Final Project (Group 2)

Group 2

2024-06-01

«««< 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

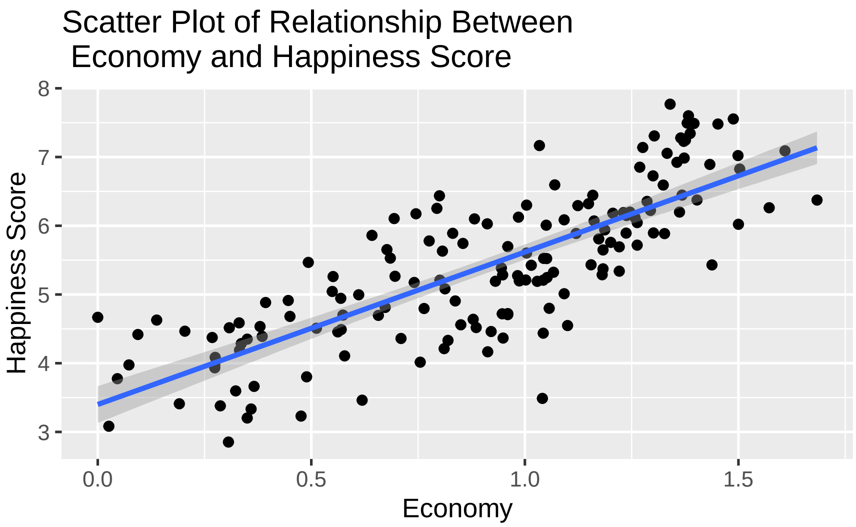
[Module 2: Junhyung Kim, Jiho Lee]

*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")
```

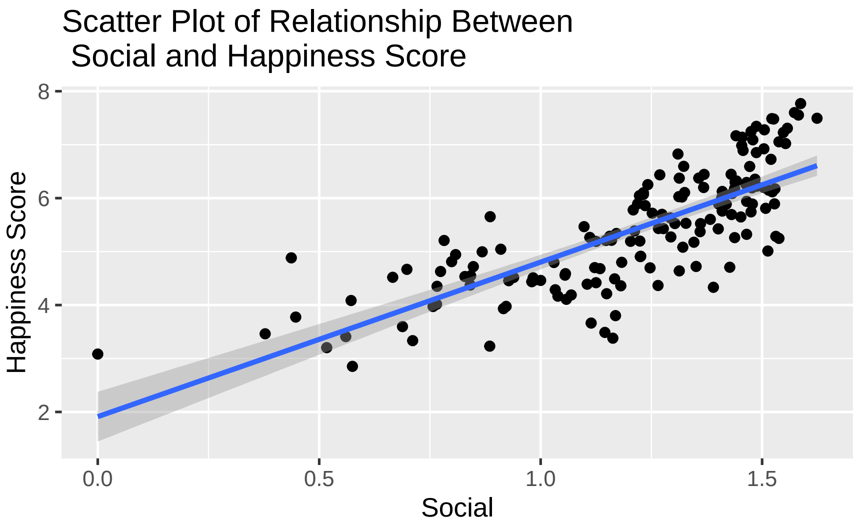
```
## Warning: ‘qplot()‘ was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## 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")
```

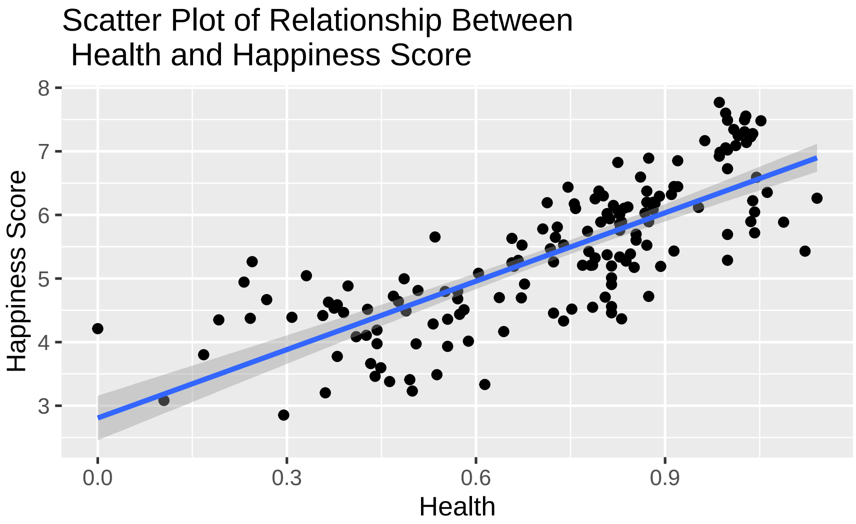
```
## Warning in geom_point(method = "lm"): Ignoring unknown parameters: ‘method‘

## ‘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")
```

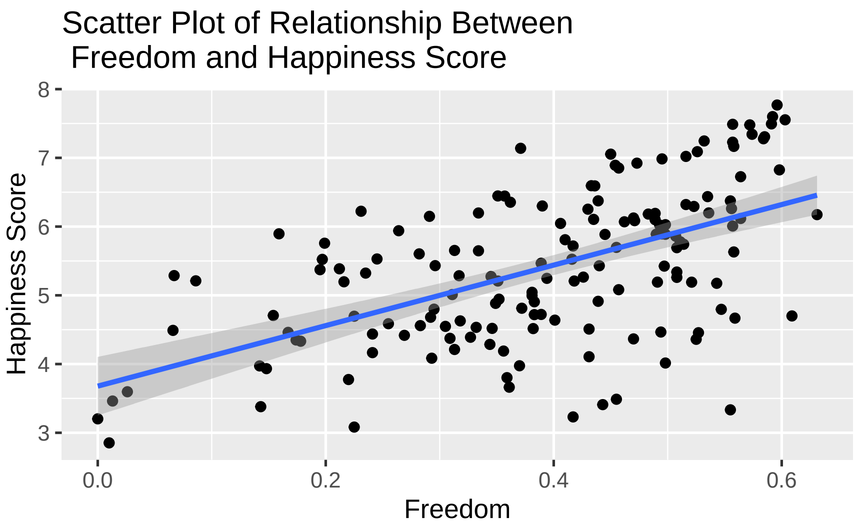
```
## Warning in geom_point(method = "lm"): Ignoring unknown parameters: ‘method‘
```

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



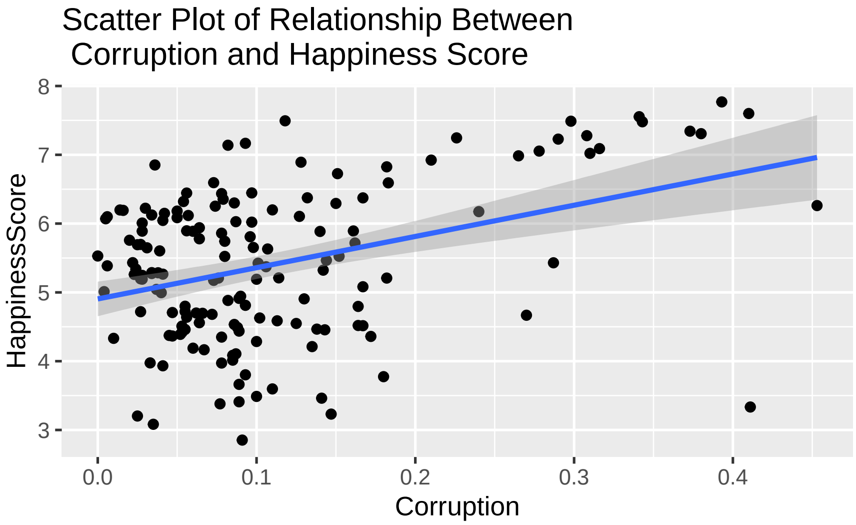
```
qplot(x= Freedom,y=Happiness_Score,data=data,
 geom=c("point","smooth"),method="lm")+
 labs(title =
 "Scatter Plot of Relationship Between
 Freedom and Happiness Score",
 x="Freedom",y="Happiness Score")
```

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



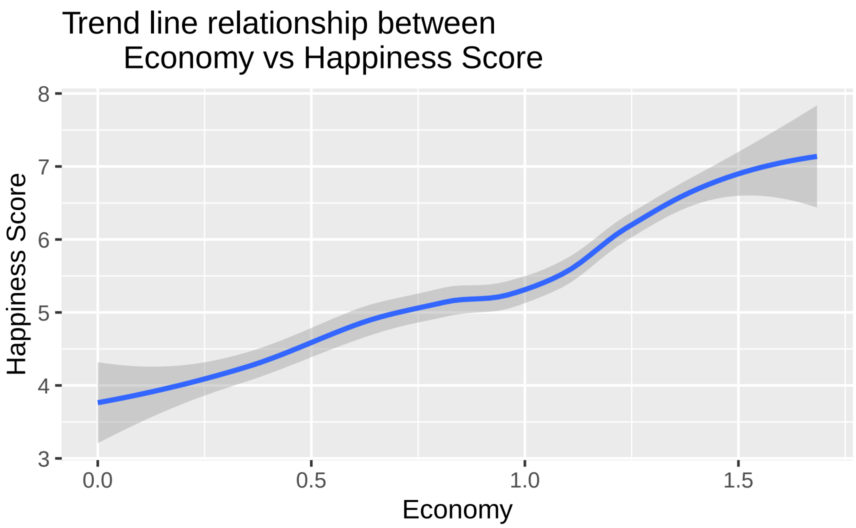
```
qplot(x= Corruption,y=Happiness_Score,data=data,
geom=c("point","smooth"),method="lm")+
labs(title ="Scatter Plot of Relationship Between
Corruption and Happiness Score",
x="Corruption",y="HappinessScore")
```

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



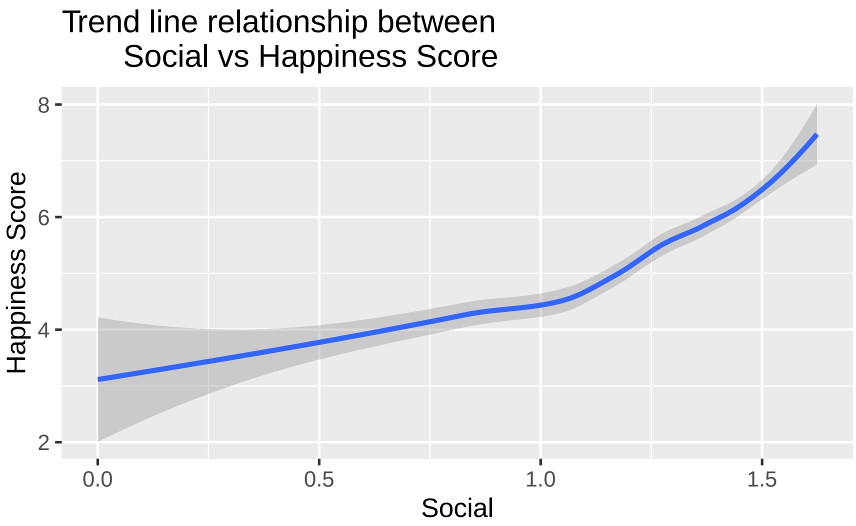
```
data %>%
  ggplot() +
  geom_smooth(mapping = aes(x = Economy, y = Happiness_Score)) +
  labs(x = "Economy", y = "Happiness Score",
       title="Trend line relationship between
       Economy vs Happiness Score")
```

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



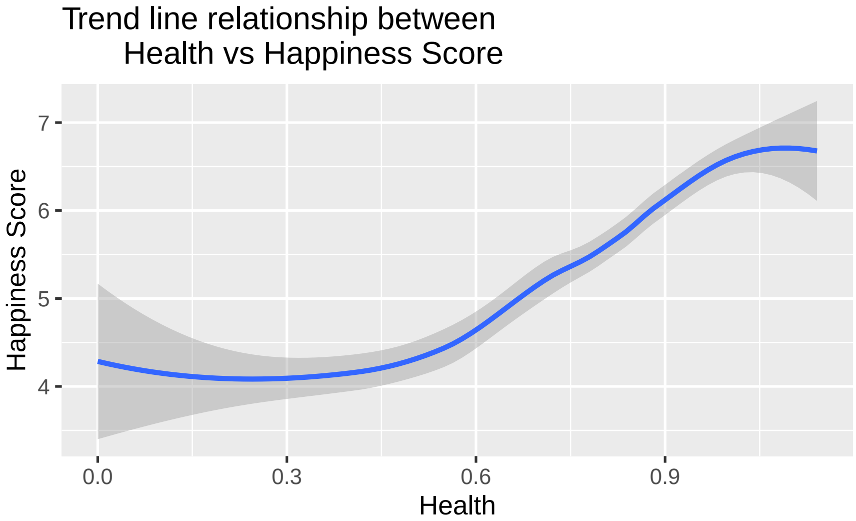
```
data %>%
  ggplot() +
  geom_smooth(mapping = aes(x = Social, y = Happiness_Score)) +
  labs(x = "Social", y = "Happiness Score",
       title="Trend line relationship between
       Social vs Happiness Score")
```

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



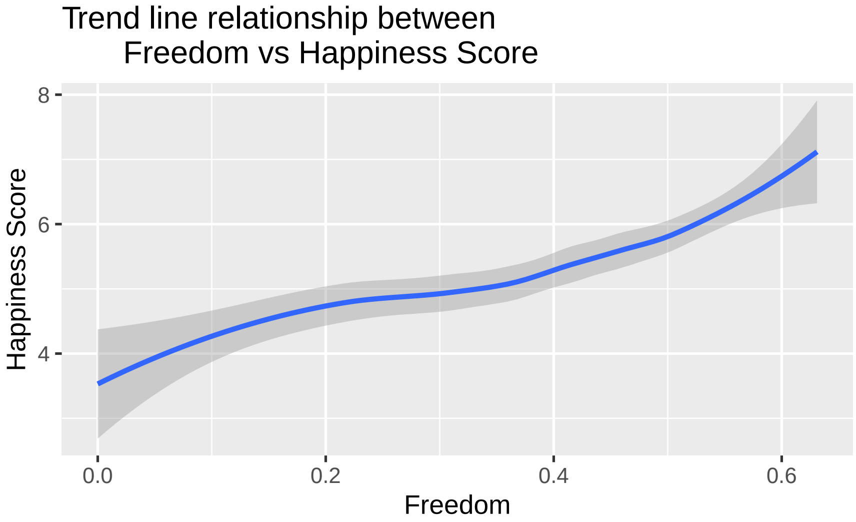
```
data %>%
  ggplot() +
  geom_smooth(mapping = aes(x = Health, y = Happiness_Score)) +
  labs(x = "Health", y = "Happiness Score",
       title="Trend line relationship between
       Health vs Happiness Score")
```

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



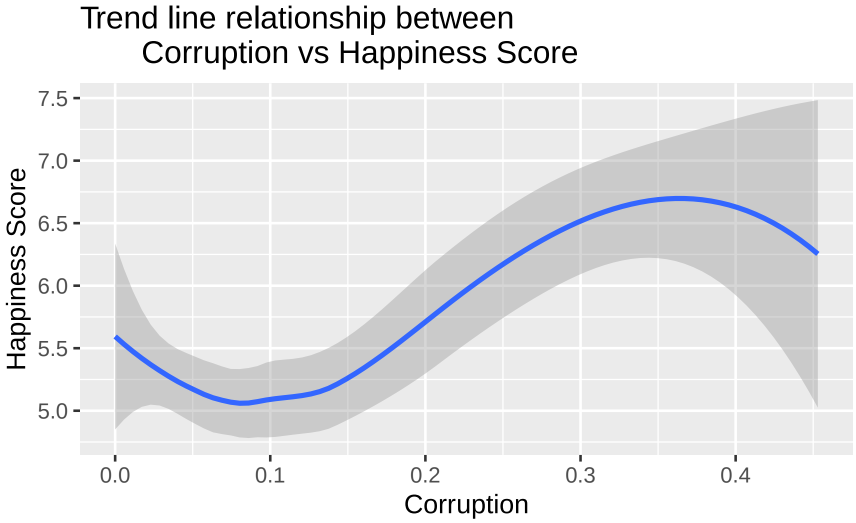
```
data %>%
  ggplot() +
  geom_smooth(mapping = aes(x = Freedom, y = Happiness_Score)) +
  labs(x = "Freedom", y = "Happiness Score",
       title="Trend line relationship between
       Freedom vs Happiness Score")
```

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



```r
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")
```

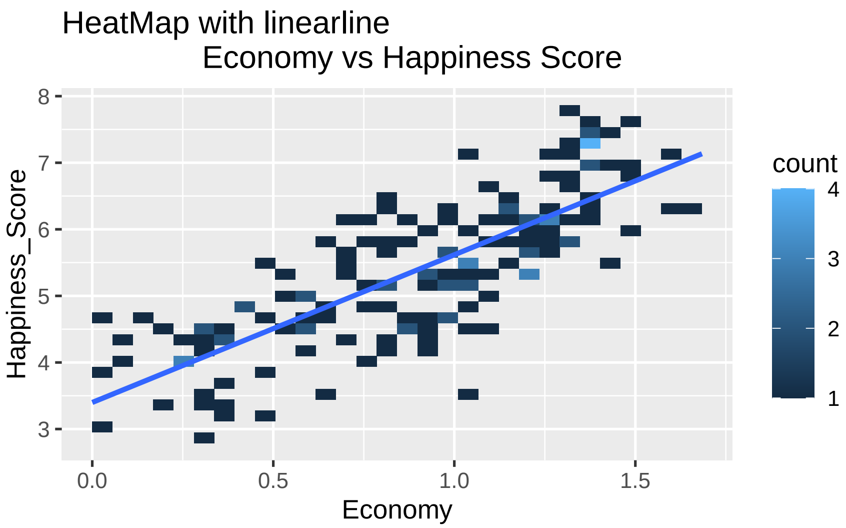
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



*HeatMap

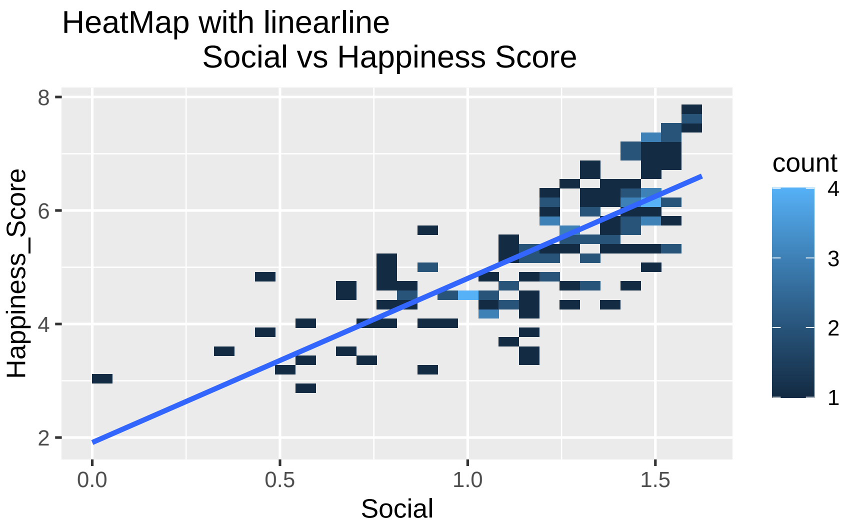
```r
data %>%
ggplot(aes(x = Economy, y = Happiness_Score)) +
  geom_bin2d(bins = 30) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "HeatMap with linearline
               Economy vs Happiness Score",
           x = "Economy", y = "Happiness_Score")
```

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



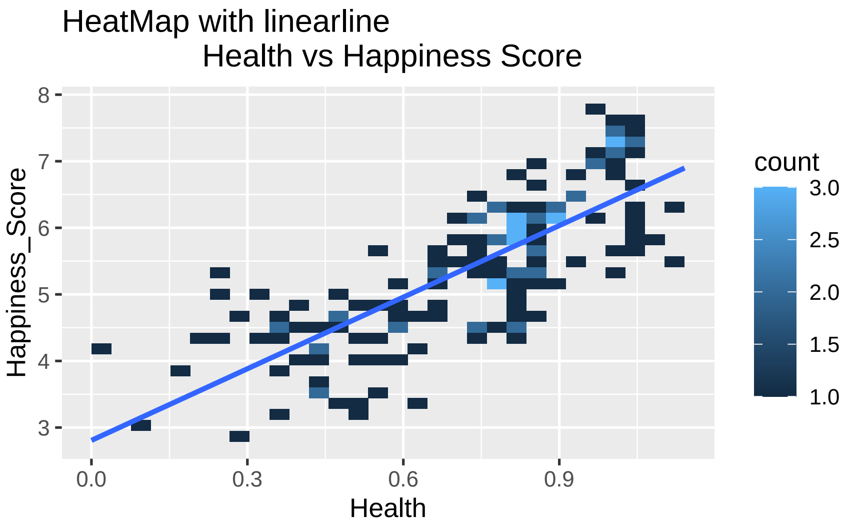
```
data %>%
ggplot(aes(x = Social, y = Happiness_Score)) +
  geom_bin2d(bins = 30) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "HeatMap with linearline
              Social vs Happiness Score",
           x = "Social", y = "Happiness_Score")
```

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



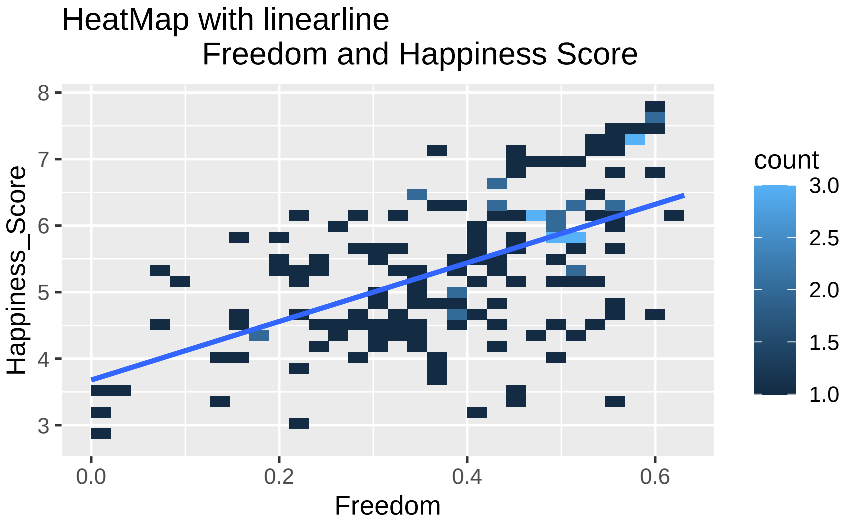
```
data %>%
ggplot(aes(x = Health, y = Happiness_Score)) +
  geom_bin2d(bins = 30) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "HeatMap with linearline
                Health vs Happiness Score",
            x = "Health", y = "Happiness_Score")
```

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



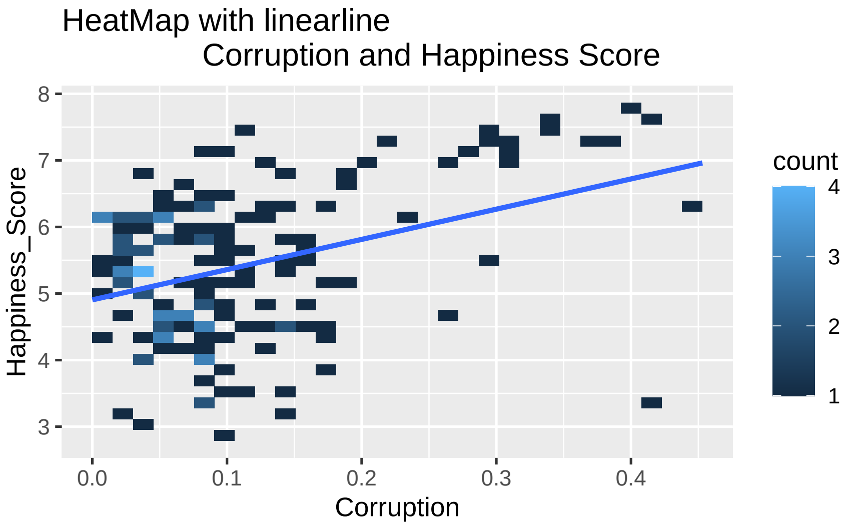
```
data %>%
ggplot(aes(x = Freedom, y = Happiness_Score)) +
  geom_bin2d(bins = 30) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "HeatMap with linearline
                Freedom and Happiness Score",
            x = "Freedom", y = "Happiness_Score")
```

```
## `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

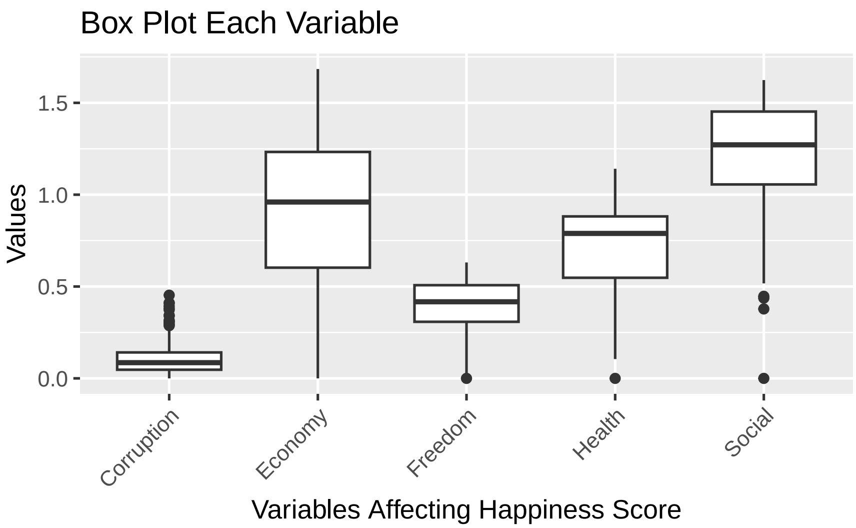
```r
tail(select(data, Economy, Social, Health, Freedom, Corruption,
            Happiness_Score)
  )
```

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

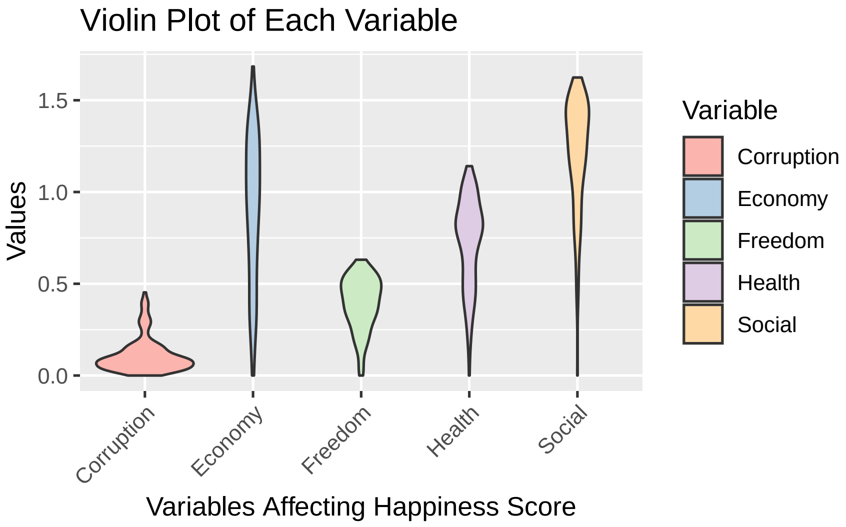
*Box Plot

```r
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
```

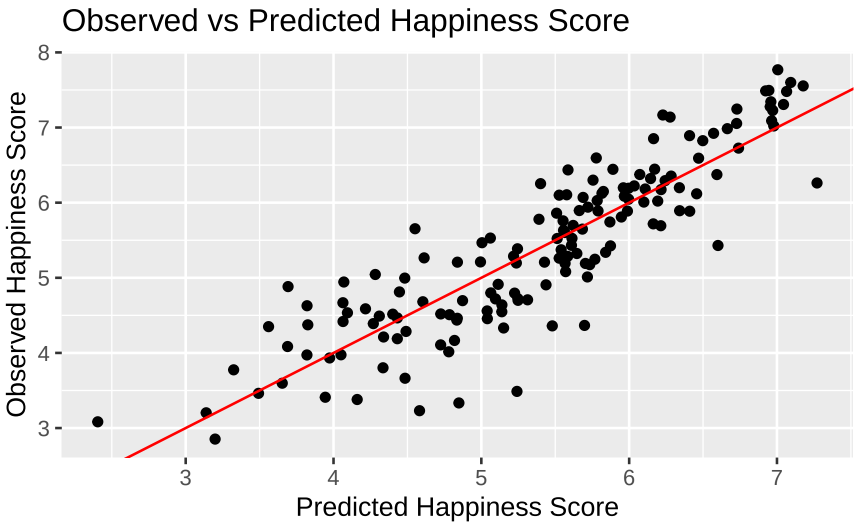
Glance to get the model's performance metrics

```
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>
```

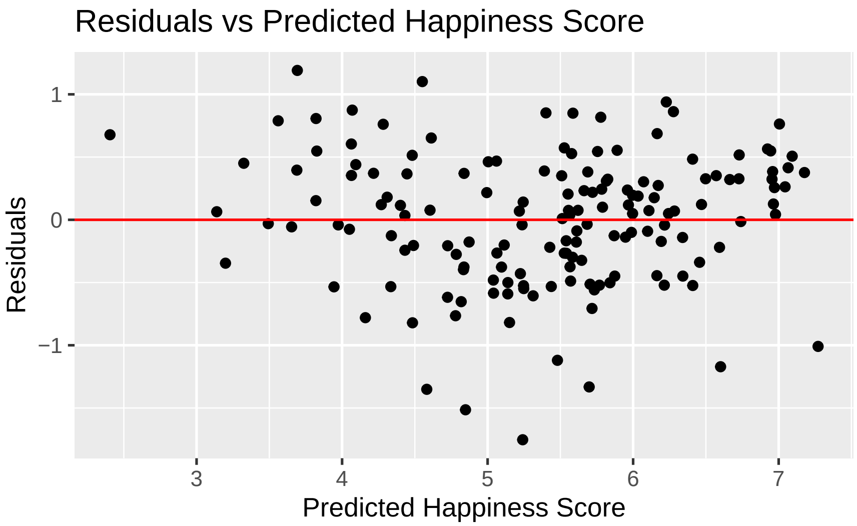
Observed vs Predicted plot

```
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")
```



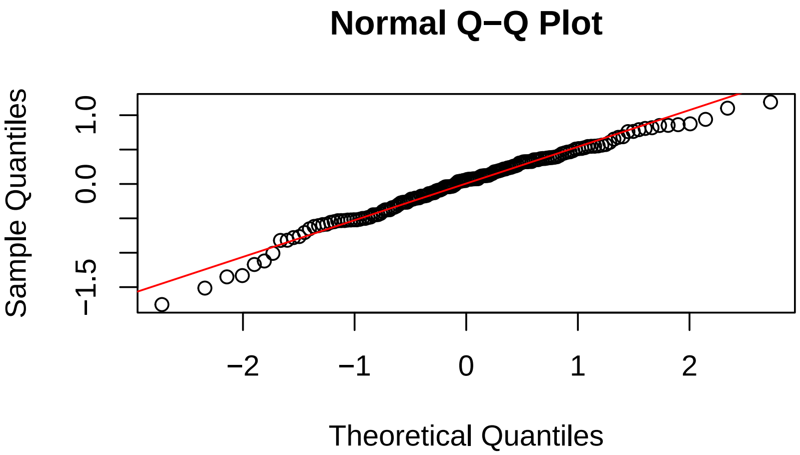
Residuals vs Predicted plot

```
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")
```



«««< HEAD

[Module 6: Sena Julsdorf and Hyeongseok Sim]

```
model_3 <- lm(Happiness_Score ~ Generosity + Corruption, data = data)

summary(model_3)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Generosity + Corruption, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4807 -0.7375  0.1946  0.8159  2.1626
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0022     0.1869  26.770  < 2e-16 ***
## Generosity   -0.6552     0.9207  -0.712    0.478
## Corruption    4.7559     0.9277   5.127 8.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.032 on 153 degrees of freedom
## Multiple R-squared:  0.1515, Adjusted R-squared:  0.1404
## F-statistic: 13.66 on 2 and 153 DF,  p-value: 3.48e-06
```

```
coefficients <- tidy(model_3)
print(coefficients)
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    5.00      0.187    26.8   1.32e-59
## 2 Generosity    -0.655     0.921    -0.712 4.78e- 1
## 3 Corruption     4.76      0.928     5.13  8.79e- 7
```

```
performance <- glance(model_3)
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     0.152         0.140  1.03      13.7 0.00000348     2  -225.  458.  470.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
model_2 <- lm(Happiness_Score ~ Economy + Social + Health + Freedom, data = data)

summary(model_2)
```

```
## 
## Call:
## lm(formula = Happiness_Score ~ Economy + Social + Health + Freedom, 
##     data = data)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.86584 -0.34594  0.03403  0.43676  1.13076 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   1.8921     0.1994   9.491  < 2e-16 ***
## Economy       0.8105     0.2165   3.745 0.000256 ***
## Social        1.0166     0.2347   4.331 2.70e-05 ***
## Health        1.1414     0.3373   3.384 0.000910 ***
## Freedom       1.8458     0.3404   5.423 2.28e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.5398 on 151 degrees of freedom
## Multiple R-squared:  0.7709, Adjusted R-squared:  0.7649 
## F-statistic:   127 on 4 and 151 DF,  p-value: < 2.2e-16
```

```r
coefficients <- tidy(model_2)
print(coefficients)
```

```
## # A tibble: 5 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    1.89      0.199      9.49 4.80e-17
## 2 Economy        0.811     0.216      3.74 2.56e- 4
## 3 Social         1.02      0.235      4.33 2.70e- 5
## 4 Health         1.14      0.337      3.38 9.10e- 4
## 5 Freedom        1.85      0.340      5.42 2.28e- 7
```

```r
performance <- glance(model_2)
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     0.771         0.765 0.540      127. 2.82e-47     4  -123.  257.  276.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
model_4 <- lm(Happiness_Score ~ Economy, data = data)

summary(model_4)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Economy, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22044 -0.48361  0.00828  0.48433  1.47409
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.3993     0.1353   25.12   <2e-16 ***
## Economy       2.2181     0.1369   16.20   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.679 on 154 degrees of freedom
## Multiple R-squared:  0.6303, Adjusted R-squared:  0.6278
## F-statistic: 262.5 on 1 and 154 DF,  p-value: < 2.2e-16
```

```r
performance <- glance(model_4)
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     0.630         0.628 0.679      262. 4.32e-35     1  -160.  326.  335.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```r
model_5 <- lm(Happiness_Score ~ Social, data = data)

summary(model_5)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Social, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89465 -0.45762 -0.01993  0.54720  1.70721
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.9124     0.2349    8.14 1.25e-13 ***
## Social        2.8910     0.1887   15.32  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.7029 on 154 degrees of freedom
## Multiple R-squared:  0.6038, Adjusted R-squared:  0.6012
## F-statistic: 234.7 on 1 and 154 DF,  p-value: < 2.2e-16
```

```
performance <- glance(model_5)
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     0.604         0.601 0.703      235. 8.98e-33     1  -165.  337.  346.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
model_6 <- lm(Happiness_Score ~ Health, data = data)

summary(model_6)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Health, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6743 -0.4621  0.0863  0.4864  1.5797
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.8068     0.1772   15.84   <2e-16 ***
## Health        3.5854     0.2319   15.46   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.699 on 154 degrees of freedom
## Multiple R-squared:  0.6082, Adjusted R-squared:  0.6057
## F-statistic: 239.1 on 1 and 154 DF,  p-value: < 2.2e-16
```

```
performance <- glance(model_6)
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     0.608         0.606 0.699      239. 3.79e-33     1  -164.  335.  344.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
model_7 <- lm(Happiness_Score ~ Freedom, data = data)

summary(model_7)
```

```
##
## Call:
## lm(formula = Happiness_Score ~ Freedom, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7882 -0.5838  0.0149  0.7029  1.8269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6788     0.2155  17.075  < 2e-16 ***
## Freedom       4.4026     0.5158   8.536 1.24e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9201 on 154 degrees of freedom
## Multiple R-squared:  0.3212, Adjusted R-squared:  0.3168
## F-statistic: 72.87 on 1 and 154 DF,  p-value: 1.238e-14
```

```
performance <- glance(model_7)
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     0.321         0.317 0.920      72.9 1.24e-14     1  -207.  421.  430.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

=======