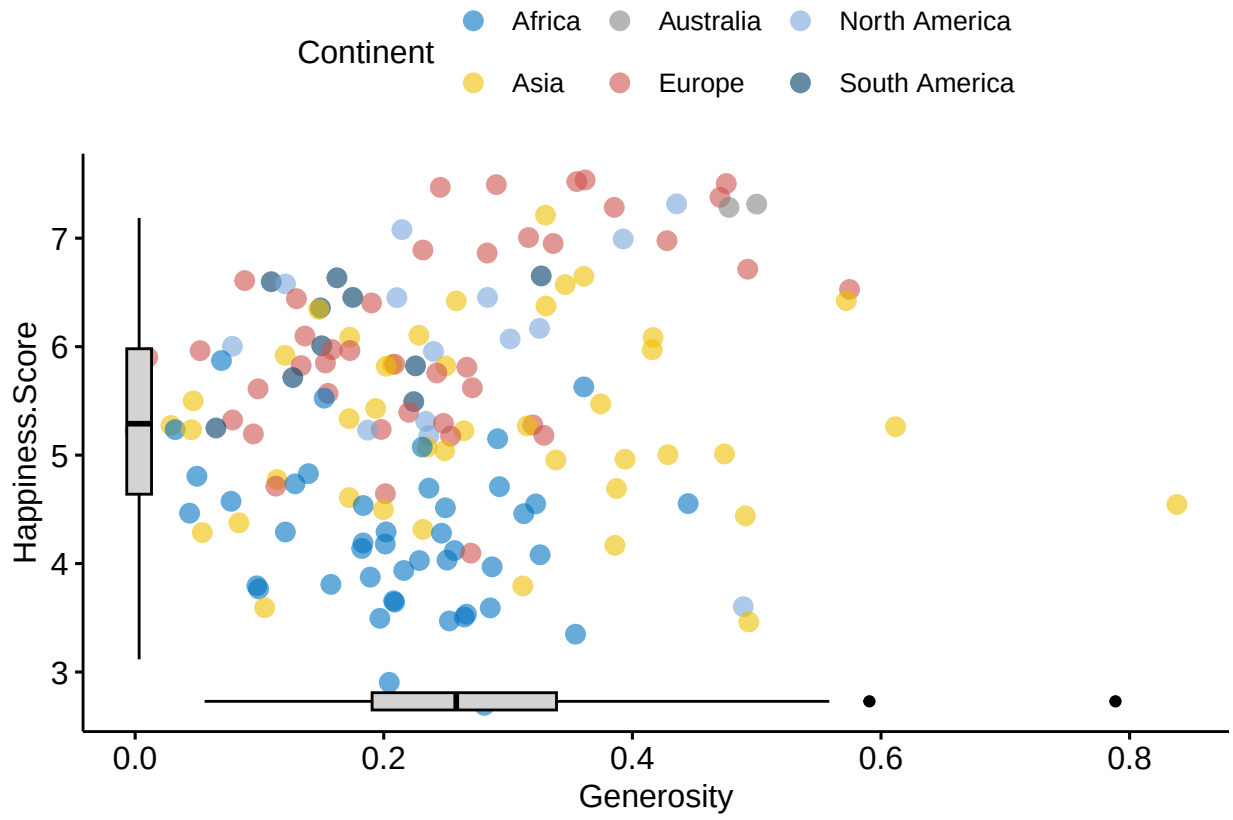
All continents act pretty the same regarding dystopia residual.

Scatter plot colored by Continents

The following is just another way of seeing happiness score distribution on different continents when taking the correlation of happiness score with different variables into account.

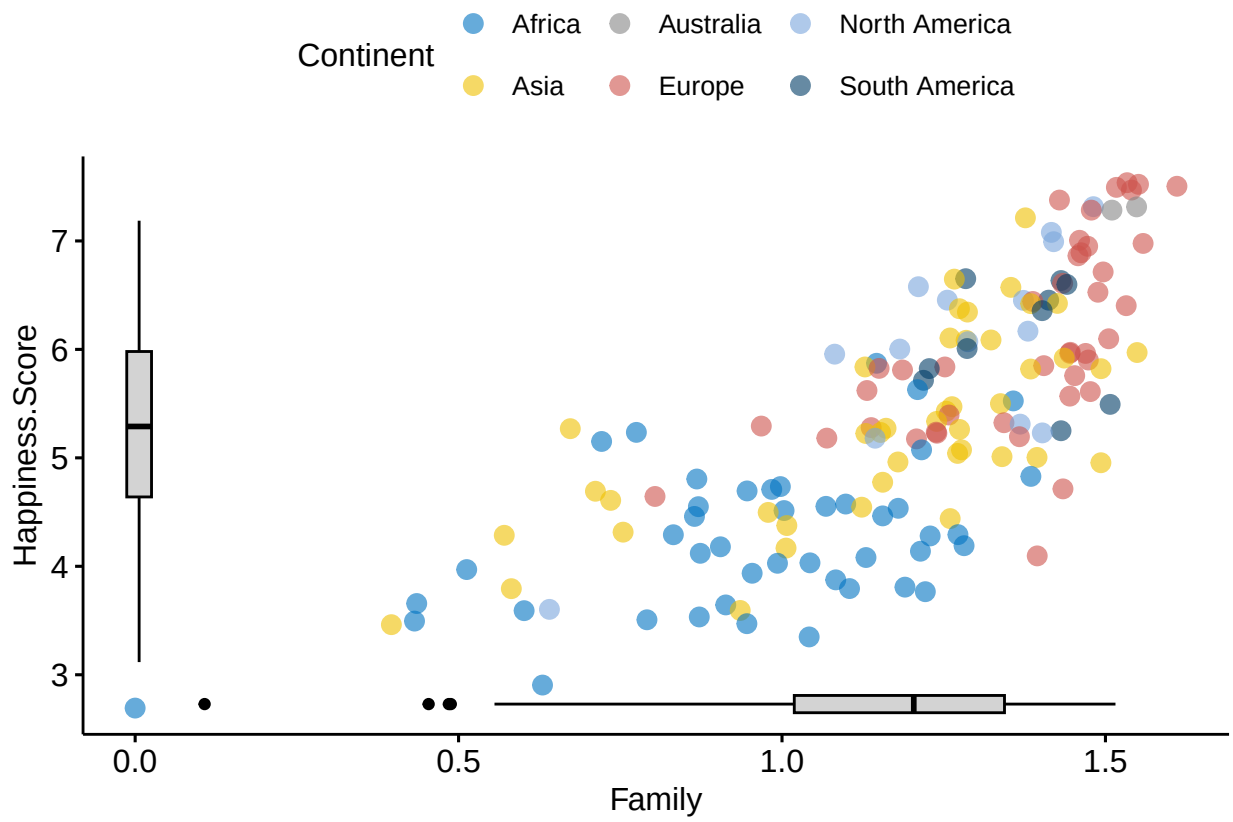
```
#:::::::::::::::::::::::::::Generosity:::::::::::::::::::::::::::::
sp <- ggscatter(Happiness, x = "Generosity", y = "Happiness.Score",
                color = "Continent", palette = "jco",
                size = 3, alpha = 0.6)
# Create box plots of x/y variables
# Box plot of the x variable
xbp <- ggboxplot(Happiness$Generosity, width = 0.3, fill = "lightgray") +
  rotate() +
  theme_transparent()
# Box plot of the y variable
ybp <- ggboxplot(Happiness$Happiness.Score, width = 0.3, fill = "lightgray") +
  theme_transparent()
# Create the external graphical objects
# called a "grop" in Grid terminology
xbp_grob <- ggplotGrob(xbp)
ybp_grob <- ggplotGrob(ybp)
# Place box plots inside the scatter plot
xmin <- min(Happiness$Generosity); xmax <- max(Happiness$Generosity)
```

```
ymin <- min(Happiness$Happiness.Score); ymax <- max(Happiness$Happiness.Score)
yoffset <- (1/15)*ymax; xoffset <- (1/15)*xmax
# Insert xbp_grob inside the scatter plot
sp + annotation_custom(grob = xbp_grob, xmin = xmin, xmax = xmax,
                       ymin = ymin-yoffset, ymax = ymin+yoffset) +
  # Insert ybp_grob inside the scatter plot
  annotation_custom(grob = ybp_grob,
                    xmin = xmin-xoffset, xmax = xmin+xoffset,
                    ymin = ymin, ymax = ymax)
```



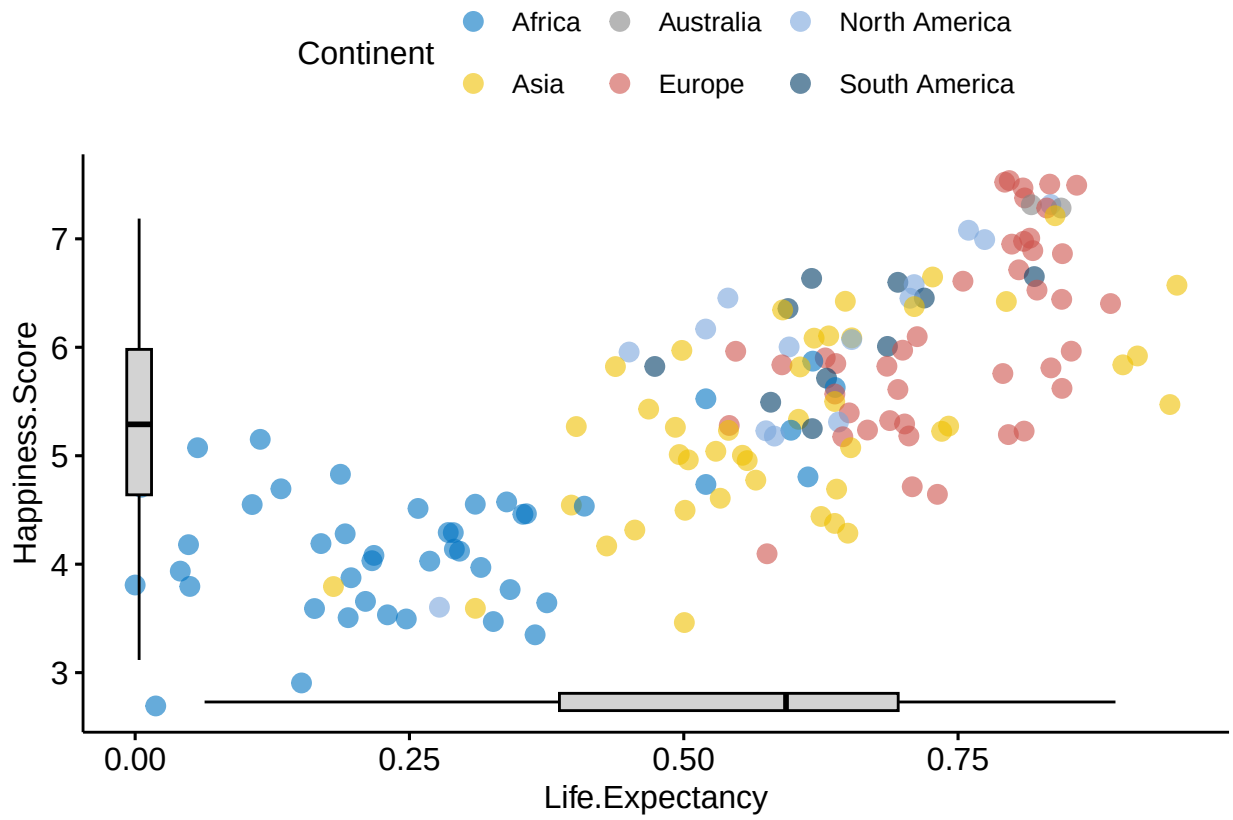

```
#:::::::::::::::::::::::::::Family::::::::::::::::::::::::::::::
sp <- ggscatter(Happiness, x = "Family", y = "Happiness.Score",
                color = "Continent", palette = "jco",
                size = 3, alpha = 0.6)
# Create box plots of x/y variables
# Box plot of the x variable
xbp <- ggboxplot(Happiness$Family, width = 0.3, fill = "lightgray") +
  rotate() +
  theme_transparent()
# Box plot of the y variable
ybp <- ggboxplot(Happiness$Happiness.Score, width = 0.3, fill = "lightgray") +
  theme_transparent()
# Create the external graphical objects
# called a "grop" in Grid terminology
xbp_grob <- ggplotGrob(xbp)
```

```
ybp_grob <- ggplotGrob(ybp)
# Place box plots inside the scatter plot
xmin <- min(Happiness$Family); xmax <- max(Happiness$Family)
ymin <- min(Happiness$Happiness.Score); ymax <- max(Happiness$Happiness.Score)
yoffset <- (1/15)*ymax; xoffset <- (1/15)*xmax
# Insert xbp_grob inside the scatter plot
sp + annotation_custom(grob = xbp_grob, xmin = xmin, xmax = xmax,
                       ymin = ymin-yoffset, ymax = ymin+yoffset) +
  # Insert ybp_grob inside the scatter plot
  annotation_custom(grob = ybp_grob,
                    xmin = xmin-xoffset, xmax = xmin+xoffset,
                    ymin = ymin, ymax = ymax)
```



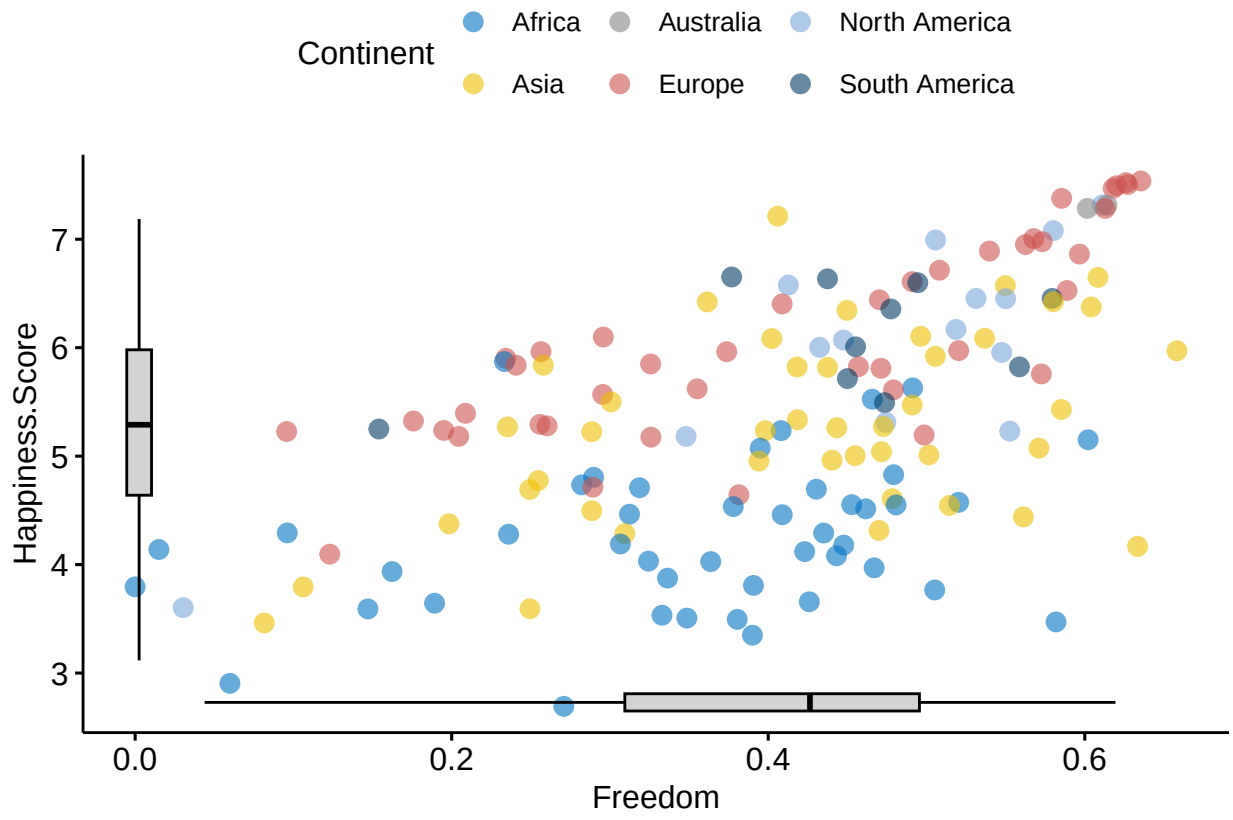

```
#:::::::::::::::::::::::::::Life.Expectancy:::::::::::::::::::::::::::::
sp <- ggscatter(Happiness, x = "Life.Expectancy", y = "Happiness.Score",
                color = "Continent", palette = "jco",
                size = 3, alpha = 0.6)
# Create box plots of x/y variables
# Box plot of the x variable
xbp <- ggboxplot(Happiness$Life.Expectancy, width = 0.3, fill = "lightgray") +
  rotate() +
  theme_transparent()
# Box plot of the y variable
ybp <- ggboxplot(Happiness$Happiness.Score, width = 0.3, fill = "lightgray") +
  theme_transparent()
```

```r
# Create the external graphical objects
# called a "grop" in Grid terminology
xbp_grob <- ggplotGrob(xbp)
ybp_grob <- ggplotGrob(ybp)
# Place box plots inside the scatter plot
xmin <- min(Happiness$Life.Expectancy); xmax <- max(Happiness$Life.Expectancy)
ymin <- min(Happiness$Happiness.Score); ymax <- max(Happiness$Happiness.Score)
yoffset <- (1/15)*ymax; xoffset <- (1/15)*xmax
# Insert xbp_grob inside the scatter plot
sp + annotation_custom(grob = xbp_grob, xmin = xmin, xmax = xmax,
                       ymin = ymin-yoffset, ymax = ymin+yoffset) +
  # Insert ybp_grob inside the scatter plot
  annotation_custom(grob = ybp_grob,
                    xmin = xmin-xoffset, xmax = xmin+xoffset,
                    ymin = ymin, ymax = ymax)
```



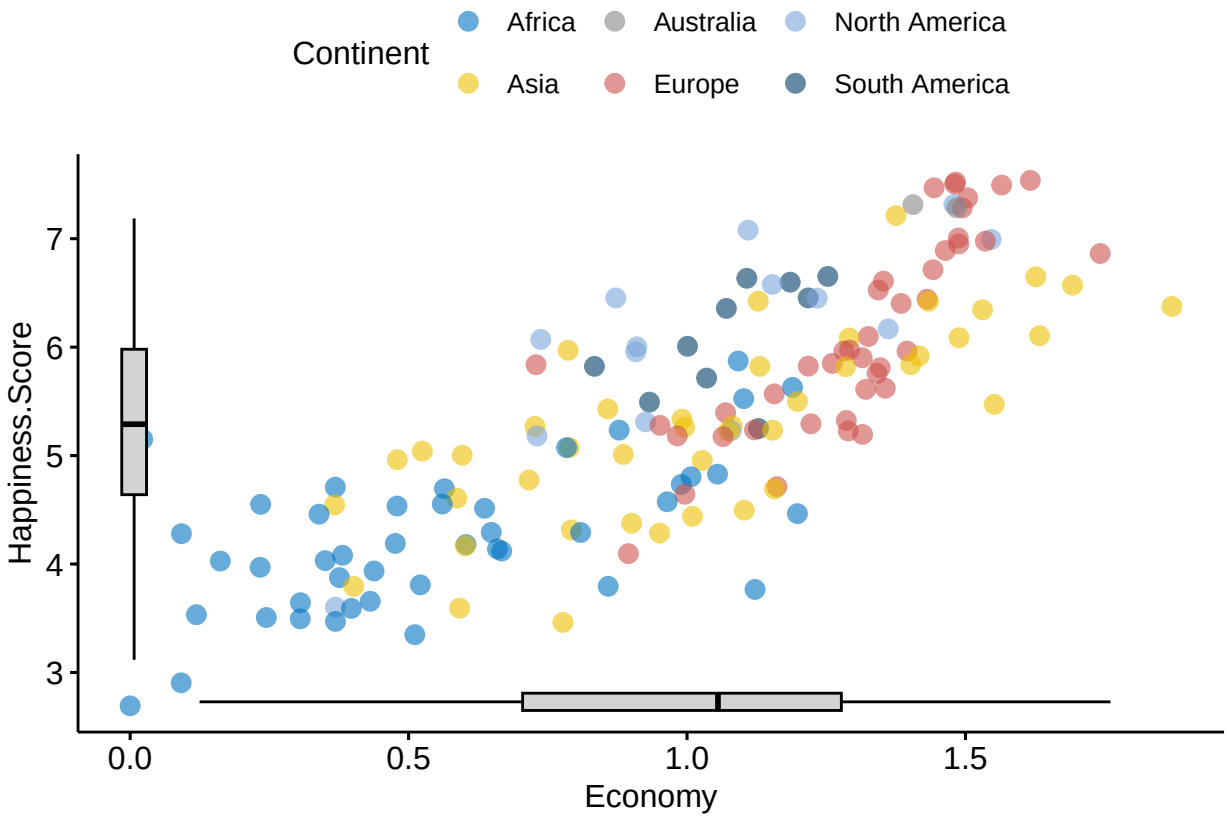

```r
#:::::::::::::::::::::::::::Freedom:::::::::::::::::::::::::::::
sp <- ggscatter(Happiness, x = "Freedom", y = "Happiness.Score",
                color = "Continent", palette = "jco",
                size = 3, alpha = 0.6)
# Create box plots of x/y variables
# Box plot of the x variable
xbp <- ggboxplot(Happiness$Freedom, width = 0.3, fill = "lightgray") +
  rotate() +
  theme_transparent()
```

```r
# Box plot of the y variable
ybp <- ggboxplot(Happiness$Happiness.Score, width = 0.3, fill = "lightgray") +
  theme_transparent()
# Create the external graphical objects
# called a "grop" in Grid terminology
xbp_grob <- ggplotGrob(xbp)
ybp_grob <- ggplotGrob(ybp)
# Place box plots inside the scatter plot
xmin <- min(Happiness$Freedom); xmax <- max(Happiness$Freedom)
ymin <- min(Happiness$Happiness.Score); ymax <- max(Happiness$Happiness.Score)
yoffset <- (1/15)*ymax; xoffset <- (1/15)*xmax
# Insert xbp_grob inside the scatter plot
sp + annotation_custom(grob = xbp_grob, xmin = xmin, xmax = xmax,
                       ymin = ymin-yoffset, ymax = ymin+yoffset) +
  # Insert ybp_grob inside the scatter plot
  annotation_custom(grob = ybp_grob,
                    xmin = xmin-xoffset, xmax = xmin+xoffset,
                    ymin = ymin, ymax = ymax)
```



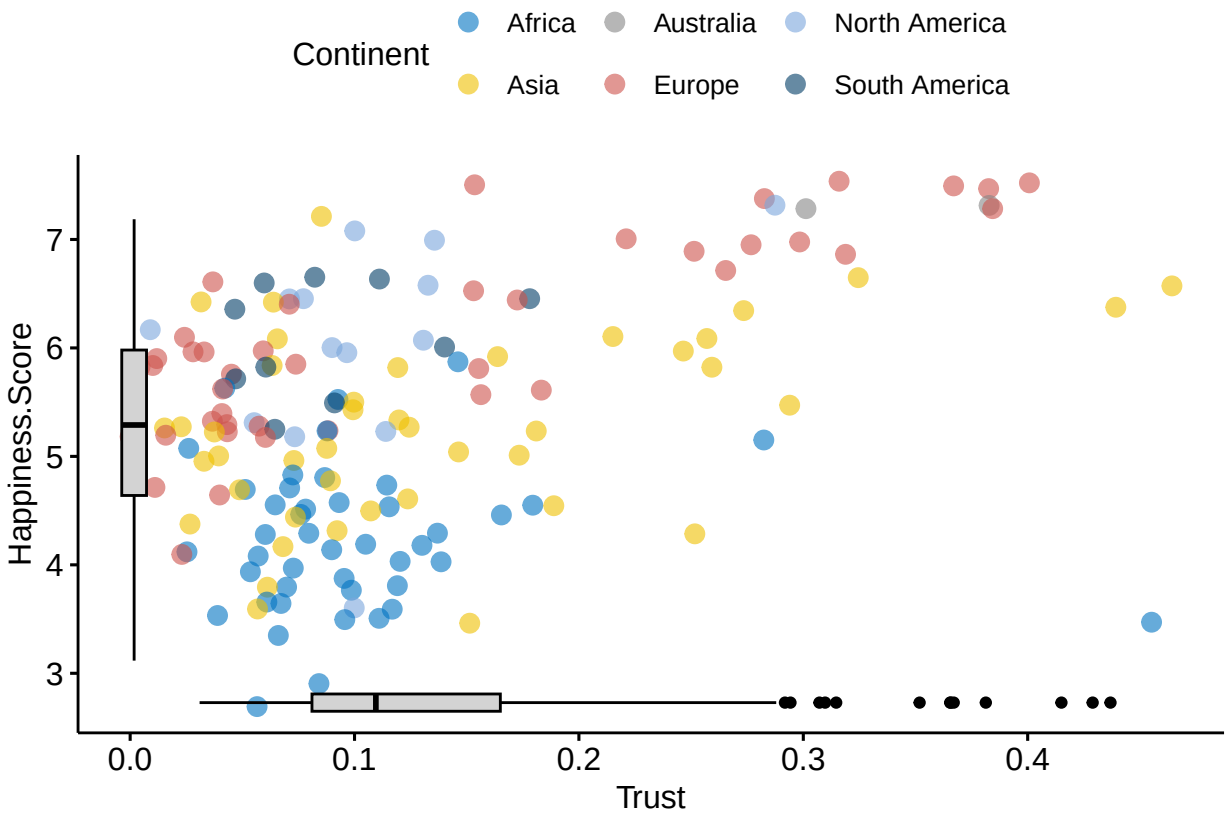

```r
#:::::::::::::::::::::::::::Economy:::::::::::::::::::::::::::::
sp <- ggscatter(Happiness, x = "Economy", y = "Happiness.Score",
                color = "Continent", palette = "jco",
                size = 3, alpha = 0.6)
# Create box plots of x/y variables
# Box plot of the x variable
```

```
xbp <- ggboxplot(Happiness$Economy, width = 0.3, fill = "lightgray") +
  rotate() +
  theme_transparent()
# Box plot of the y variable
ybp <- ggboxplot(Happiness$Happiness.Score, width = 0.3, fill = "lightgray") +
  theme_transparent()
# Create the external graphical objects
# called a "grop" in Grid terminology
xbp_grob <- ggplotGrob(xbp)
ybp_grob <- ggplotGrob(ybp)
# Place box plots inside the scatter plot
xmin <- min(Happiness$Economy); xmax <- max(Happiness$Economy)
ymin <- min(Happiness$Happiness.Score); ymax <- max(Happiness$Happiness.Score)
yoffset <- (1/15)*ymax; xoffset <- (1/15)*xmax
# Insert xbp_grob inside the scatter plot
sp + annotation_custom(grob = xbp_grob, xmin = xmin, xmax = xmax,
                       ymin = ymin-yoffset, ymax = ymin+yoffset) +
  # Insert ybp_grob inside the scatter plot
  annotation_custom(grob = ybp_grob,
                    xmin = xmin-xoffset, xmax = xmin+xoffset,
                    ymin = ymin, ymax = ymax)
```

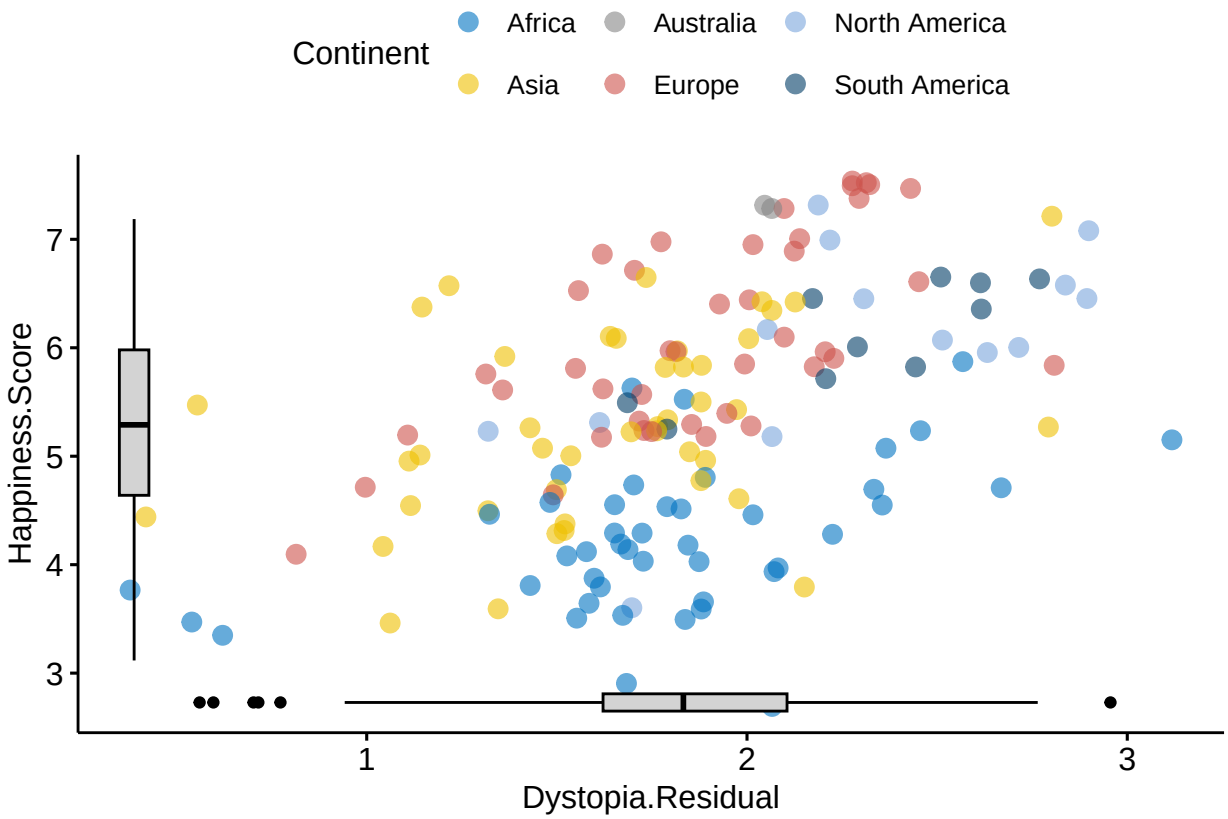



```
#:::::::::::::::::::::::::::Trust::::::::::::::::::::::::::::::
sp <- ggscatter(Happiness, x = "Trust", y = "Happiness.Score",
                color = "Continent", palette = "jco",
```

```
                size = 3, alpha = 0.6)
# Create box plots of x/y variables
# Box plot of the x variable
xbp <- ggboxplot(Happiness$Trust, width = 0.3, fill = "lightgray") +
  rotate() +
  theme_transparent()
# Box plot of the y variable
ybp <- ggboxplot(Happiness$Happiness.Score, width = 0.3, fill = "lightgray") +
  theme_transparent()
# Create the external graphical objects
# called a "grop" in Grid terminology
xbp_grob <- ggplotGrob(xbp)
ybp_grob <- ggplotGrob(ybp)
# Place box plots inside the scatter plot
xmin <- min(Happiness$Trust); xmax <- max(Happiness$Trust)
ymin <- min(Happiness$Happiness.Score); ymax <- max(Happiness$Happiness.Score)
yoffset <- (1/15)*ymax; xoffset <- (1/15)*xmax
# Insert xbp_grob inside the scatter plot
sp + annotation_custom(grob = xbp_grob, xmin = xmin, xmax = xmax,
                       ymin = ymin-yoffset, ymax = ymin+yoffset) +
  # Insert ybp_grob inside the scatter plot
  annotation_custom(grob = ybp_grob,
                    xmin = xmin-xoffset, xmax = xmin+xoffset,
                    ymin = ymin, ymax = ymax)
```



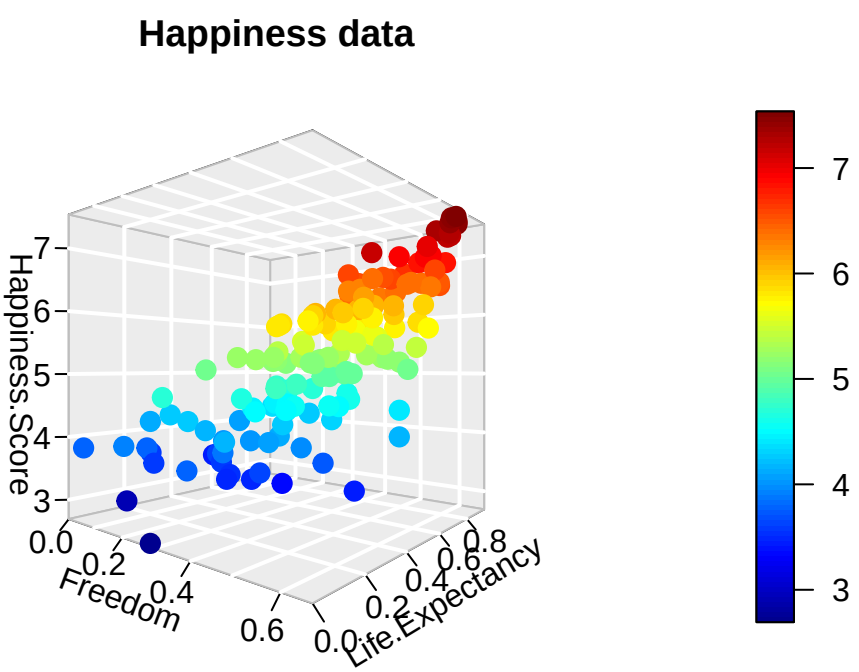
```r
#::::::::::::::::::::::::::Dystopia.Residual::::::::::::::::::::::::::::
sp <- ggscatter(Happiness, x = "Dystopia.Residual", y = "Happiness.Score",
                color = "Continent", palette = "jco",
                size = 3, alpha = 0.6)
# Create box plots of x/y variables
# Box plot of the x variable
xbp <- ggboxplot(Happiness$Dystopia.Residual, width = 0.3, fill = "lightgray") +
  rotate() +
  theme_transparent()
# Box plot of the y variable
ybp <- ggboxplot(Happiness$Happiness.Score, width = 0.3, fill = "lightgray") +
  theme_transparent()
# Create the external graphical objects
# called a "grop" in Grid terminology
xbp_grob <- ggplotGrob(xbp)
ybp_grob <- ggplotGrob(ybp)
# Place box plots inside the scatter plot
xmin <- min(Happiness$Dystopia.Residual); xmax <- max(Happiness$Dystopia.Residual)
ymin <- min(Happiness$Happiness.Score); ymax <- max(Happiness$Happiness.Score)
yoffset <- (1/15)*ymax; xoffset <- (1/15)*xmax
# Insert xbp_grob inside the scatter plot
sp + annotation_custom(grob = xbp_grob, xmin = xmin, xmax = xmax,
                       ymin = ymin-yoffset, ymax = ymin+yoffset) +
  # Insert ybp_grob inside the scatter plot
  annotation_custom(grob = ybp_grob,
                    xmin = xmin-xoffset, xmax = xmin+xoffset,
                    ymin = ymin, ymax = ymax)
```



3D Plot

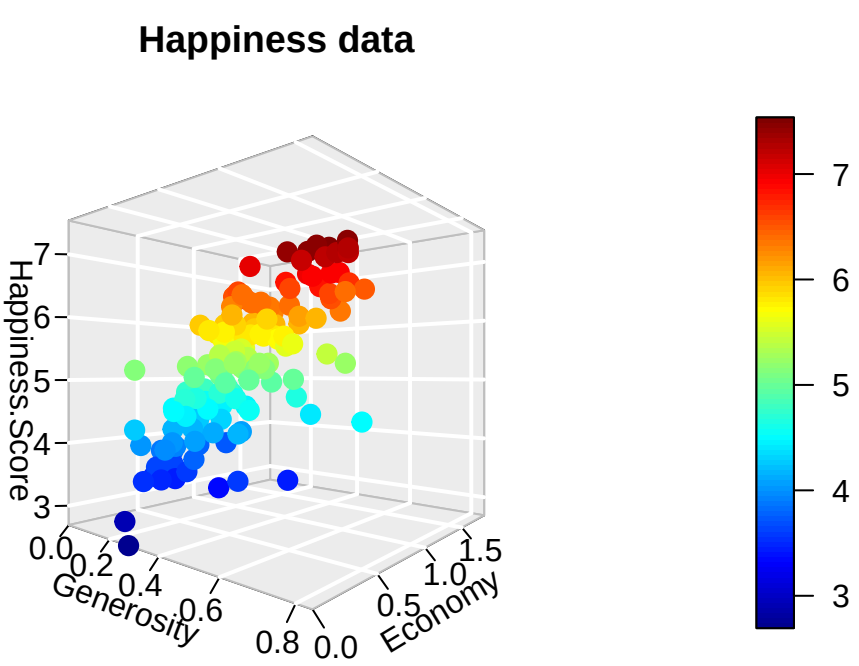
For the last part of our visualization let's create some fancy plots. I have to indicate that I am not in favor of 3D plots or any fancy plots but let's do this just for fun!

```
scatter3D(Happiness$Freedom, Happiness$Life.Expectancy, Happiness$Happiness.Score, phi = 0, bty = "g",
          pch = 20, cex = 2, ticktype = "detailed",
          main = "Happiness data", xlab = "Freedom",
          ylab ="Life.Expectancy", zlab = "Happiness.Score")
```



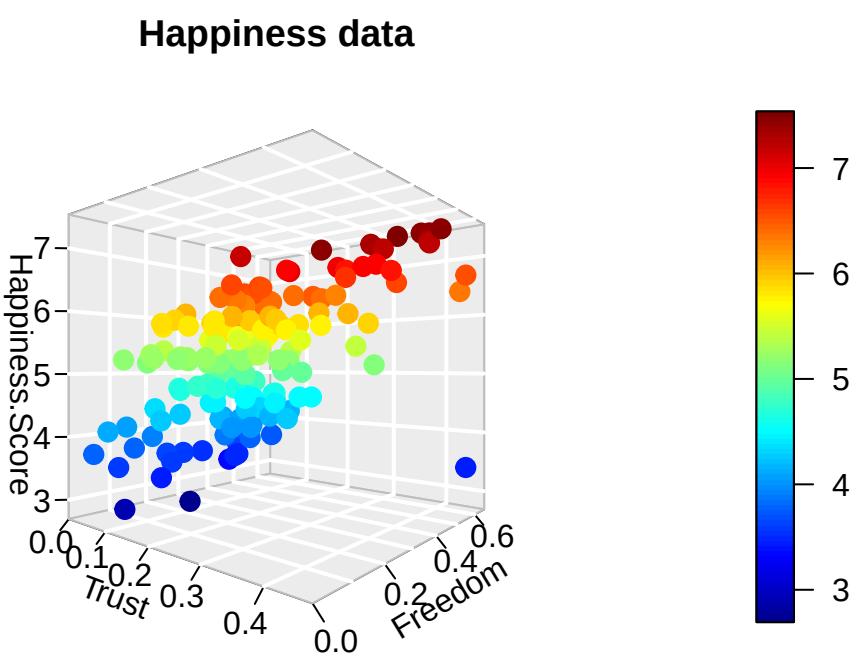
According to this plot, the higher the life expectancy and freedom scores, the higher will be the happiness score.

```
scatter3D(Happiness$Generosity, Happiness$Economy, Happiness$Happiness.Score, phi = 0, bty = "g",
          pch = 20, cex = 2, ticktype = "detailed",
          main = "Happiness data", xlab = "Generosity",
          ylab ="Economy", zlab = "Happiness.Score")
```



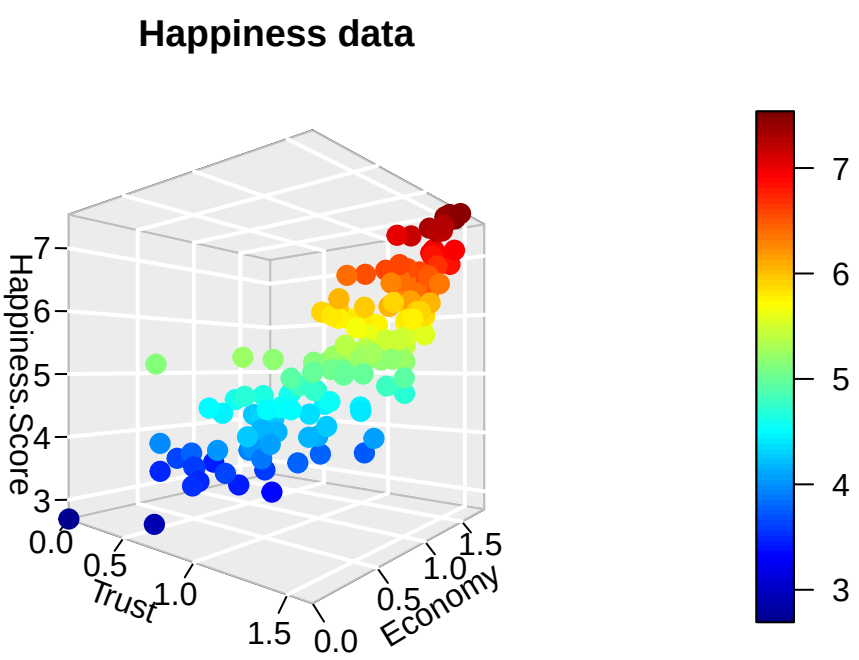
The higher economy score and the lower generosity score will lead to the higher level of happiness.

```
scatter3D(Happiness$Trust, Happiness$Freedom, Happiness$Happiness.Score, phi = 0, bty = "g",
          pch = 20, cex = 2, ticktype = "detailed",
          main = "Happiness data", xlab = "Trust",
          ylab ="Freedom", zlab = "Happiness.Score")
```



In general, trust is not a significant factor to have a higher happiness score, but we can see that for the countries which freedom is a significant factor, and the happiness score is more than 7, trust plays a significant role.

```
scatter3D(Happiness$Family, Happiness$Economy, Happiness$Happiness.Score, phi = 0, bty = "g",
          pch = 20, cex = 2, ticktype = "detailed",
          main = "Happiness data", xlab = "Trust",
          ylab ="Economy", zlab = "Happiness.Score")
```



With an increase in the economy score and the happiness score, trust remains constant. This is the trend for happiness scores below 5. After this point, we can see that the impact of trust on happiness score increases gradually.

Prediction

In this section, we will implement several machine learning algorithms to predict happiness score. First, we should split our dataset into training and test set. Our dependent variable is happiness score, and the independent variables are family, economy, life expectancy, trust, freedom, generosity, and dystopia residual.

```
# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)
set.seed(123)
dataset <- Happiness[4:11]
split = sample.split(dataset$Happiness.Score, SplitRatio = 0.8)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)
```

Multiple Linear Regression

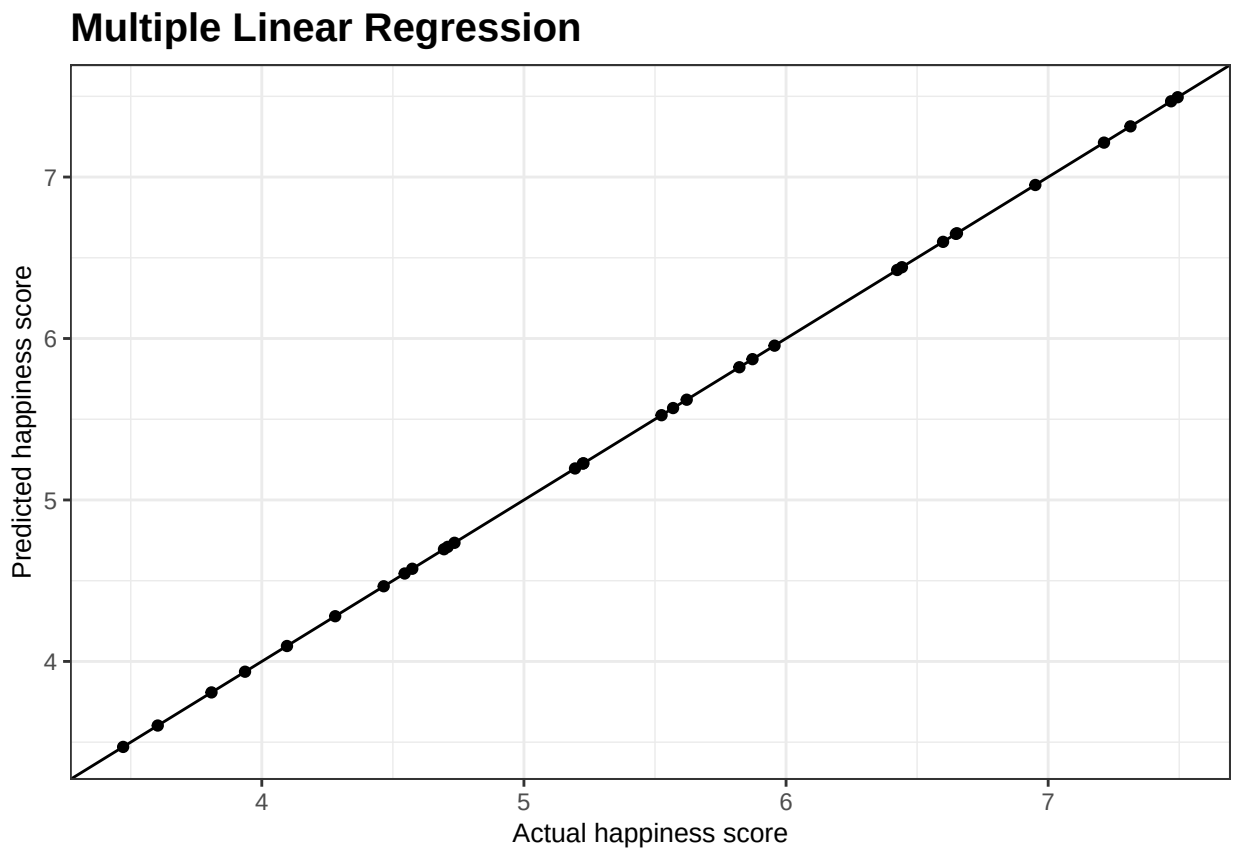
```
# Fitting Multiple Linear Regression to the Training set
regressor_lm = lm(formula = Happiness.Score ~ .,
                  data = training_set)
```

```r
summary(regressor_lm)
```

```
##
## Call:
## lm(formula = Happiness.Score ~ ., data = training_set)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.907e-04 -2.008e-04 -1.600e-07  2.510e-04  4.855e-04
##
## Coefficients:
##                     Estimate Std. Error   t value Pr(>|t|)
## (Intercept)       1.701e-04  1.509e-04     1.127    0.262
## Economy           1.000e+00  1.300e-04  7690.839   <2e-16 ***
## Family            9.999e-01  1.253e-04  7981.804   <2e-16 ***
## Life.Expectancy   9.997e-01  2.122e-04  4711.655   <2e-16 ***
## Freedom           9.999e-01  2.245e-04  4453.253   <2e-16 ***
## Generosity        1.000e+00  2.310e-04  4330.040   <2e-16 ***
## Trust             9.997e-01  3.335e-04  2997.191   <2e-16 ***
## Dystopia.Residual 1.000e+00  5.452e-05 18343.021   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0002848 on 116 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.689e+08 on 7 and 116 DF,  p-value: < 2.2e-16
```

The summary shows that all independent variables have a significant impact, and adjusted R squared is 1! As we discussed, it is clear that there is a linear correlation between dependent and independent variables. Again, I should mention that the sum of the independent variables is equal to the dependent variable which is the happiness score. This is the justification for having an adjusted R squared equals to 1. As a result, I guess Multiple Linear Regression will predict happiness scores with 100 % accuracy!

```r
####### Predicting the Test set results
y_pred_lm = predict(regressor_lm, newdata = test_set)

Pred_Actual_lm <- as.data.frame(cbind(Prediction = y_pred_lm, Actual = test_set$Happiness.Score))

gg.lm <- ggplot(Pred_Actual_lm, aes(Actual, Prediction )) +
  geom_point() + theme_bw() + geom_abline() +
  labs(title = "Multiple Linear Regression", x = "Actual happiness score",
       y = "Predicted happiness score") +
  theme(plot.title = element_text(family = "Helvetica", face = "bold", size = (15)),
        axis.title = element_text(family = "Helvetica", size = (10)))
gg.lm
```

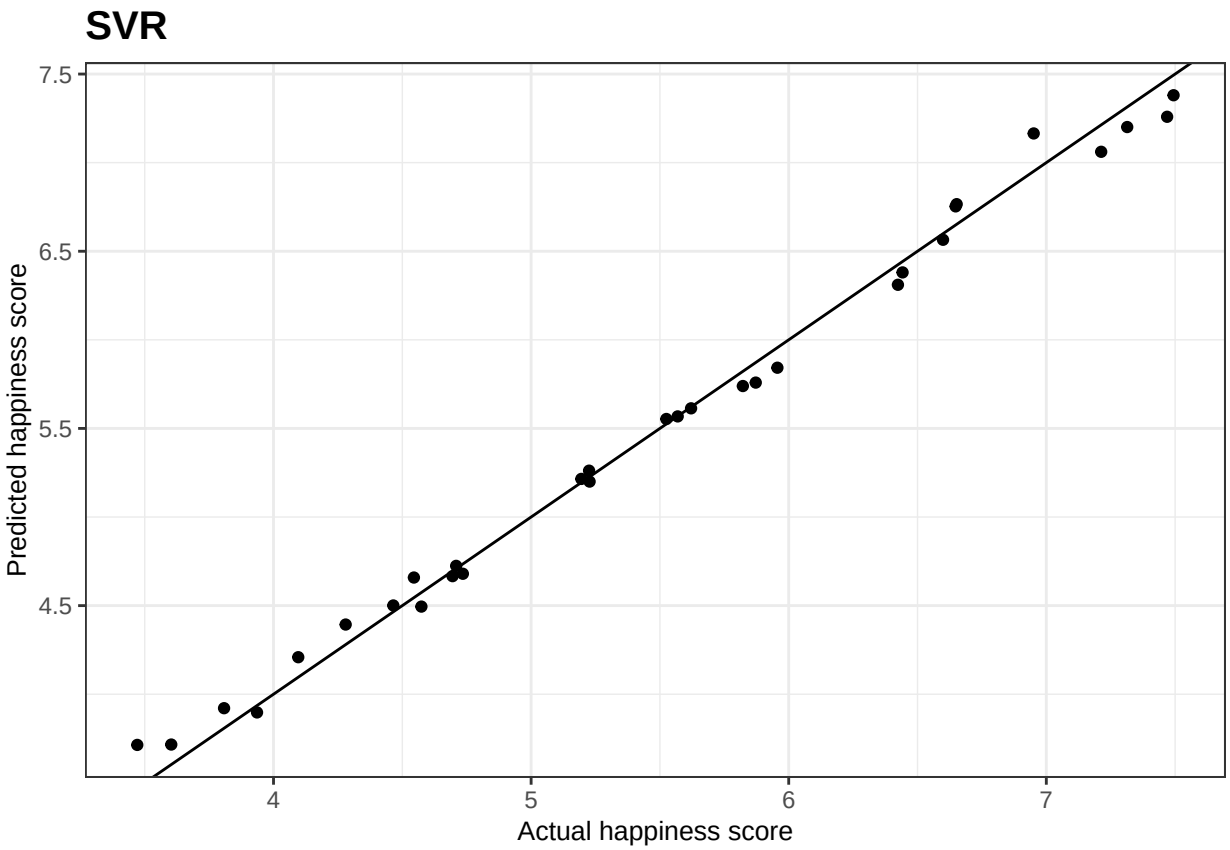


As we expected, actual versus predicted plot shows the accuracy of our model.

SVR

```
# Fitting SVR to the dataset
library(e1071)
regressor_svr = svm(formula = Happiness.Score ~ .,
                    data = dataset,
                    type = 'eps-regression',
                    kernel = 'radial')
```

```
# Predicting a new result
y_pred_svr = predict(regressor_svr,  newdata = test_set)

Pred_Actual_svr <- as.data.frame(cbind(Prediction = y_pred_svr, Actual = test_set$Happiness.Score))


Pred_Actual_lm.versus.svr <- cbind(Prediction.lm = y_pred_lm, Prediction.svr = y_pred_svr, Actual = tes


gg.svr <- ggplot(Pred_Actual_svr, aes(Actual, Prediction )) +
  geom_point() + theme_bw() + geom_abline() +
  labs(title = "SVR", x = "Actual happiness score",
       y = "Predicted happiness score") +
  theme(plot.title = element_text(family = "Helvetica", face = "bold", size = (15)),
        axis.title = element_text(family = "Helvetica", size = (10)))
gg.svr
```

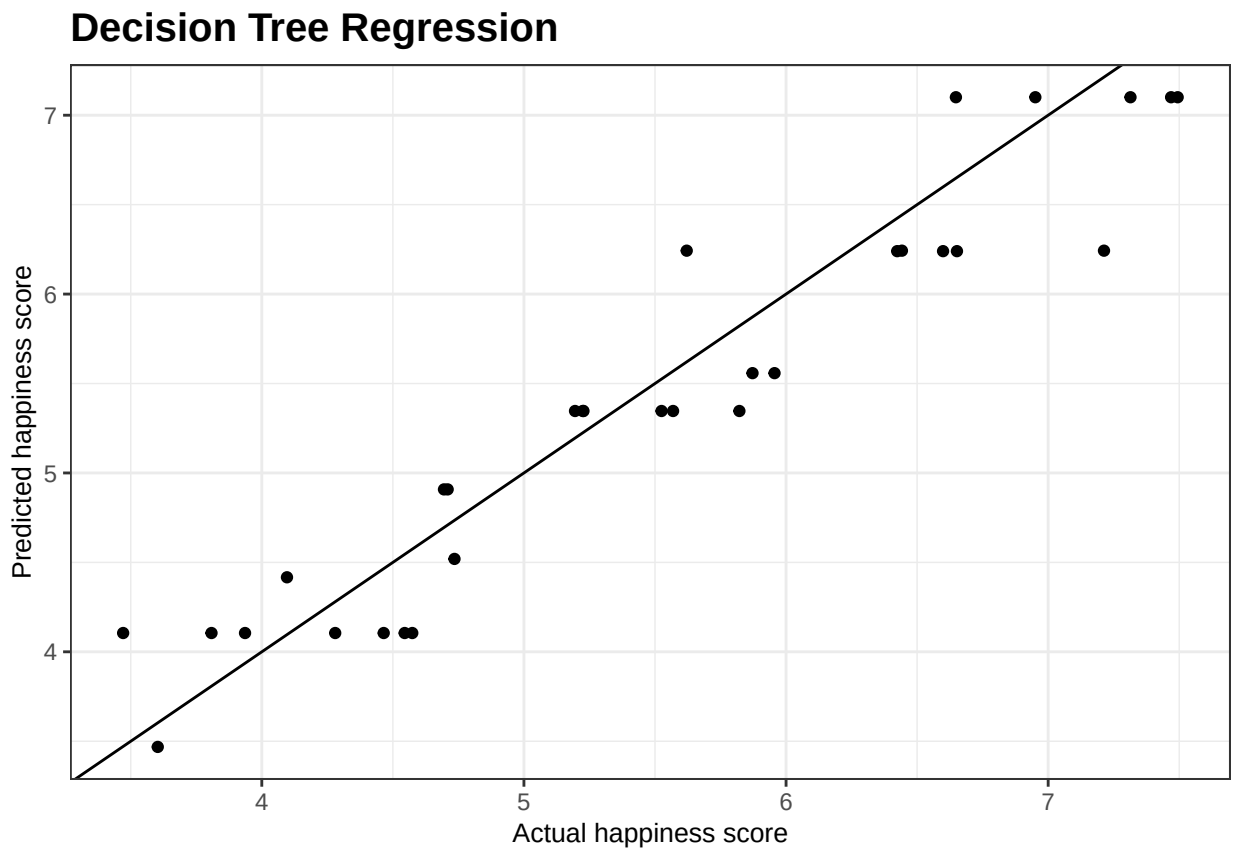


Support Vector Regression predicted happiness scores with pretty high accuracy.

Decision Tree Regression

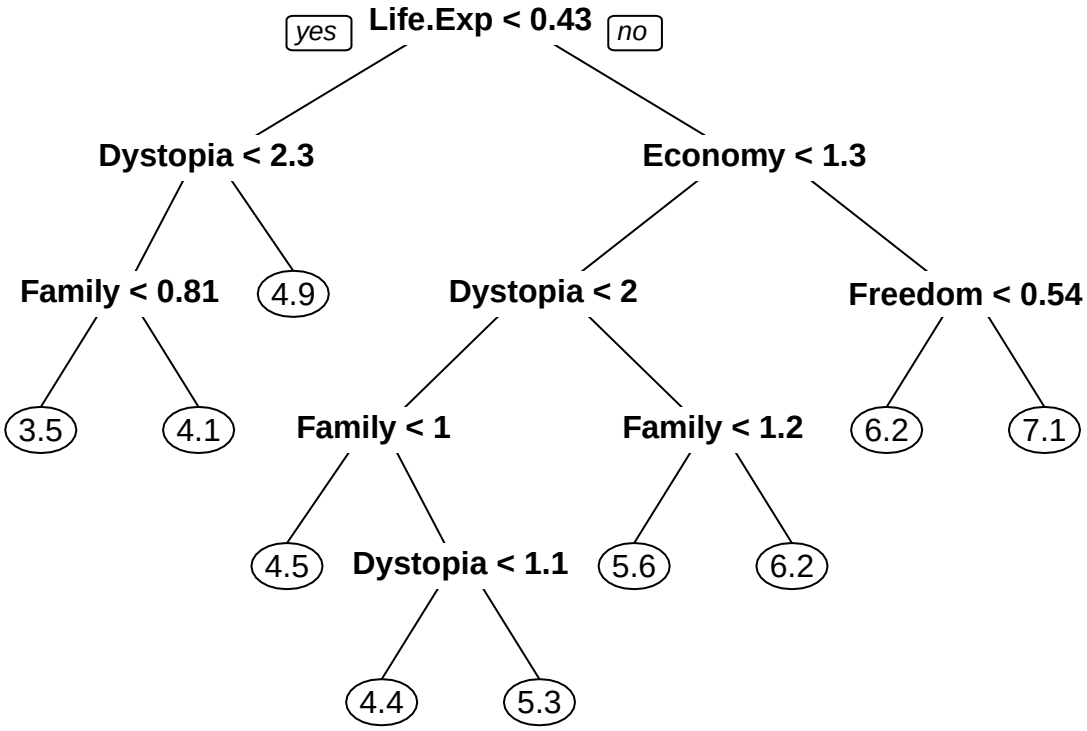
```
# Fitting Decision Tree Regression to the dataset
library(rpart)
regressor_dt = rpart(formula = Happiness.Score ~ .,
                     data = dataset,
                     control = rpart.control(minsplit = 10))
```

```
# Predicting a new result with Decision Tree Regression
y_pred_dt = predict(regressor_dt, newdata = test_set)

Pred_Actual_dt <- as.data.frame(cbind(Prediction = y_pred_dt, Actual = test_set$Happiness.Score))


gg.dt <- ggplot(Pred_Actual_dt, aes(Actual, Prediction )) +
  geom_point() + theme_bw() + geom_abline() +
  labs(title = "Decision Tree Regression", x = "Actual happiness score",
       y = "Predicted happiness score") +
  theme(plot.title = element_text(family = "Helvetica", face = "bold", size = (15)),
        axis.title = element_text(family = "Helvetica", size = (10)))
gg.dt
```



It seems that Decision Tree Regression is not an excellent choice for this dataset. Let's see the tree.

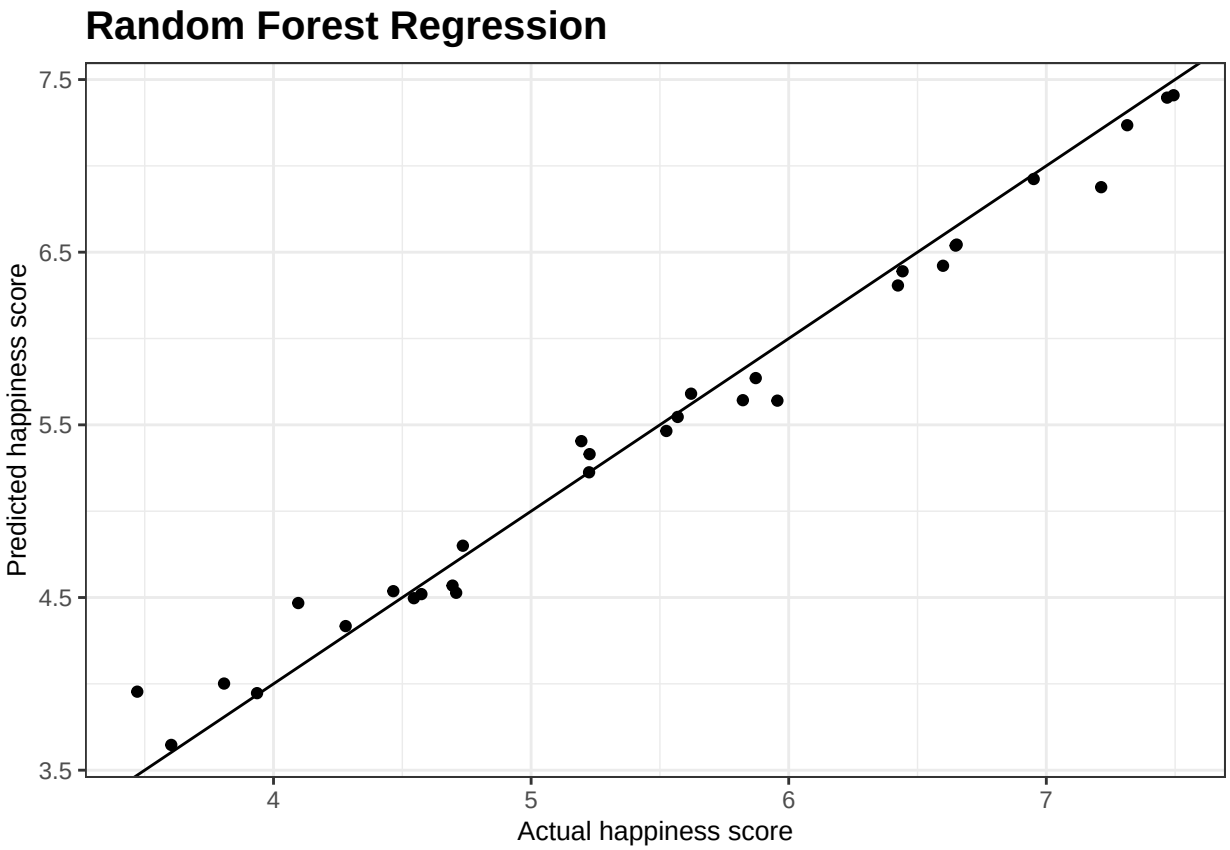
```
# Plotting the tree
library(rpart.plot)
prp(regressor_dt)
```



Random Forest Regression

```
# Fitting Random Forest Regression to the dataset
library(randomForest)
set.seed(1234)
regressor_rf = randomForest(x = dataset[-1],
                            y = dataset$Happiness.Score,
                            ntree = 500)
```

```
# Predicting a new result with Random Forest Regression
y_pred_rf = predict(regressor_rf, newdata = test_set)

Pred_Actual_rf <- as.data.frame(cbind(Prediction = y_pred_rf, Actual = test_set$Happiness.Score))


gg.rf <- ggplot(Pred_Actual_rf, aes(Actual, Prediction )) +
  geom_point() + theme_bw() + geom_abline() +
  labs(title = "Random Forest Regression", x = "Actual happiness score",
       y = "Predicted happiness score") +
  theme(plot.title = element_text(family = "Helvetica", face = "bold", size = (15)),
        axis.title = element_text(family = "Helvetica", size = (10)))
gg.rf
```

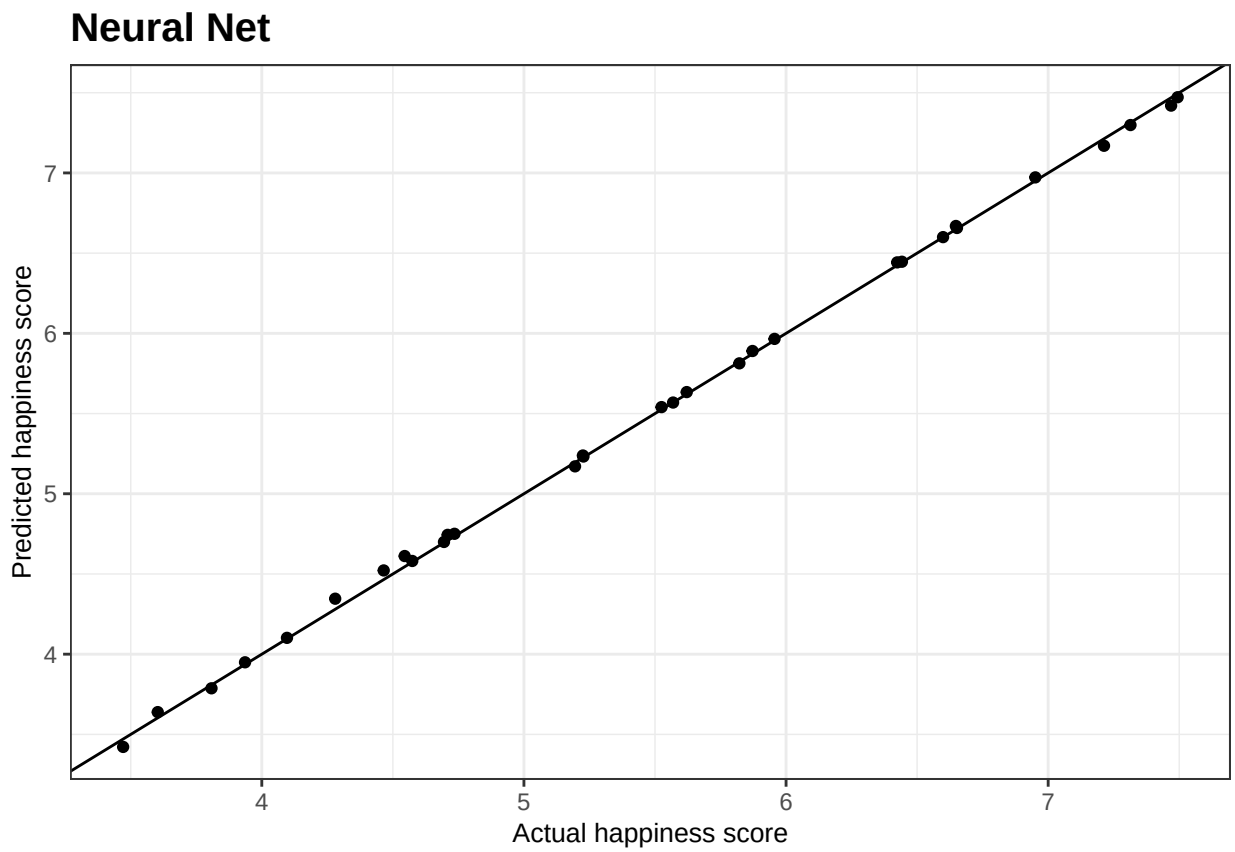


Randon Forest regression is not as good as SVR regarding predicted happiness scores but did a better job than Decision Tree.

Neural Net

```
# Fitting Neural Net to the training set
library(neuralnet)

nn <- neuralnet(Happiness.Score ~ Economy + Family + Life.Expectancy + Freedom + Generosity + Trust + Dy
                data=training_set,hidden=10,linear.output=TRUE)
plot(nn)
```

```
predicted.nn.values <- compute(nn,test_set[,2:8])

Pred_Actual_nn <- as.data.frame(cbind(Prediction = predicted.nn.values$net.result, Actual = test_set$Hap

gg.nn <- ggplot(Pred_Actual_nn, aes(Actual, V1 )) +
  geom_point() + theme_bw() + geom_abline() +
  labs(title = "Neural Net", x = "Actual happiness score",
       y = "Predicted happiness score") +
  theme(plot.title = element_text(family = "Helvetica", face = "bold", size = (15)),
      axis.title = element_text(family = "Helvetica", size = (10)))
gg.nn
```



Neural Net is the best predictor after Multiple Linear Regression. In fact, this model predicted happiness scores with the accuracy close to 100 %. Let's calculate the mean squared error for Multiple Linear Regression and Neural Net model.

```
MSE.lm <- sum((test_set$Happiness.Score - y_pred_lm)^2)/nrow(test_set)
MSE.nn <- sum((test_set$Happiness.Score - predicted.nn.values$net.result)^2)/nrow(test_set)

print(paste("Mean Squared Error (Multiple Linear Regression):", MSE.lm))
```

```
## [1] "Mean Squared Error (Multiple Linear Regression): 9.12868493257418e-08"
```

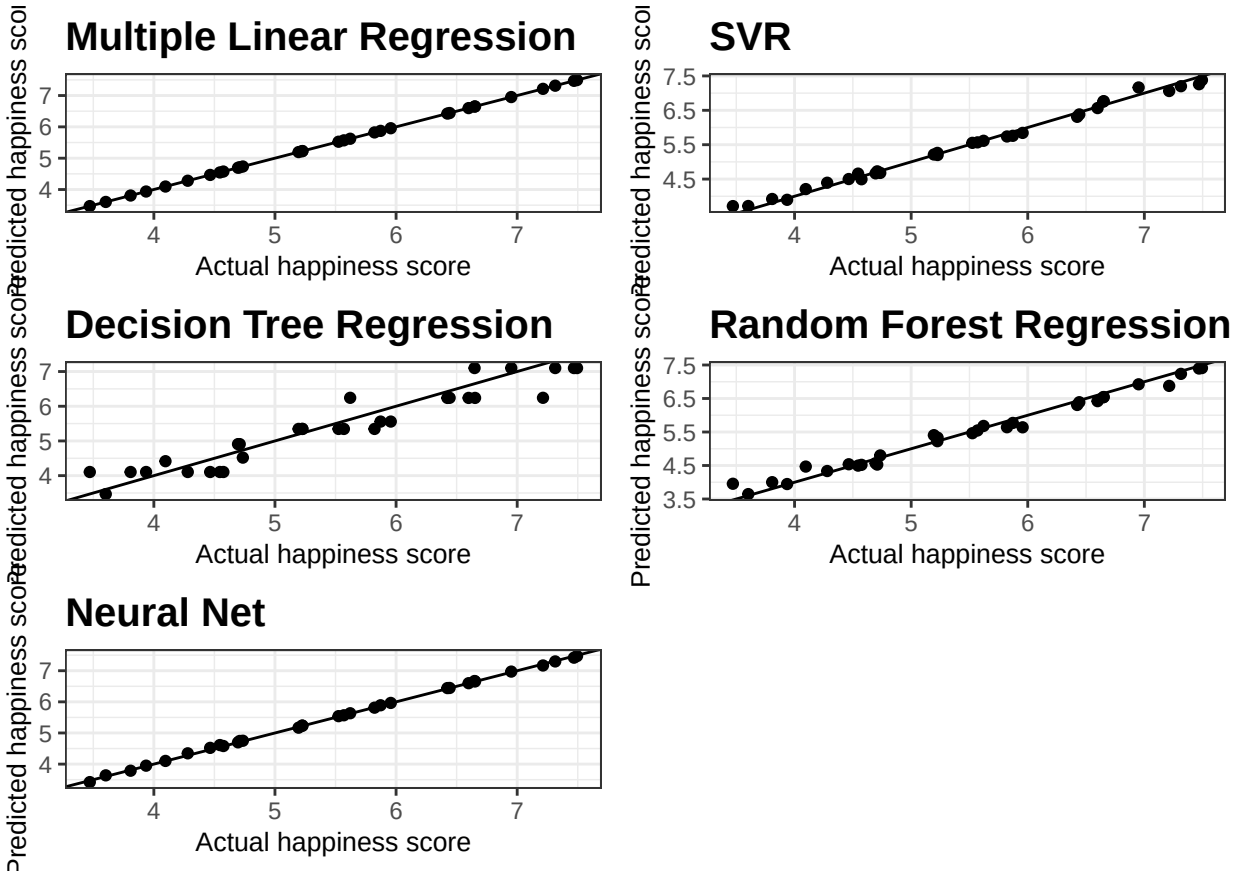
```
print(paste("Mean Squared Error (Neural Net):", MSE.nn))
```

```
## [1] "Mean Squared Error (Neural Net): 0.000837343644496843"
```

As we expected, mean squared error for Multiple Linear Regression is smaller that Neural Net.

Real versus predicted for different machine learning algorithms
Let's see one more time the result of our predictors to see their accuracy visually.

```
ggarrange(gg.lm, gg.svr, gg.dt, gg.rf, gg.nn, ncol = 2, nrow = 3)
```



Multiple Linear Regression and neural net did the best job and predicted approximately the same. SVR and Random Forest stood in the second place regarding accuracy in prediction. And finally, Decision Tree was the worst algorithm to predict happiness scores.