STA437 Project

-Yuhan Hu, 1001311626 -Yuying Huang, 1

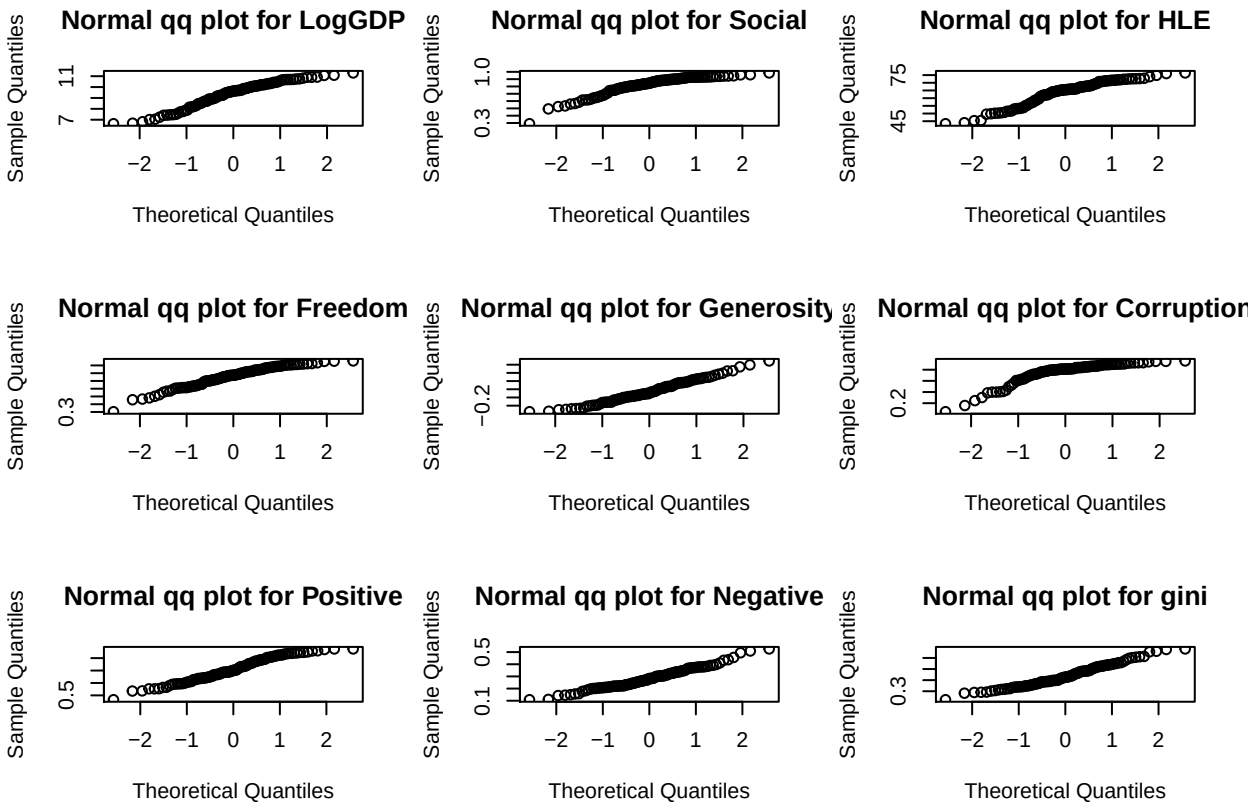
2020/3/29

Contents

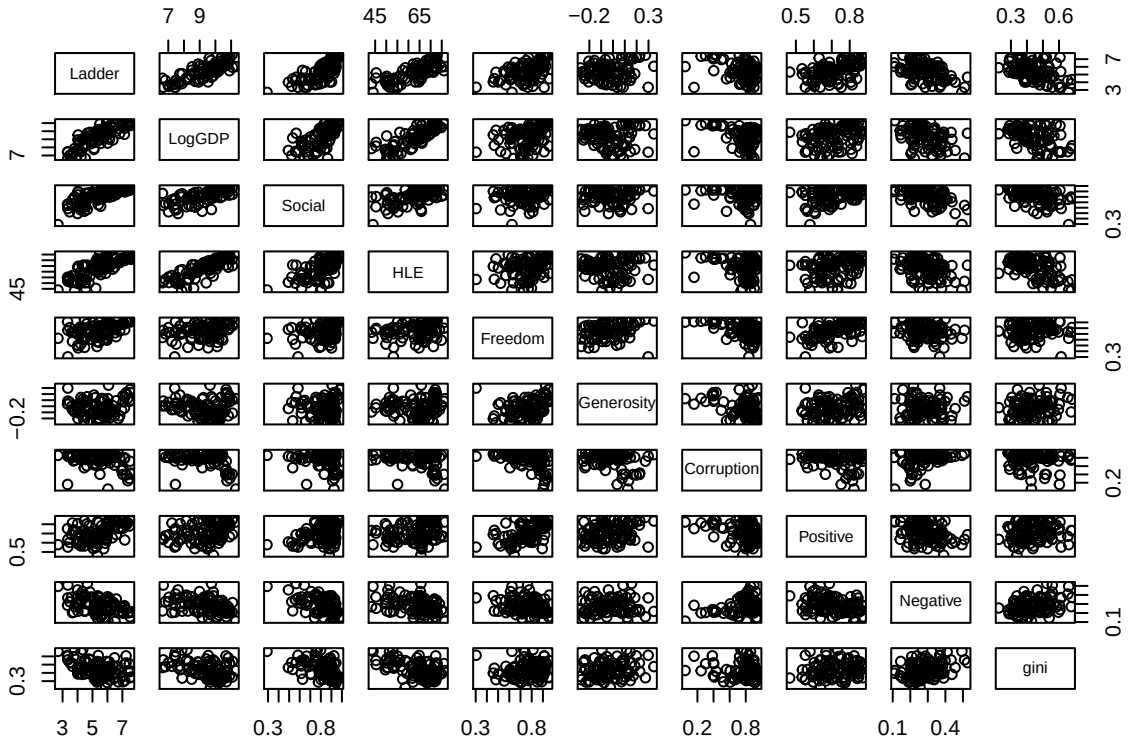
Summary

Data manipulation

```
#prepare data
original_data <- read.csv('happiness2017.csv')
set.seed(1626)
data_for_analysis <- original_data[sample(1:141,100),-1]
par(mfrow=c(3,3))
for(i in 2:10){
  qqnorm(unlist(data_for_analysis[i]),main = paste('Normal qq plot for',colnames(data_for_analysis)[i]))
}
```

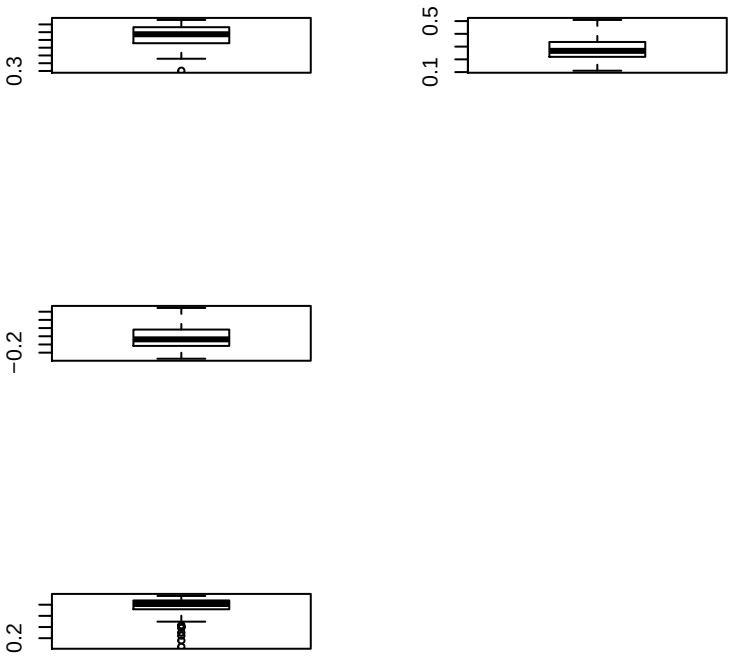



```
plot(data_for_analysis)
```




```
data_for_analysis_na.omit <- na.omit(data_for_analysis)

boxplot(data_for_analysis_na.omit[5])
boxplot(data_for_analysis_na.omit[6])
boxplot(data_for_analysis_na.omit[7])
boxplot(data_for_analysis_na.omit[9])
```



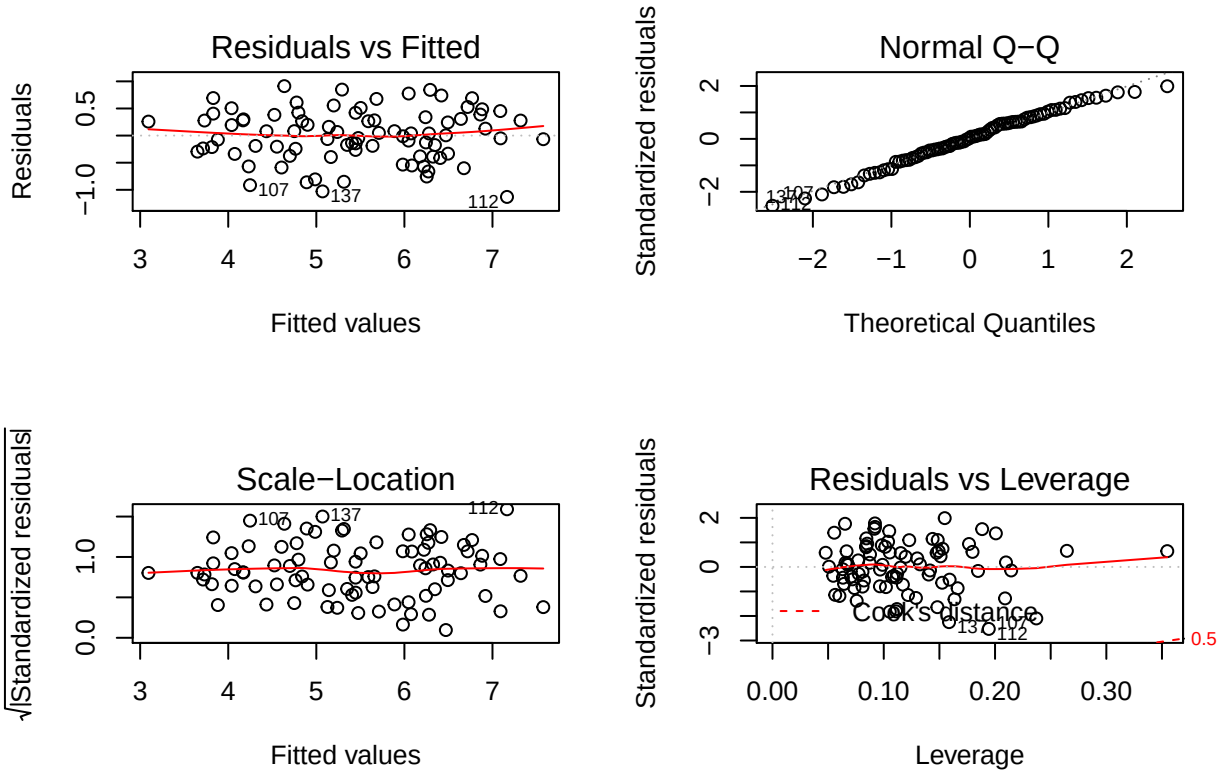
Multiple Linear Model using Original value

```
#Multiple Linear Model using Original value
fit1 <- lm(Ladder~.,data = data_for_analysis)
summary(fit1)
```

```
##
## Call:
## lm(formula = Ladder ~ ., data = data_for_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1313 -0.3054  0.0218  0.3153  0.9119
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.32813    1.06179  -3.134 0.002469 **
## LogGDP       0.38194    0.10096   3.783 0.000312 ***
## Social       1.50152    0.74894   2.005 0.048635 *
## HLE          0.02648    0.01403   1.888 0.062995 .
## Freedom      1.57340    0.53934   2.917 0.004673 **
## Generosity   1.03210    0.44732   2.307 0.023839 *
## Corruption   0.28868    0.37738   0.765 0.446733
```

```
## Positive     2.27430    0.75932   2.995 0.003730 **
## Negative    -0.05585    0.90200  -0.062 0.950792
## gini        -1.55823    0.83197  -1.873 0.065025 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4986 on 74 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:   0.83,  Adjusted R-squared:  0.8094
## F-statistic: 40.15 on 9 and 74 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2,2))
plot(fit1)
```




```
summary(fit1)
```

```
##
## Call:
## lm(formula = Ladder ~ ., data = data_for_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1313 -0.3054  0.0218  0.3153  0.9119
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.32813    1.06179  -3.134 0.002469 **
## LogGDP       0.38194    0.10096   3.783 0.000312 ***
## Social       1.50152    0.74894   2.005 0.048635 *
## HLE          0.02648    0.01403   1.888 0.062995 .
## Freedom      1.57340    0.53934   2.917 0.004673 **
## Generosity   1.03210    0.44732   2.307 0.023839 *
## Corruption   0.28868    0.37738   0.765 0.446733
## Positive     2.27430    0.75932   2.995 0.003730 **
## Negative    -0.05585    0.90200  -0.062 0.950792
## gini        -1.55823    0.83197  -1.873 0.065025 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4986 on 74 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:   0.83,  Adjusted R-squared:  0.8094
## F-statistic: 40.15 on 9 and 74 DF,  p-value: < 2.2e-16
```

```
print(xtable::xtable(summary(fit1)),type='html')
```

```
## <!-- html table generated in R 3.6.1 by xtable 1.8-4 package -->
## <!-- Fri Apr 03 14:10:58 2020 -->
## <table border=1>
## <tr> <th>  </th> <th> Estimate </th> <th> Std. Error </th> <th> t value </th> <th> Pr(&gt;|t|) </th>
##   <tr> <td align="right"> (Intercept) </td> <td align="right"> -3.3281 </td> <td align="right"> 1.06
##   <tr> <td align="right"> LogGDP </td> <td align="right"> 0.3819 </td> <td align="right"> 0.1010 </t
##   <tr> <td align="right"> Social </td> <td align="right"> 1.5015 </td> <td align="right"> 0.7489 </t
##   <tr> <td align="right"> HLE </td> <td align="right"> 0.0265 </td> <td align="right"> 0.0140 </td>
##   <tr> <td align="right"> Freedom </td> <td align="right"> 1.5734 </td> <td align="right"> 0.5393 </
##   <tr> <td align="right"> Generosity </td> <td align="right"> 1.0321 </td> <td align="right"> 0.4473
##   <tr> <td align="right"> Corruption </td> <td align="right"> 0.2887 </td> <td align="right"> 0.3774
##   <tr> <td align="right"> Positive </td> <td align="right"> 2.2743 </td> <td align="right"> 0.7593 <
##   <tr> <td align="right"> Negative </td> <td align="right"> -0.0559 </td> <td align="right"> 0.9020
##   <tr> <td align="right"> gini </td> <td align="right"> -1.5582 </td> <td align="right"> 0.8320 </td
##    </table>
```

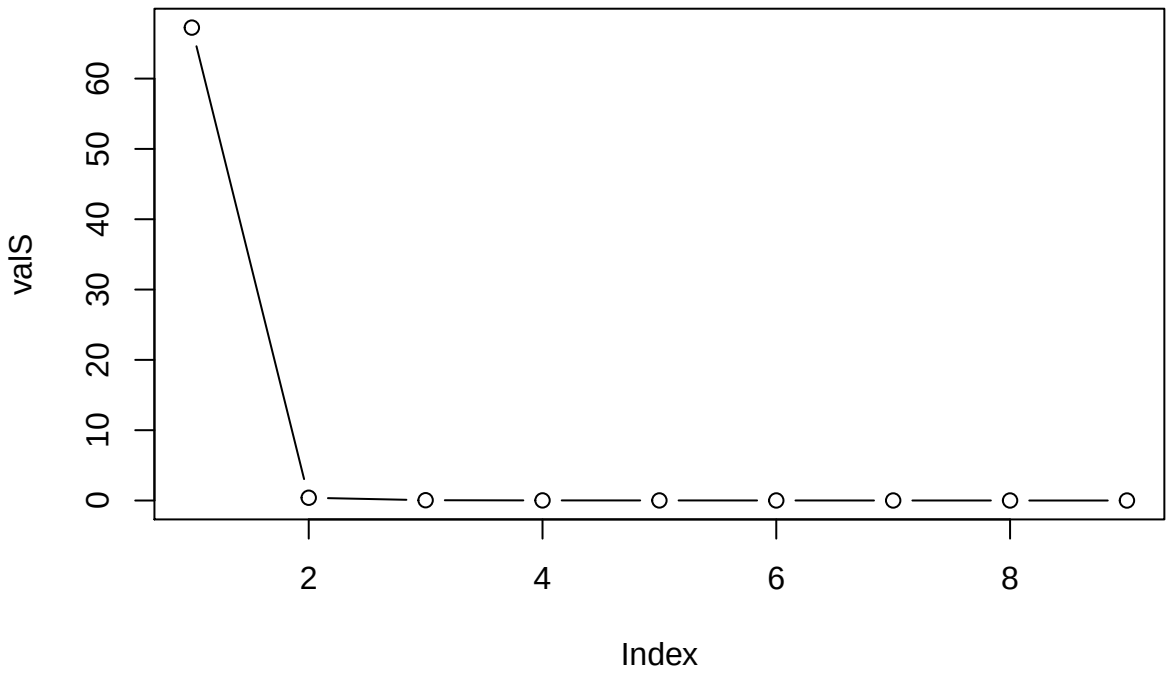
Multiple Linear Model using Principle Component

```
#Multiple Linear Model using Principle Component
##covariance approach
X <- as.matrix(data_for_analysis_na.omit[2:10])
X.bar <- apply(X,2,mean)
s <- cov(X)
valS <-eigen(s)$values
vecS <- eigen(s)$vectors
round(valS/sum(valS),3)
```

```
## [1] 0.993 0.006 0.001 0.000 0.000 0.000 0.000 0.000 0.000
```

```r
W = X

for(i in 1:9){
    for(j in 1:nrow(X)){
    W[j,i] = vecS[,i] %*% ( X[j,] -X.bar)  # centered PCs
    }}
plot(valS, type="b")
```




```r
pc1 <- lm(data_for_analysis_na.omit$Ladder~W[,1])
summary(pc1)
```

```
##
## Call:
## lm(formula = data_for_analysis_na.omit$Ladder ~ W[, 1])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.53369 -0.48937  0.04697  0.53729  1.71295
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.448077   0.079687   68.37   <2e-16 ***
## W[, 1]      -0.107498   0.009776  -11.00   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7303 on 82 degrees of freedom
## Multiple R-squared:  0.5959, Adjusted R-squared:  0.591
## F-statistic: 120.9 on 1 and 82 DF,  p-value: < 2.2e-16
```

```
pc2 <- lm(data_for_analysis_na.omit$Ladder~W[,1]+W[,2])
summary(pc2)
```

```
##
## Call:
## lm(formula = data_for_analysis_na.omit$Ladder ~ W[, 1] + W[,
##     2])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.57549 -0.44311 -0.02011  0.48823  1.28896
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.448077   0.069674  78.193  < 2e-16 ***
## W[, 1]      -0.107498   0.008547 -12.577  < 2e-16 ***
## W[, 2]       0.570792   0.111386   5.124 1.99e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6386 on 81 degrees of freedom
## Multiple R-squared:  0.6948, Adjusted R-squared:  0.6873
## F-statistic: 92.22 on 2 and 81 DF,  p-value: < 2.2e-16
```

```
pc3 <- lm(data_for_analysis_na.omit$Ladder~W[,1]+W[,2]+W[,3])
summary(pc3)
```

```
##
## Call:
## lm(formula = data_for_analysis_na.omit$Ladder ~ W[, 1] + W[,
##     2] + W[, 3])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.64373 -0.31012  0.02317  0.31513  1.23765
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.448077   0.065830  82.759  < 2e-16 ***
## W[, 1]      -0.107498   0.008076 -13.311  < 2e-16 ***
## W[, 2]       0.570792   0.105241   5.424 6.04e-07 ***
## W[, 3]      -1.021727   0.311831  -3.277  0.00156 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6033 on 80 degrees of freedom
## Multiple R-squared:  0.7309, Adjusted R-squared:  0.7209
## F-statistic: 72.45 on 3 and 80 DF,  p-value: < 2.2e-16
```

```r
#Multiple Linear Model using Principle Component
##correlation approach
R <- cor(X)
valR <- eigen(R)$values
vecR <- eigen(R)$vectors
round(valR/sum(valR),3)
```

```
## [1] 0.401 0.230 0.089 0.079 0.071 0.050 0.039 0.028 0.014
```

```r
rownames(vecR) <- colnames(X)
colnames(vecR) <- c('PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9')
round(vecR,2)
```

```
##              PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9
## LogGDP     -0.46 -0.16  0.03 -0.11 -0.26 -0.17 -0.13 -0.34  0.72
## Social     -0.43 -0.12  0.27 -0.07  0.32 -0.23 -0.52  0.54 -0.10
## HLE        -0.44 -0.19 -0.08 -0.33 -0.31 -0.12  0.05 -0.33 -0.66
## Freedom    -0.26  0.44  0.22  0.10 -0.24  0.74 -0.28 -0.04 -0.05
## Generosity -0.07  0.46 -0.37 -0.69  0.40  0.05  0.04 -0.03  0.12
## Corruption  0.27 -0.36  0.54 -0.31  0.38  0.24 -0.13 -0.43 -0.01
## Positive   -0.26  0.39  0.59  0.05  0.11 -0.24  0.60  0.00  0.00
## Negative    0.36  0.03  0.29 -0.53 -0.59 -0.08 -0.03  0.37  0.08
## gini        0.26  0.49  0.10  0.14 -0.06 -0.49 -0.51 -0.39 -0.12
```

```r
W.new = X  # just to create a data matrix of the same size of X
colnames(W.new) = c('PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9')

# now fill in the entries by calculating sample PCs

for(i in 1:9){ # PC's
    for(j in 1:nrow(X)){
    W.new[j,i] = vecR[,i] %*% X[j,]   # no need to center when using normalized PCCs
 }}


pc1 <- lm(data_for_analysis_na.omit$Ladder~W[,1]+W[,2]+W[,3])
summary(pc1)
```

```
##
## Call:
## lm(formula = data_for_analysis_na.omit$Ladder ~ W[, 1] + W[,
##     2] + W[, 3])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.64373 -0.31012  0.02317  0.31513  1.23765
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.448077   0.065830  82.759  < 2e-16 ***
## W[, 1]      -0.107498   0.008076 -13.311  < 2e-16 ***
## W[, 2]       0.570792   0.105241   5.424 6.04e-07 ***
```

```
## W[, 3]      -1.021727   0.311831  -3.277  0.00156 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6033 on 80 degrees of freedom
## Multiple R-squared:  0.7309, Adjusted R-squared:  0.7209
## F-statistic: 72.45 on 3 and 80 DF,  p-value: < 2.2e-16
```

Appendix

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
#prepare data
original_data <- read.csv('happiness2017.csv')
set.seed(1626)
data_for_analysis <- original_data[sample(1:141,100),-1]
par(mfrow=c(3,3))
for(i in 2:10){
  qqnorm(unlist(data_for_analysis[i]),main = paste('Normal qq plot for',colnames(data_for_analysis)[i]))
}
plot(data_for_analysis)
data_for_analysis_na.omit <- na.omit(data_for_analysis)

boxplot(data_for_analysis_na.omit[5])
boxplot(data_for_analysis_na.omit[6])
boxplot(data_for_analysis_na.omit[7])
boxplot(data_for_analysis_na.omit[9])
#Multiple Linear Model using Original value
fit1 <- lm(Ladder~.,data = data_for_analysis)
summary(fit1)
par(mfrow = c(2,2))
plot(fit1)
summary(fit1)
print(xtable::xtable(summary(fit1)),type='html')
#Multiple Linear Model using Principle Component
##covariance approach
X <- as.matrix(data_for_analysis_na.omit[2:10])
X.bar <- apply(X,2,mean)
s <- cov(X)
valS <-eigen(s)$values
vecS <- eigen(s)$vectors
round(valS/sum(valS),3)
W = X

for(i in 1:9){
    for(j in 1:nrow(X)){
    W[j,i] = vecS[,i] %*% ( X[j,] -X.bar)  # centered PCs
    }}
plot(valS, type="b")

pc1 <- lm(data_for_analysis_na.omit$Ladder~W[,1])
summary(pc1)
```

```r
pc2 <- lm(data_for_analysis_na.omit$Ladder~W[,1]+W[,2])
summary(pc2)
pc3 <- lm(data_for_analysis_na.omit$Ladder~W[,1]+W[,2]+W[,3])
summary(pc3)

#Multiple Linear Model using Principle Component
##correlation approach
R <- cor(X)
valR <- eigen(R)$values
vecR <- eigen(R)$vectors
round(valR/sum(valR),3)

rownames(vecR) <- colnames(X)
colnames(vecR) <- c('PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9')
round(vecR,2)

W.new = X  # just to create a data matrix of the same size of X
colnames(W.new) = c('PC1','PC2','PC3','PC4','PC5','PC6','PC7','PC8','PC9')

# now fill in the entries by calculating sample PCs

for(i in 1:9){ # PC's
    for(j in 1:nrow(X)){
    W.new[j,i] = vecR[,i] %*% X[j,]   # no need to center when using normalized PCCs
 }}


pc1 <- lm(data_for_analysis_na.omit$Ladder~W[,1]+W[,2]+W[,3])
summary(pc1)
```