Final Project (Group 2)

Group 2

2024-06-02

«««< HEAD - Research Question/Hypothesis: What variable in the world happiness report (family, health, trust, generosity, and economics) has the greatest effect on a nation's happiness score? =======

- Research Question/Hypothesis: What variable in the world happiness report (family, health, trust, generosity, and economics) has the greatest effect on a nation's happiness score? »»»> 81e6af9ac23978bbafb85c9fa12c41d73c572ee5
- Hypothesis: Economics plays the largest role in a nation's happiness score.

```
library(readxl)
library(dplyr)
library(ggplot2)
library(tidyr)

data <- read_excel("2019.xls")

colnames(data)
```

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

```
library(readxl)

data <- read_excel("2019.xls")

print(colnames(data))
```

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

```r
data <- data %>%
  rename(
    Economy = `GDP per capita`,
    Social = 'Social support',
    Health = `Healthy life expectancy`,
    Freedom = `Freedom to make life choices`,
    Corruption = 'Perceptions of corruption',
    Happiness_Score = `Score`
  )
print(colnames(data))
```

```
## [1] "Overall rank"      "Country or region" "Happiness_Score"
## [4] "Economy"           "Social"            "Health"
## [7] "Freedom"           "Generosity"        "Corruption"
```

```r
  head(
    select(data, Economy, Social, Health, Freedom, Corruption, Happiness_Score)
  )
```

Economy	Social	Health	Freedom	Corruption	Happiness_Score
1.340	1.587	0.986	0.596	0.393	7.769
1.383	1.573	0.996	0.592	0.410	7.600
1.488	1.582	1.028	0.603	0.341	7.554
1.380	1.624	1.026	0.591	0.118	7.494
1.396	1.522	0.999	0.557	0.298	7.488
1.452	1.526	1.052	0.572	0.343	7.480

[Module 2: Junhyung Kim, Jiho Lee]

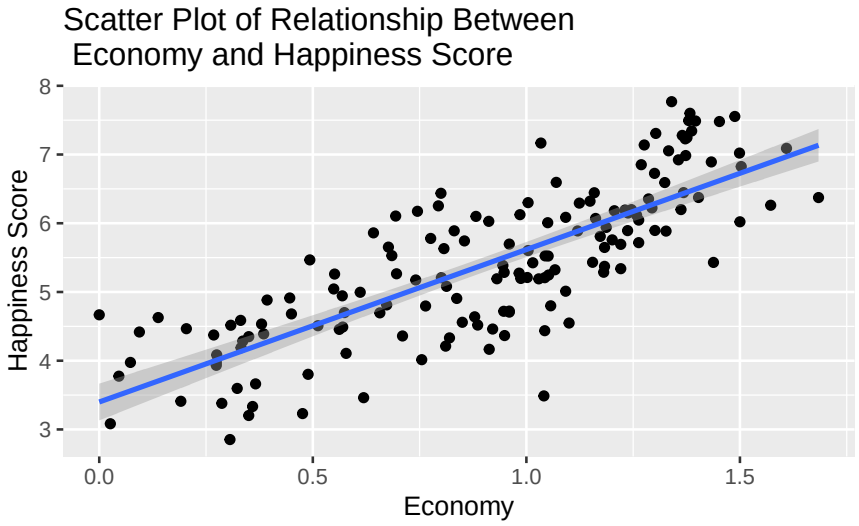
*Scatter Plot

```r
qplot(x = Economy, y= Happiness_Score, data = data,
 geom = c("point", "smooth"), method = "lm") +
 labs(title =
 "Scatter Plot of Relationship Between
 Economy and Happiness Score",
 x = "Economy", y = "Happiness Score")
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning in geom_point(method = "lm"): Ignoring unknown parameters: `method`
```

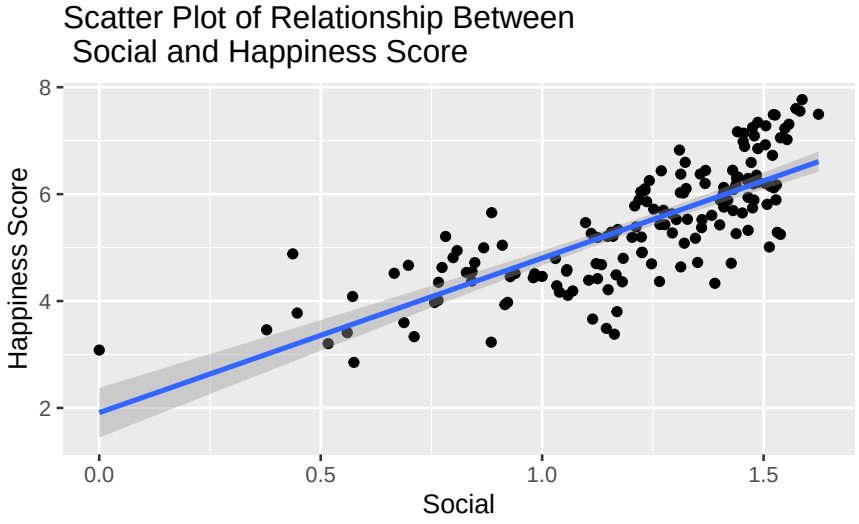
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
qplot(x= Social,y=Happiness_Score,data=data,
geom=c("point","smooth"),method="lm")+
labs(title =
"Scatter Plot of Relationship Between
Social and Happiness Score",
x="Social",y="Happiness Score")
```

```
## Warning in geom_point(method = "lm"): Ignoring unknown parameters: `method`
```

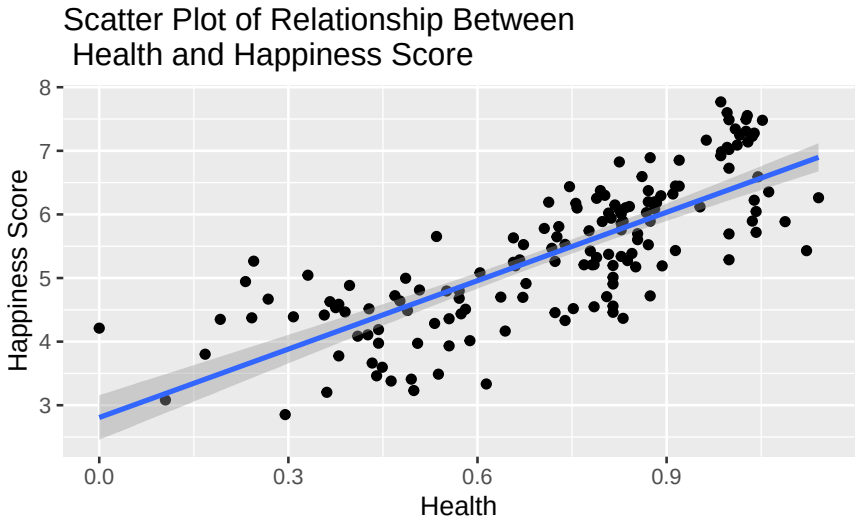
```
## `geom_smooth()` using formula = 'y ~ x'
```



```
qplot(x= Health,y=Happiness_Score,data=data,
 geom=c("point","smooth"),method="lm")+
 labs(title =
 "Scatter Plot of Relationship Between
 Health and Happiness Score",
 x="Health",y="Happiness Score")
```

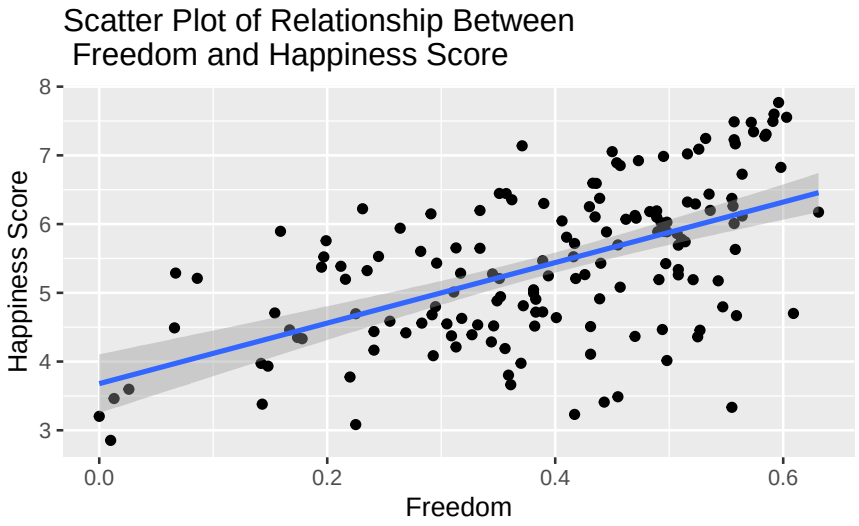
```
## Warning in geom_point(method = "lm"): Ignoring unknown parameters: ‘method‘
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



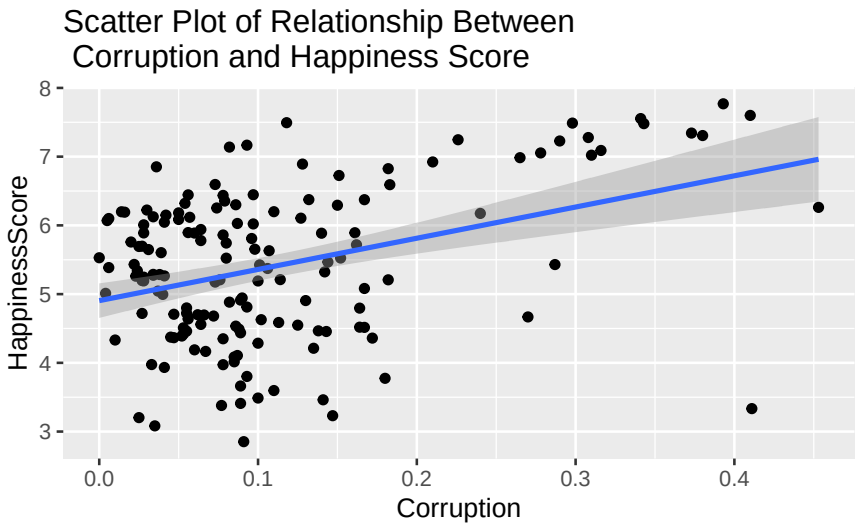

```
qplot(x= Freedom,y=Happiness_Score,data=data,
 geom=c("point","smooth"),method="lm")+
 labs(title =
 "Scatter Plot of Relationship Between
 Freedom and Happiness Score",
 x="Freedom",y="Happiness Score")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



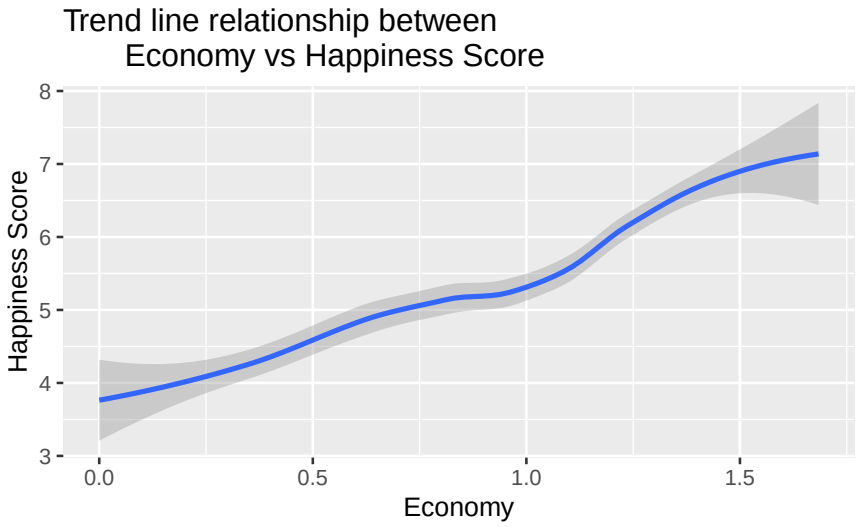

```
qplot(x= Corruption,y=Happiness_Score,data=data,
geom=c("point","smooth"),method="lm")+
labs(title ="Scatter Plot of Relationship Between
Corruption and Happiness Score",
x="Corruption",y="HappinessScore")
```

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



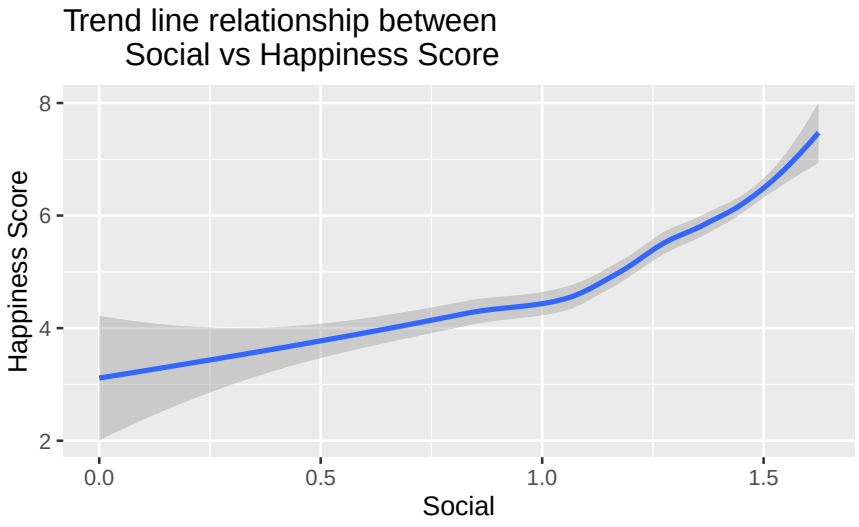

```
data %>%
  ggplot() +
  geom_smooth(mapping = aes(x = Economy, y = Happiness_Score)) +
  labs(x = "Economy", y = "Happiness Score",
       title="Trend line relationship between
       Economy vs Happiness Score")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
data %>%
  ggplot() +
  geom_smooth(mapping = aes(x = Social, y = Happiness_Score)) +
  labs(x = "Social", y = "Happiness Score",
       title="Trend line relationship between
       Social vs Happiness Score")
```

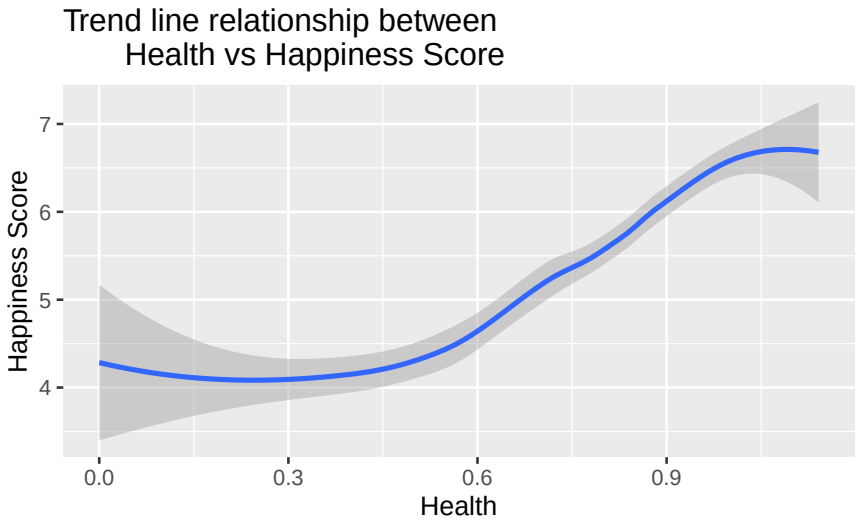
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
data %>%
  ggplot() +
  geom_smooth(mapping = aes(x = Health, y = Happiness_Score)) +
```

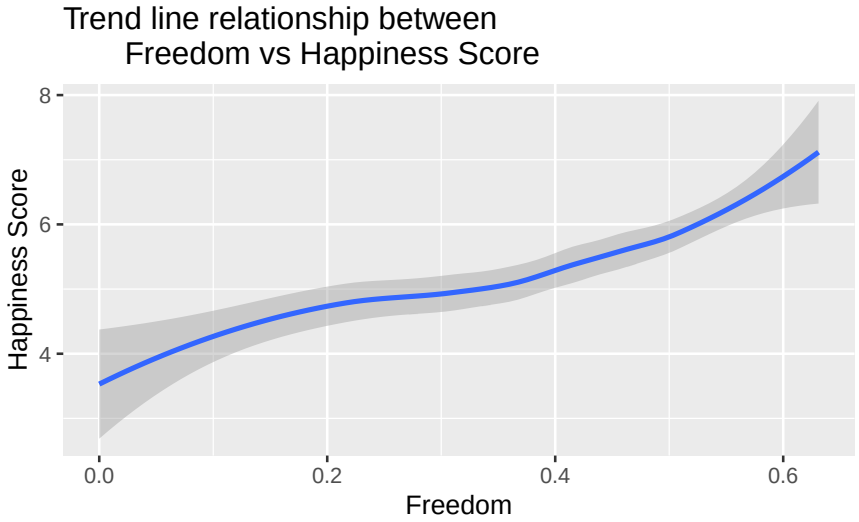
```
  labs(x = "Health", y = "Happiness Score",
       title="Trend line relationship between
       Health vs Happiness Score")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



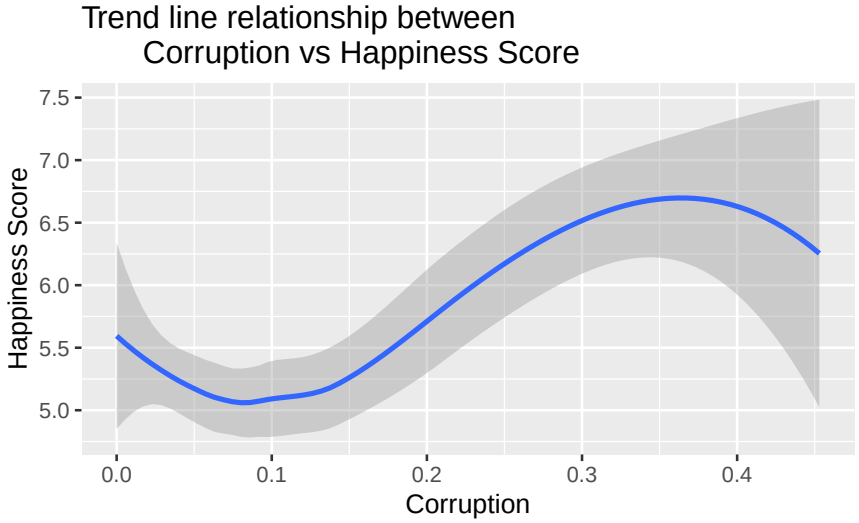

```
data %>%
  ggplot() +
  geom_smooth(mapping = aes(x = Freedom, y = Happiness_Score)) +
  labs(x = "Freedom", y = "Happiness Score",
       title="Trend line relationship between
       Freedom vs Happiness Score")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
data %>%
  ggplot() +
  geom_smooth(mapping = aes(x = Corruption, y = Happiness_Score)) +
  labs(x = "Corruption", y = "Happiness Score",
       title="Trend line relationship between
       Corruption vs Happiness Score")
```

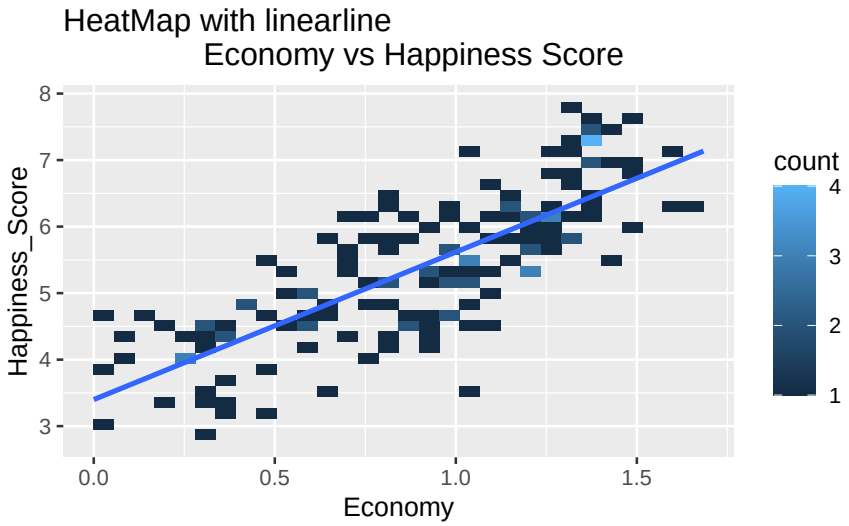
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



*HeatMap

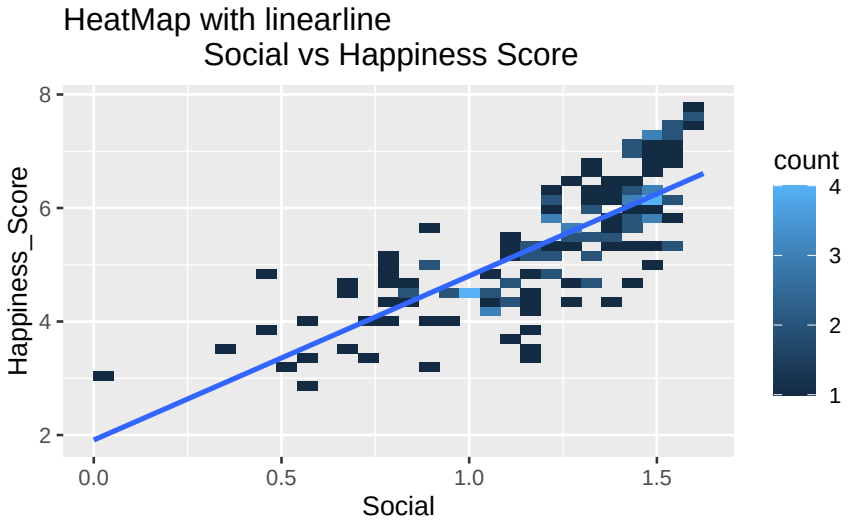
```
data %>%
ggplot(aes(x = Economy, y = Happiness_Score)) +
  geom_bin2d(bins = 30) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "HeatMap with linearline
                Economy vs Happiness Score",
           x = "Economy", y = "Happiness_Score")
```

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




```
data %>%
ggplot(aes(x = Social, y = Happiness_Score)) +
  geom_bin2d(bins = 30) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "HeatMap with linearline
                Social vs Happiness Score",
          x = "Social", y = "Happiness_Score")
```

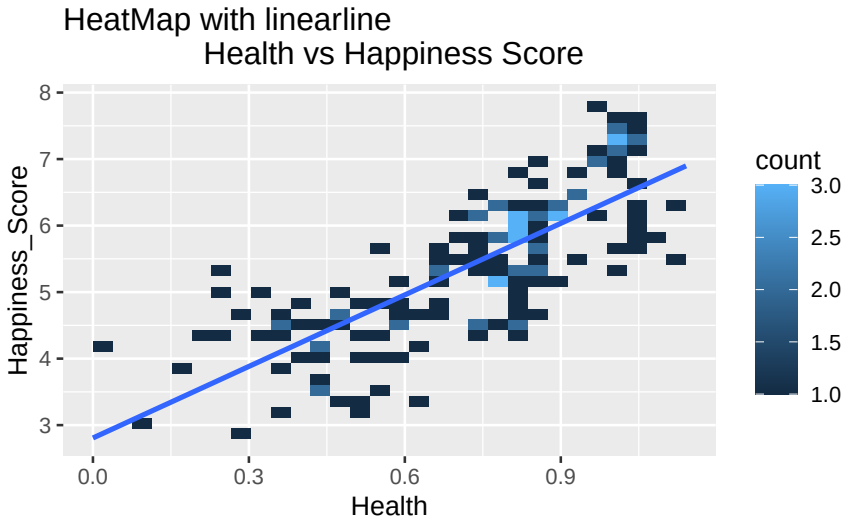
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
data %>%
ggplot(aes(x = Health, y = Happiness_Score)) +
  geom_bin2d(bins = 30) +
  geom_smooth(method = "lm", se = FALSE) +
```

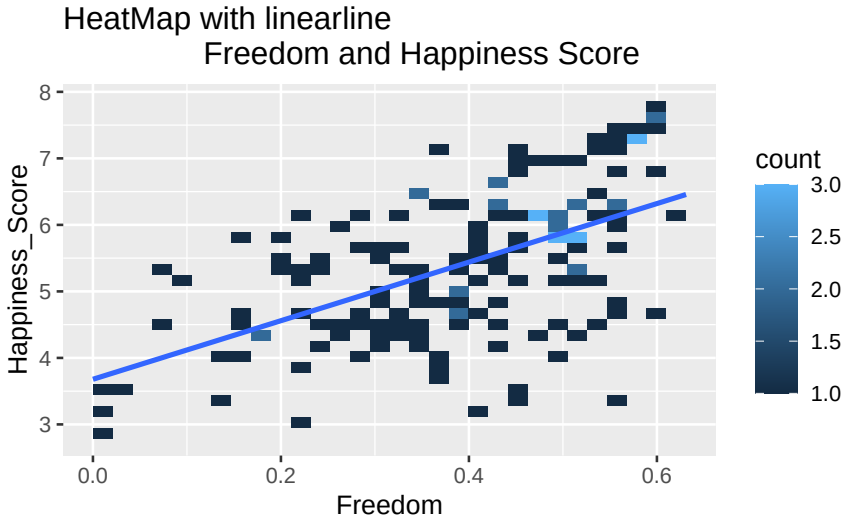
```
  labs(title = "HeatMap with linearline
                Health vs Happiness Score",
           x = "Health", y = "Happiness_Score")
```

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



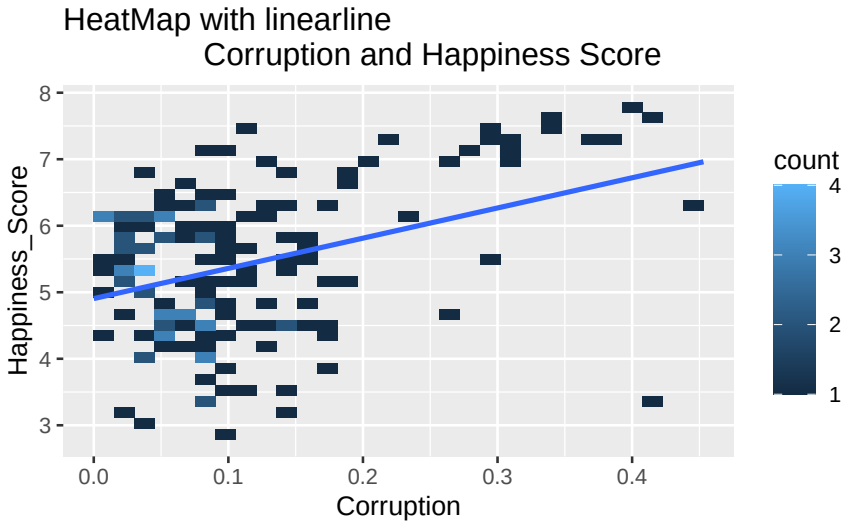

```
data %>%
ggplot(aes(x = Freedom, y = Happiness_Score)) +
  geom_bin2d(bins = 30) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "HeatMap with linearline
                Freedom and Happiness Score",
           x = "Freedom", y = "Happiness_Score")
```

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



```
data %>%
ggplot(aes(x = Corruption, y = Happiness_Score)) +
  geom_bin2d(bins = 30) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "HeatMap with linearline
                Corruption and Happiness Score",
           x = "Corruption", y = "Happiness_Score")
```

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



======= [Module 4: Eugene Kim, Harold Lee - Explanatory Data Analysis]

```
str(data, vec.len = 2)
```

```
## tibble [156 x 9] (S3: tbl_df/tbl/data.frame)
##  $ Overall rank     : num [1:156] 1 2 3 4 5 ...
##  $ Country or region: chr [1:156] "Finland" "Denmark" ...
##  $ Happiness_Score  : num [1:156] 7.77 7.6 ...
##  $ Economy          : num [1:156] 1.34 1.38 ...
##  $ Social           : num [1:156] 1.59 1.57 ...
##  $ Health           : num [1:156] 0.986 0.996 ...
##  $ Freedom          : num [1:156] 0.596 0.592 0.603 0.591 0.557 ...
##  $ Generosity       : num [1:156] 0.153 0.252 0.271 0.354 0.322 ...
##  $ Corruption       : num [1:156] 0.393 0.41 0.341 0.118 0.298 ...
```

```
head(
    select(data, Economy, Social, Health, Freedom, Corruption,
           Happiness_Score)
  )
```

Economy	Social	Health	Freedom	Corruption	Happiness_Score
1.340	1.587	0.986	0.596	0.393	7.769
1.383	1.573	0.996	0.592	0.410	7.600
1.488	1.582	1.028	0.603	0.341	7.554
1.380	1.624	1.026	0.591	0.118	7.494
1.396	1.522	0.999	0.557	0.298	7.488
1.452	1.526	1.052	0.572	0.343	7.480

```
tail(select(data, Economy, Social, Health, Freedom, Corruption,
            Happiness_Score)
  )
```

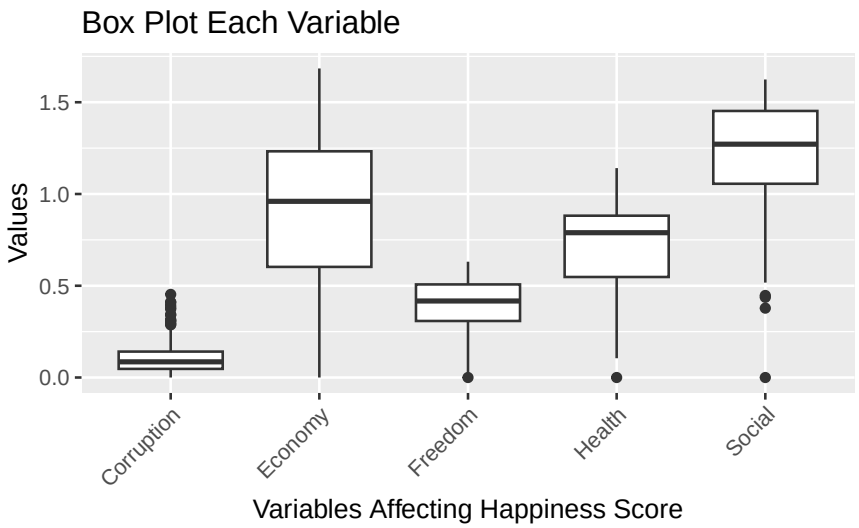
Economy	Social	Health	Freedom	Corruption	Happiness_Score
0.287	1.163	0.463	0.143	0.077	3.380
0.359	0.711	0.614	0.555	0.411	3.334
0.476	0.885	0.499	0.417	0.147	3.231
0.350	0.517	0.361	0.000	0.025	3.203
0.026	0.000	0.105	0.225	0.035	3.083
0.306	0.575	0.295	0.010	0.091	2.853

*Summary statistics

*Box Plot

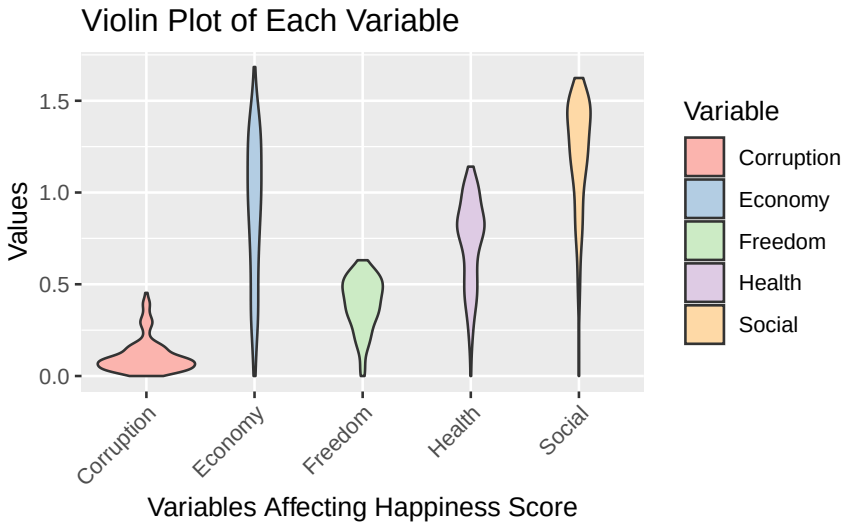
```
data_long <- data %>%
  gather(key = "Variable", value = "Score", Economy, Social, Health,
         Freedom, Corruption)

ggplot(data_long, aes(x = Variable, y = Score)) +
  geom_boxplot(width = 0.7) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Box Plot Each Variable",
       x = "Variables Affecting Happiness Score", y = "Values")
```



*Violin Plot

```
ggplot(data_long, aes(x = Variable, y = Score, fill = Variable)) +
  geom_violin(trim = TRUE) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Violin Plot of Each Variable",
       x = "Variables Affecting Happiness Score", y = "Values") +
  scale_fill_brewer(palette = "Pastel1")
```



*Summary

```
data %>%
  summarize(
    mean= mean(Economy),
```

```
    median = median(Economy),
    sd = sd(Economy),
    iqr = IQR(Economy),
    min = min(Economy),
    max = max(Economy)
 )
```

mean	median	sd	iqr	min	max
0.9051474	0.96	0.3983895	0.62975	0	1.684

```
data %>%
  summarize(
    mean= mean(Social),
    median = median(Social),
    sd = sd(Social),
    iqr = IQR(Social),
    min = min(Social),
    max = max(Social)
 )
```

mean	median	sd	iqr	min	max
1.208814	1.2715	0.2991914	0.39675	0	1.624

```
data %>%
  summarize(
    mean= mean(Health),
    median = median(Health),
    sd = sd(Health),
    iqr = IQR(Health),
    min = min(Health),
    max = max(Health)
 )
```

mean	median	sd	iqr	min	max
0.7252436	0.789	0.242124	0.334	0	1.141

```
data %>%
  summarize(
    mean= mean(Freedom),
    median = median(Freedom),
    sd = sd(Freedom),
    iqr = IQR(Freedom),
```

```
    min = min(Freedom),
    max = max(Freedom)
 )
```

mean	median	sd	iqr	min	max
0.3925705	0.417	0.1432895	0.19925	0	0.631

```
data %>%
  summarize(
    mean= mean(Corruption),
    median = median(Corruption),
    sd = sd(Corruption),
    iqr = IQR(Corruption),
    min = min(Corruption),
    max = max(Corruption)
 )
```

mean	median	sd	iqr	min	max
0.1106026	0.0855	0.0945378	0.09425	0	0.453

```
data_E <- data %>%
  summarize(
    mean= mean(Economy),
    median = median(Economy),
    sd = sd(Economy),
    iqr = IQR(Economy),
    min = min(Economy),
    max = max(Economy)
 )
```

```
data_S <- data %>%
  summarize(
    mean= mean(Social),
    median = median(Social),
    sd = sd(Social),
    iqr = IQR(Social),
    min = min(Social),
    max = max(Social)
 )
```

```
data_H <- data %>%
  summarize(
    mean= mean(Health),
```

```
    median = median(Health),
    sd = sd(Health),
    iqr = IQR(Health),
    min = min(Health),
    max = max(Health)
 )
```

```
data_F <- data %>%
  summarize(
    mean= mean(Freedom),
    median = median(Freedom),
    sd = sd(Freedom),
    iqr = IQR(Freedom),
    min = min(Freedom),
    max = max(Freedom)
 )
```

```
data_C <- data %>%
  summarize(
    mean= mean(Corruption),
    median = median(Corruption),
    sd = sd(Corruption),
    iqr = IQR(Corruption),
    min = min(Corruption),
    max = max(Corruption)
 )
```

```
combined_data <- rbind(data_E, data_S, data_H, data_F, data_C)
```

```
combined_data_rounded <- combined_data %>%
  mutate(across(everything(), ~ round(., 2)))
```

```
row.names(combined_data_rounded) <- c("Economy", "Social",
                                      "Health","Freedom", "Corruption")
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
print(combined_data_rounded)
```

```
## # A tibble: 5 x 6
##    mean median    sd   iqr   min   max
## * <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  0.91   0.96  0.4   0.63     0  1.68
## 2  1.21   1.27  0.3   0.4      0  1.62
## 3  0.73   0.79  0.24  0.33     0  1.14
## 4  0.39   0.42  0.14  0.2      0  0.63
## 5  0.11   0.09  0.09  0.09     0  0.45
```

[Module 5: Chun Jin Park - Modeling]

Linear model using lm

```
model <- lm(Happiness_Score ~ Economy + Social + Health + Freedom + Generosity
            + Corruption, data = data)

summary(model)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Economy + Social + Health + Freedom +
##     Generosity + Corruption, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.75304 -0.35306  0.05703  0.36695  1.19059
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.7952     0.2111   8.505 1.77e-14 ***
## Economy       0.7754     0.2182   3.553 0.000510 ***
## Social        1.1242     0.2369   4.745 4.83e-06 ***
## Health        1.0781     0.3345   3.223 0.001560 **
## Freedom       1.4548     0.3753   3.876 0.000159 ***
## Generosity    0.4898     0.4977   0.984 0.326709
## Corruption    0.9723     0.5424   1.793 0.075053 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5335 on 149 degrees of freedom
## Multiple R-squared:  0.7792, Adjusted R-squared:  0.7703
## F-statistic: 87.62 on 6 and 149 DF,  p-value: < 2.2e-16
```

Tidy to get the model coefficients

```
coefficients <- tidy(model)
print(coefficients)
```

```
## # A tibble: 7 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    1.80      0.211     8.51  1.77e-14
```

```
## 2 Economy        0.775     0.218     3.55  5.10e- 4
## 3 Social         1.12      0.237     4.75  4.83e- 6
## 4 Health         1.08      0.335     3.22  1.56e- 3
## 5 Freedom        1.45      0.375     3.88  1.59e- 4
## 6 Generosity     0.490     0.498     0.984 3.27e- 1
## 7 Corruption     0.972     0.542     1.79  7.51e- 2
```

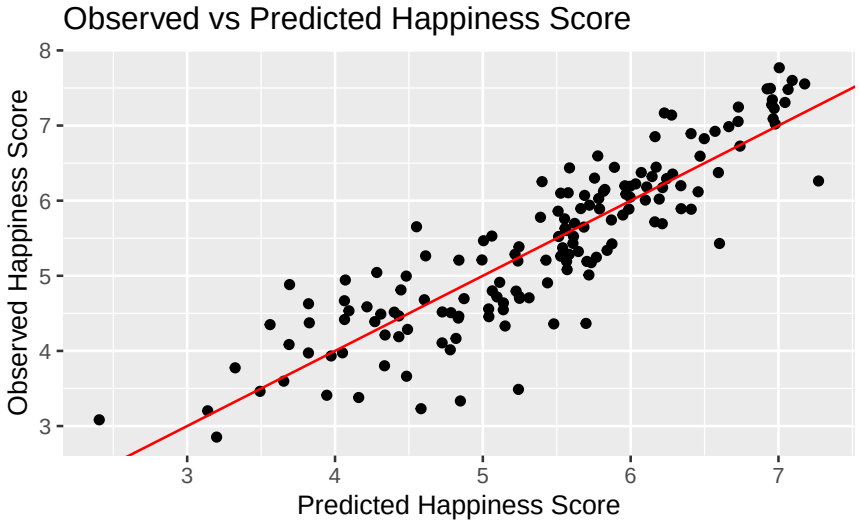
Glance to get the model's performance metrics

```
performance <- glance(model)
print(performance)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.779         0.770 0.534      87.6 2.40e-46     6  -120.  256.  280.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

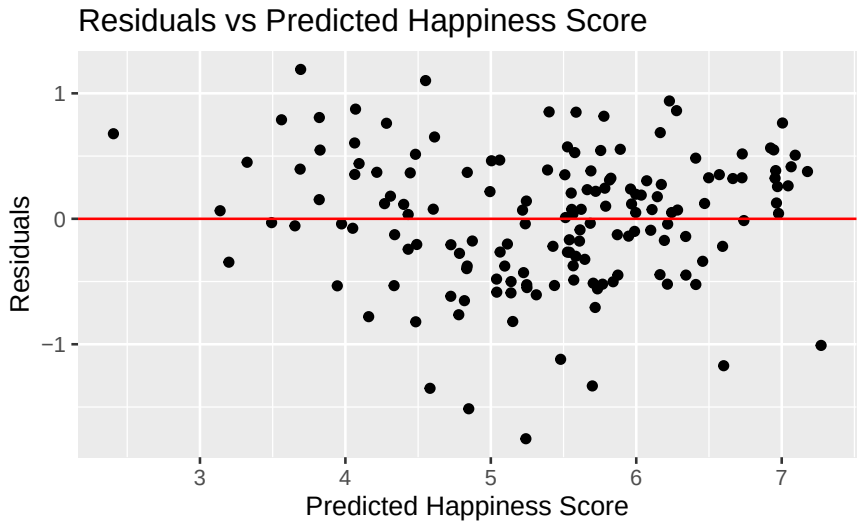
Observed vs Predicted plot

```
ggplot(data, aes(x = predict(model), y = Happiness_Score)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, col = "red") +
  labs(title =
         "Observed vs Predicted Happiness Score",
       x = "Predicted Happiness Score",
       y = "Observed Happiness Score")
```



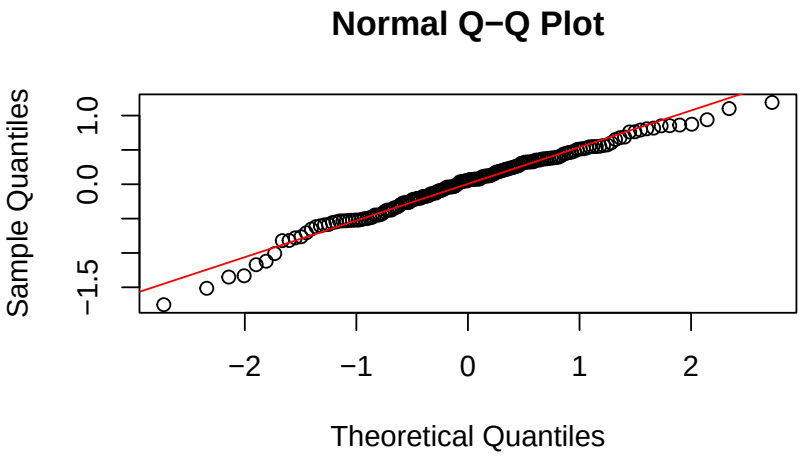
Residuals vs Predicted plot

```
ggplot(data, aes(x = predict(model), y = residuals(model))) +
  geom_point() +
  geom_hline(yintercept = 0, col = "red") +
  labs(title = "Residuals vs Predicted Happiness Score",
       x = "Predicted Happiness Score",
       y = "Residuals")
```



Q-Q plot

```
qqnorm(residuals(model))
qqline(residuals(model), col = "red")
```



[Module 6: Sena Julsdorf and Hyeongseok Sim]

```
ggplot(data, aes(x = Economy, y = Happiness_Score)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, col = "blue") +
  labs(title = "Scatter Plot of Happiness Score based on Economy",
       x = "Economic Score",
       y = "Happiness Score")
```

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


```
library(ggplot2)
library(caTools)
```

```
set.seed(123)
split <- sample.split(data$Happiness_Score, SplitRatio = 0.5)
training_set <- subset(data, split == TRUE)
testing_set <- subset(data, split == FALSE)
```

```
lmmodel <- lm(Happiness_Score ~ Economy, data = testing_set)
summary(lmmodel)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Economy, data = testing_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.30817 -0.49891  0.02652  0.50746  1.27935
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.2913     0.2002   16.44   <2e-16 ***
## Economy       2.3317     0.1974   11.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6552 on 76 degrees of freedom
## Multiple R-squared:  0.6474, Adjusted R-squared:  0.6428
## F-statistic: 139.6 on 1 and 76 DF,  p-value: < 2.2e-16
```

```
predictions <- predict(lmmodel, newdata = testing_set)
```

```
rmse <- sqrt(mean((testing_set$Happiness_Score - predictions)^2))
print(paste("RMSE: ", rmse))
```

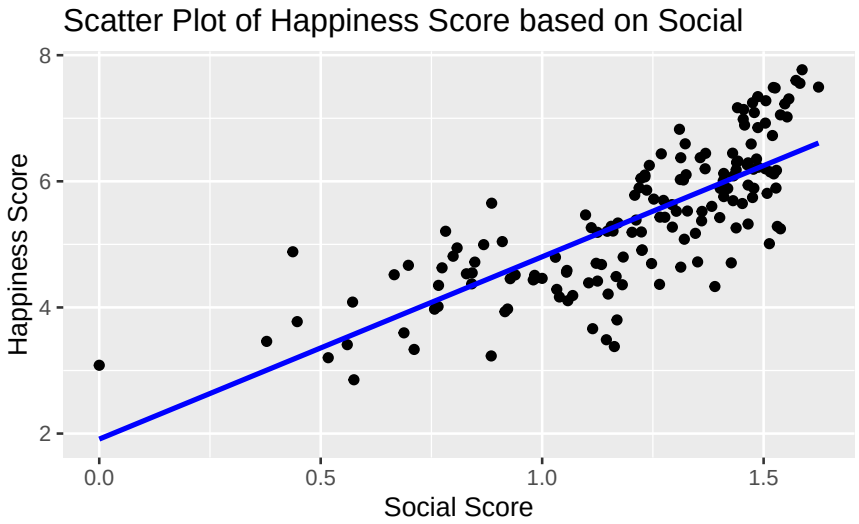
```
## [1] "RMSE:  0.646727299146686"
```

```
r_squared <- summary(lmmodel)$r.squared
print(paste("R-squared: ", r_squared))
```

```
## [1] "R-squared:  0.647436919488881"
```

```
ggplot(data, aes(x = Social, y = Happiness_Score)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, col = "blue") +
  labs(title = "Scatter Plot of Happiness Score based on Social",
       x = "Social Score",
       y = "Happiness Score")
```

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



```r
lmmodel_2 <- lm(Happiness_Score ~ Social, data = testing_set)
summary(lmmodel_2)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Social, data = testing_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.91246 -0.50723  0.04253  0.55227  1.19156
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8034     0.3742   4.819 7.23e-06 ***
## Social        3.0001     0.2974  10.089 1.13e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7214 on 76 degrees of freedom
## Multiple R-squared:  0.5725, Adjusted R-squared:  0.5669
## F-statistic: 101.8 on 1 and 76 DF,  p-value: 1.128e-15
```

```r
predictions_2 <- predict(lmmodel_2, newdata = testing_set)
```

```r
rmse_2 <- sqrt(mean((testing_set$Happiness_Score - predictions_2)^2))
print(paste("RMSE (Social model): ", rmse_2))
```

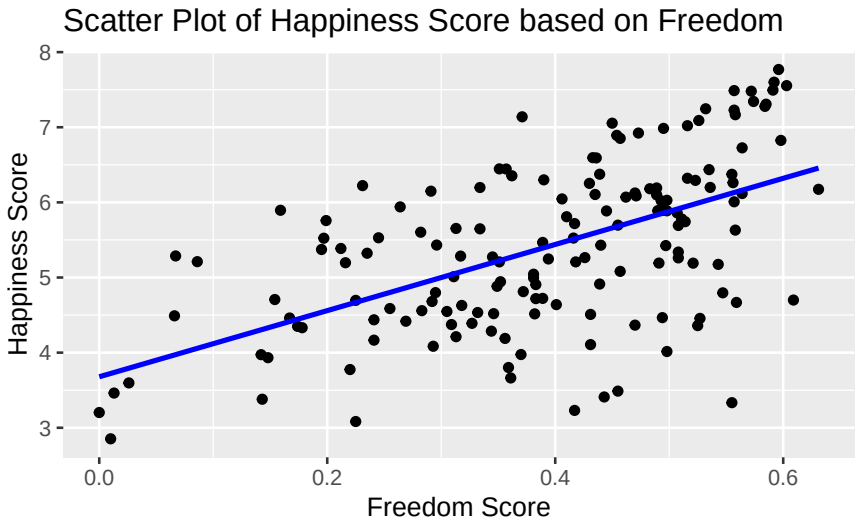
```
## [1] "RMSE (Social model):  0.712139313572519"
```

```r
r_squared_2 <- summary(lmmodel_2)$r.squared
print(paste("R-squared (Social model): ", r_squared_2))
```

```
## [1] "R-squared (Social model):  0.57251156656046"
```

```r
ggplot(data, aes(x = Freedom, y = Happiness_Score)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, col = "blue") +
  labs(title = "Scatter Plot of Happiness Score based on Freedom",
       x = "Freedom Score",
       y = "Happiness Score")
```

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




```
lmmodel_3 <- lm(Happiness_Score ~ Freedom, data = testing_set)
summary(lmmodel_3)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Freedom, data = testing_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8451 -0.6640 -0.0348  0.8907  1.7633
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7556     0.3318  11.317  < 2e-16 ***
## Freedom       4.3667     0.7930   5.507 4.77e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9329 on 76 degrees of freedom
## Multiple R-squared:  0.2852, Adjusted R-squared:  0.2758
## F-statistic: 30.32 on 1 and 76 DF,  p-value: 4.768e-07
```

```
predictions_3 <- predict(lmmodel_3, newdata = testing_set)
```

```
rmse_3 <- sqrt(mean((testing_set$Happiness_Score - predictions_3)^2))
print(paste("RMSE (Freedom model): ", rmse_3))
```

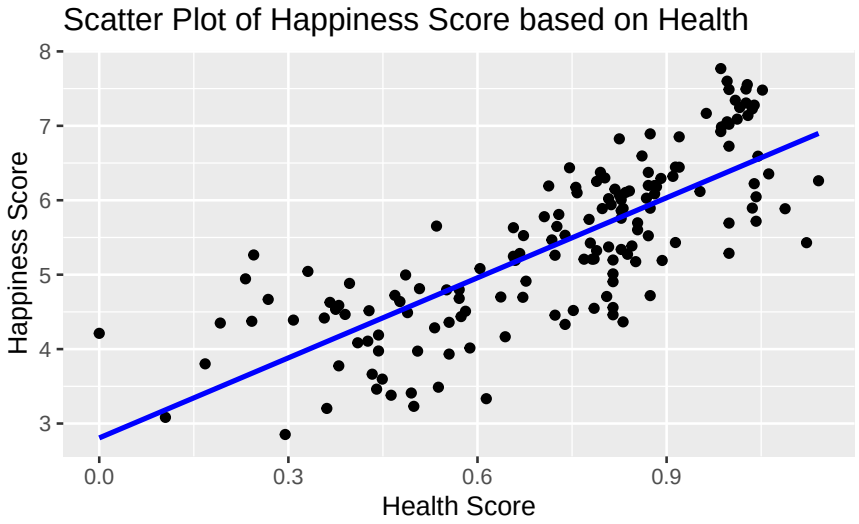
```
## [1] "RMSE (Freedom model):  0.920858315286326"
```

```r
r_squared_3 <- summary(lmmodel_3)$r.squared
print(paste("R-squared (Freedom model): ", r_squared_3))
```

```
## [1] "R-squared (Freedom model):  0.285207357638025"
```

```r
ggplot(data, aes(x = Health, y = Happiness_Score)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, col = "blue") +
  labs(title = "Scatter Plot of Happiness Score based on Health",
       x = "Health Score",
       y = "Happiness Score")
```

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




```r
lmmodel_4 <- lm(Happiness_Score ~ Health, data = testing_set)
summary(lmmodel_4)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Health, data = testing_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.64520 -0.40654  0.06847  0.54871  1.50718
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.7048     0.2762   9.794 4.08e-15 ***
## Health        3.7042     0.3519  10.525  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7039 on 76 degrees of freedom
## Multiple R-squared:  0.5931, Adjusted R-squared:  0.5878
## F-statistic: 110.8 on 1 and 76 DF,  p-value: < 2.2e-16
```

```
predictions_4 <- predict(lmmodel_4, newdata = testing_set)
```

```
rmse_4 <- sqrt(mean((testing_set$Happiness_Score - predictions_4)^2))
print(paste("RMSE (Health model): ", rmse_4))
```

```
## [1] "RMSE (Health model):  0.694771303098578"
```

```
r_squared_4 <- summary(lmmodel_4)$r.squared
print(paste("R-squared (Health model): ", r_squared_4))
```

```
## [1] "R-squared (Health model):  0.593108901180756"
```

```
lmmodel_5 <- lm(Happiness_Score ~ Economy, data = training_set)
summary(lmmodel_5)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Economy, data = training_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.20542 -0.45196  0.01283  0.52414  1.48846
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.4813     0.1863   18.68   <2e-16 ***
## Economy       2.1250     0.1937   10.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7083 on 76 degrees of freedom
## Multiple R-squared:  0.6129, Adjusted R-squared:  0.6078
## F-statistic: 120.3 on 1 and 76 DF,  p-value: < 2.2e-16
```

```
predictions_5 <- predict(lmmodel, newdata = training_set)
```

```r
rmse_5 <- sqrt(mean((testing_set$Happiness_Score - predictions_5)^2))
print(paste("RMSE: ", rmse))
```

```
## [1] "RMSE:  0.646727299146686"
```

```r
r_squared_5 <- summary(lmmodel_5)$r.squared
```

```r
print(paste("R-squared (Economy model): ", r_squared_5))
```

```
## [1] "R-squared (Economy model):  0.612897269110227"
```