## Main Report

### Summary/Abstract

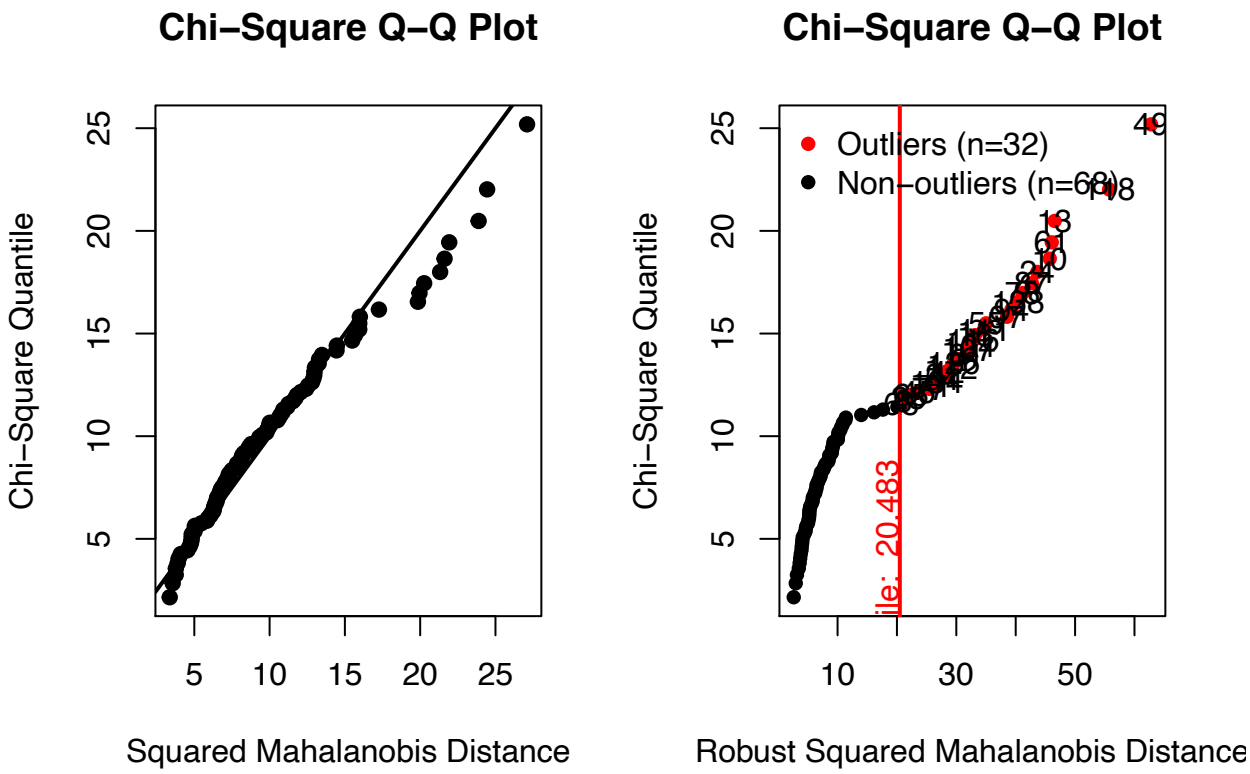
The goal of this project is to model the relationship between happiness score (which is called Ladder in our data) and other explanatory variables in the data set. We will attempt to do this by utilizing multiple regression and principal component approaches. Each approach has its own requirements that needs to be fulfilled, before either method can be used. We will use different analysis techniques and make inferences about the data, in order to check if the basic requirements have been met. If analysis results failed to meet necessary expectations, then we will attempt to utilize a number of techniques to transform our data into a form that suits our needs.

### Data Manipulation and Summary

Before we can begin analysis, we first need to clean up the data and make sure that it fulfils the multivariate normality assumption. Let's begin our data manipulation, with first by cleaning and removing rows that contain missing values. Next, since we will be performing our analysis on the sample, we would like to take a sample of the size 100 using random sampling technique. Since we would like to ensure that our data is entirely numerical, the categorical variable "country" will need to be removed from our sample.

```
##     Ladder LogGDP Social   HLE Freedom Generosity Corruption Positive Negative
## 24    4.03   7.58   0.62 45.11    0.53       0.04       0.82     0.58     0.47
## 37    6.14   9.02   0.79 64.11    0.80      -0.19       0.80     0.76     0.35
## 110   5.75   9.48   0.89 66.07    0.61      -0.07       0.89     0.53     0.30
## 84    5.58   8.47   0.84 63.91    0.56       0.01       0.97     0.62     0.27
## 40    7.66  10.57   0.95 71.38    0.95      -0.04       0.25     0.80     0.18
## 107   3.33   7.46   0.67 55.09    0.91       0.03       0.16     0.75     0.29
##     gini
## 24  0.50
## 37  0.46
## 110 0.35
## 84  0.37
## 40  0.39
## 107 0.60
```

Now let's continue the data manipulation, by testing for multivariate normality. In R MVN library is perfect for this purpose. The reason we test for multivariate normality assumption is because most of the techniques that we will cover at the later stages of analysis require normality and quality of inferences also depends on it as well. Since our data is multivariate, to check for normality assumption let's begin by looking at the Chi-Square QQ plot.




As we can see from the plot, there are several outliers that deviate from the line, which makes us suspect that our data is not multivariate normal. To visualize outliers better, let's check Chi-Square Quantile vs Robust Squared Mahalonobis Distance plot. The plot shows a significant number of outliers. For the purpose of checking the properties of our sample, we will need to examine summary statistic.

```
## Summary Statics:
```

| | n | Mean | Std.Dev | Median | Min | Max | 25th | 75th | Skew | Kurtosis |
|---|---|---|---|---|---|---|---|---|---|---|
| Ladder | 100 | 5.38 | 1.20 | 5.41 | 2.89 | 7.66 | 4.50 | 6.16 | 0.03 | -0.86 |
| LogGDP | 100 | 9.11 | 1.22 | 9.28 | 6.63 | 11.30 | 8.10 | 10.16 | -0.22 | -1.13 |
| Social | 100 | 0.81 | 0.12 | 0.84 | 0.49 | 0.98 | 0.75 | 0.91 | -0.78 | -0.36 |
| HLE | 100 | 62.19 | 8.29 | 63.96 | 43.38 | 76.41 | 55.26 | 68.36 | -0.46 | -0.84 |
| Freedom | 100 | 0.77 | 0.12 | 0.77 | 0.30 | 0.96 | 0.70 | 0.86 | -0.87 | 0.99 |
| Generosity | 100 | 0.00 | 0.15 | -0.02 | -0.27 | 0.49 | -0.09 | 0.08 | 0.56 | 0.30 |
| Corruption | 100 | 0.75 | 0.19 | 0.81 | 0.05 | 0.97 | 0.72 | 0.86 | -1.69 | 2.42 |
| Positive | 100 | 0.72 | 0.09 | 0.73 | 0.53 | 0.92 | 0.64 | 0.81 | -0.14 | -1.08 |
| Negative | 100 | 0.28 | 0.09 | 0.27 | 0.11 | 0.55 | 0.22 | 0.34 | 0.62 | 0.05 |
| gini | 100 | 0.46 | 0.10 | 0.44 | 0.28 | 0.74 | 0.38 | 0.53 | 0.48 | -0.40 |

In order to get a general idea of central tendencies (i.e. mean, median) and measure the spread (i.e. IQR, range, standard deviation), we analyze the summary statistics that describe our data. From our summary, we can see that variable with largest spread is "HLE". Since mean value for each variable is very close to the median values, we can deduce that distribution of values for each variable is mostly symmetric.

```
## Marginal Normality Test:
```

| | Test | Variable | Statistic | p value | Normality |
|---|---|---|---|---|---|
| 1 | Shapiro-Wilk | Ladder | 0.9793 | 0.1177 | YES |
| 2 | Shapiro-Wilk | LogGDP | 0.9586 | 0.0032 | NO |

```
## 3  Shapiro-Wilk   Social      0.9133  <0.001      NO
## 4  Shapiro-Wilk    HLE        0.9501   8e-04      NO
## 5  Shapiro-Wilk  Freedom      0.9509   9e-04      NO
## 6  Shapiro-Wilk Generosity    0.9774  0.0829      YES
## 7  Shapiro-Wilk Corruption    0.8011  <0.001      NO
## 8  Shapiro-Wilk  Positive     0.9668  0.0126      NO
## 9  Shapiro-Wilk  Negative     0.9671  0.0134      NO
## 10 Shapiro-Wilk    gini       0.9702  0.0227      NO
```

Next, lets take a look at marginal normality. By applying, Shapiro-Wilk on all of the variables. Since p-value of some of the variables is less than the significance level of 0.05, there is evidence that those variables are not Normal. We should not that marginal normality is less crucial than Joint normality, which is why we should also test for joint normality.

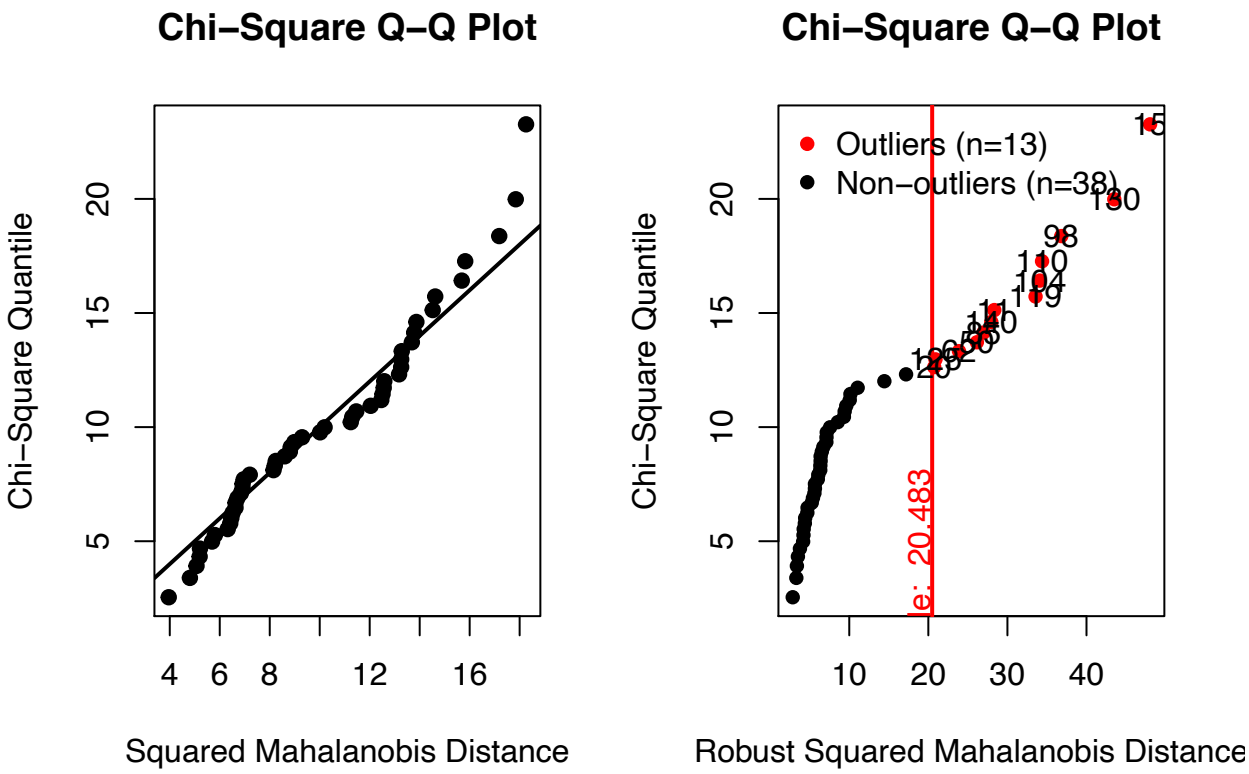
```
## Multivariate Normality Test:
```

```
##              Test        Statistic              p value Result
## 1 Mardia Skewness 405.757275018031 3.54581795042121e-13     NO
## 2 Mardia Kurtosis 1.97739337344036   0.0479971824300476     NO
## 3             MVN             <NA>                 <NA>     NO
```

As we see, our sample fails multivariate normality test. We can fix this by making a suitable transformation and removing outliers, to see if this allows us to reach near normality. One of the commonly used transformations is square root transformation. Upon examining our data, we realized that our sample contains negative values absolute value of which are less than 1. Square root transformation on negative values would've failed, which is why we also had to scale the values by 1. The final transformation, that we decided to go with is.

$$\sqrt{X+1}$$

where X, is our data matrix.




Now that we have made a suitable transformation, we can see that made changes helped us to reach near normality. Our chi-squared qq plot, shows scatter points close to the line. In order to test if our sample, is multivariate normal, we will once again perform hypothesis testing.

```
## Multivariate Normality Test:

##              Test         Statistic           p value Result
## 1 Mardia Skewness  236.722064273558 0.209131515270336    YES
## 2 Mardia Kurtosis -1.32824987206464 0.184095573786128    YES
## 3             MVN              <NA>              <NA>    YES
```

As we can see now our data fulfils, multivariate normality assumption.

### Multiple Linear Model using Original variables

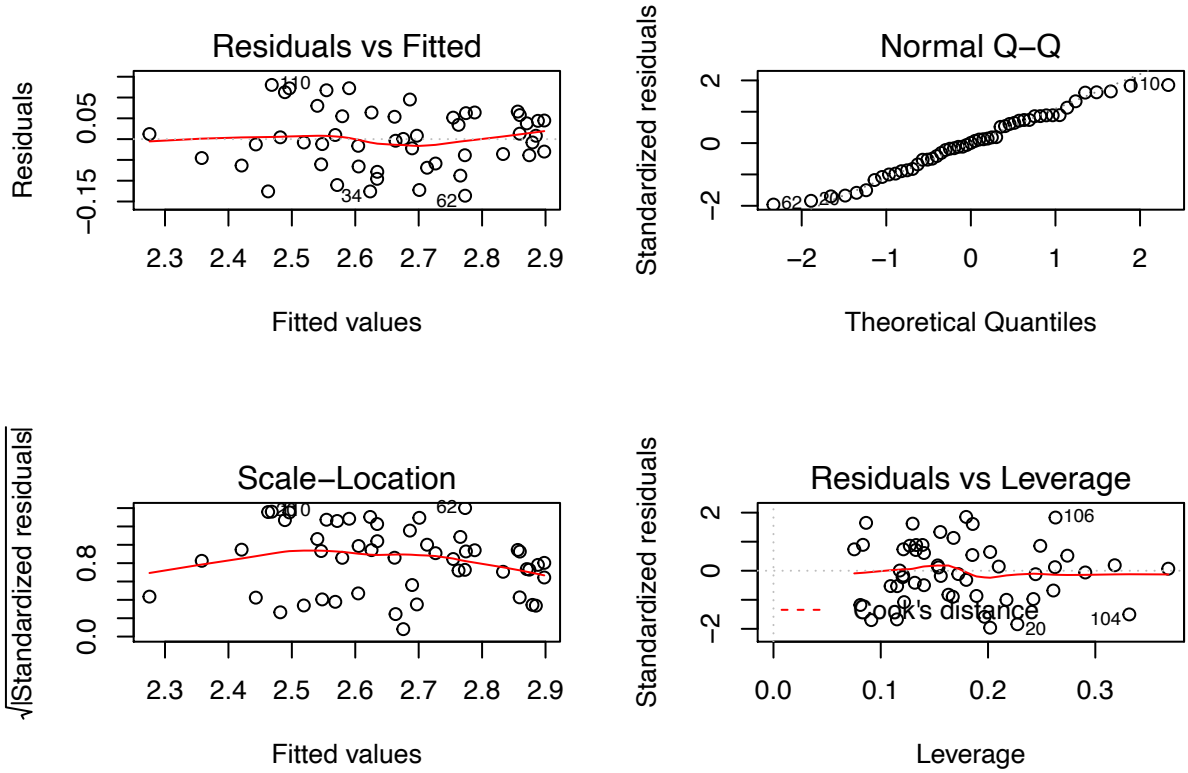
Now let's attempt modeling the relationship between happiness score (which is called Ladder in our data) and other explanatory variables. In this section we will be using multiple regression approach, which will allow us to build to fit a linear model for the happiness score. However, before we can proceed with that, we need to check whether the assumptions of this model are satisfied. This can be done by examining model residuals, through diagnostic plots. The assumptions that we are checking for are:

- Independent variables (the covariates) are not highly correlated with each other
- Y value can be expressed as linear function of x variables
- Variation of residuals is constant (called homoscedasticity)
- Error term is Normally distributed

To check, if covariates are highly correlated, we will need to check the VIF score. The smallest possible value of VIF is one (absence of multicollinearity). As a rule of thumb, a VIF value that exceeds 10 indicates a problematic amount of collinearity.

```
## VIF Scores are:
##     LogGDP     Social        HLE    Freedom Generosity Corruption   Positive
##     13.392      6.505      6.084      4.777      1.815      2.871      3.931
##   Negative       gini
##      3.777      2.483
```

As we can see, the VIF score for the predictor variable LogGDP is very high (VIF = 13.392). This might be problematic. When faced to multicollinearity, the concerned variables should be removed, since the presence of multicollinearity implies that the information that this variable provides about the response is redundant in the presence of the other variables.




Now that we have removed the problematic variable, let's check other assumptions are satisfied. First plot shows red trend line of our model appears to be more or less flat, hence I don't this there is any violation of linearity assumption. Second plot (QQ plot), shows that normality assumption of error terms is mostly satisfied. Third plot shows that, variance of residuals more or less constant, hence homoscedasticity assumption is not violated. Since all the multivariate assumptions have been satisfied, we can proceed fitting our model.

```
## Adjusted R Squared is:
```

```
## [1] 0.7945998
```

As we can see, our model has yielded a fairly larger R squared value, which illustrates the proportion of the variance in the happiness score that is predictable from other 8 covariates.

### Multiple Linear Model using Principal Components

Now let us try another approach for modeling the relationship between happiness score (which is called Ladder) and other explanatory variables. In this section we will be using principal component approach,

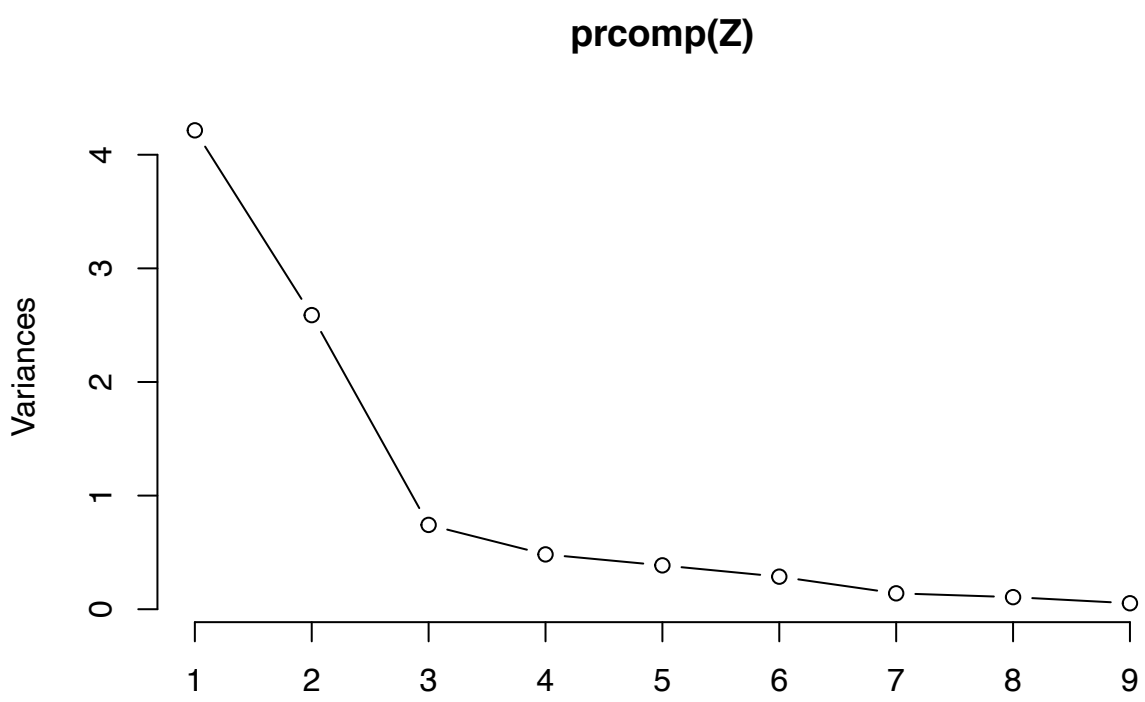
which concerns finding a good linear combination of a set of variables which explains most of the variability or randomness in the set. Since we are using 9 explanatory variables to predict happiness score, there are 9 possible principal components. In other word our goal is to find a random vector W whose components are uncorrelated. To do that, we will use following formula:

$$W_i = e_i' X$$

Where X is a data matrix of associated variables and $e_i'$ is an eigen vector. To begin let's first decide if we should use covariance or corrolation matrix to find principal components. We will do that by analyzing the covariance matrix.

```
##            LogGDP Social    HLE Freedom Generosity Corruption Positive Negative
## LogGDP      0.016  0.002  0.035   0.001      0.001     -0.006    0.001   -0.003
## Social      0.002  0.000  0.005   0.000      0.000     -0.001    0.000   -0.001
## HLE         0.035  0.005  0.096   0.003      0.004     -0.013    0.003   -0.006
## Freedom     0.001  0.000  0.003   0.002      0.002     -0.002    0.001    0.000
## Generosity  0.001  0.000  0.004   0.002      0.006     -0.003    0.001    0.000
## Corruption -0.006 -0.001 -0.013  -0.002     -0.003      0.007   -0.001    0.001
## Positive    0.001  0.000  0.003   0.001      0.001     -0.001    0.001    0.000
## Negative   -0.003 -0.001 -0.006   0.000      0.000      0.001    0.000    0.001
## gini       -0.003  0.000 -0.006   0.001      0.001      0.000    0.000    0.001
##               gini
## LogGDP     -0.003
## Social      0.000
## HLE        -0.006
## Freedom     0.001
## Generosity  0.001
## Corruption  0.000
## Positive    0.000
## Negative    0.001
## gini        0.002
```

Notice that in the covariance matrix, the scales for explanatory variables are quite different, which suggests that we may be better off using correlation to get the principal components.




Now that we have computed principal components, lets decide how many components we should keep. We will be using scree-plot method for deciding how many principal components to retain. From the scree plot, we can see each principal component in successive order, we can see that components 1 to 5 form an elbow in the curve, which means that we retain 5 principal components.

```
##                 Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)  3.289527684 0.003982873 825.918199 1.017198e-95
## W.new[, 1]  -0.057904321 0.001959344 -29.552906 4.099155e-31
## W.new[, 2]  -0.017539414 0.002500153  -7.015337 9.670057e-09
## W.new[, 3]  -0.006334601 0.004669078  -1.356713 1.816397e-01
## W.new[, 4]  -0.024239224 0.005790336  -4.186152 1.300650e-04
## W.new[, 5]   0.011938373 0.006474261   1.843975 7.177694e-02
```

```
## Adjusted R Squared is:
```

```
## [1] 0.949513
```

Now that we have decided which components to keep lets fit a linear model for the response variable happiness score using 4 principal components and take a look at the model Coefficients and . From the p values, we can see that component W3.new and W4.new seem not to be significant, which is why let's remove W3.new and W4.new.

```
## Adjusted R Squared is:
```

```
## [1] 0.9308602
```

Notice that removing 2 components did not significantly change Adjusted R squared value. We also should note that removing those principal components from the model does not change the coefficients. Now that we have our model, lets interpret the principal components that we have retained, by checking the importance of each variable in the first, second and fifth principal component.

```
##     LogGDP     Social        HLE    Freedom Generosity Corruption   Positive
##     -0.457     -0.423     -0.423     -0.224     -0.168      0.340     -0.173
##   Negative       gini
##      0.401      0.218
```

As we can see LogGDP, Social, HLE contribute to first principal component more than other variables.

```
##     LogGDP     Social        HLE    Freedom Generosity Corruption   Positive
##     -0.139     -0.202     -0.093      0.490      0.408     -0.256      0.474
##   Negative       gini
##      0.143      0.463
```

"Freedom", "Generosity", "Positive" and "gini" contribute to the second principal component more than other variables.

```
##     LogGDP     Social        HLE    Freedom Generosity Corruption   Positive
##      0.094      0.207     -0.248      0.078     -0.532     -0.337     -0.243
##   Negative       gini
##     -0.259      0.599
```

"Generosity", "Corruption" and "gini" contribute to the fifth principal component more than other variables.

# 2005 HS Winter 2020 Project Report

## Appendix

```
library(tidyverse)
library(MVN)
library(car)
```

### Data Manipulation and Summary

```
# read the data
hapiness = read.csv("./happiness2017.csv")

#=======================================
#             Clean The Data
#=======================================

# clean raw data by removing rows with NA values
hapiness = na.omit(hapiness)
# Set the seed of your randomization to be the last four digits of your student number
set.seed(6066)
# take a random sample of 100 countries
hapiness.sample = sample(nrow(hapiness), 100, replace = FALSE)
hapiness.sample_hapiness = hapiness[hapiness.sample,]

# remove country variable from our sample
hapiness.sample_hapiness = hapiness.sample_hapiness[,-1]

# change data into matrix form
hapiness.sample_matrix = data.matrix(hapiness.sample_hapiness)

# view some of the data
head(round(hapiness.sample_hapiness,2))

#=======================================
#         Multivariate Normality
#=======================================
# set to view 2 plots in 1
par(mfrow=c(1,2))
# Chi squared plot along the line to assess normality
mvn_result = mvn(data = hapiness.sample_matrix, multivariatePlot = "qq", multivariateOutlierMethod = "q

# obtain summary statistics
cat("Summary Statics: \n")
mvn_result$Descriptives
# Marginal Normality test
cat("Marginal Normality Test: \n")
mvn_result$univariateNormality
# Multivariate Normality Test
cat("Multivariate Normality Test: \n")
mvn_result$multivariateNormality

#=======================================
```

```
#   Transformations and Outliers
#=======================================

# transform the data
transform_data = sqrt(mvn_result$newData + 1)

# remove the outliers
mvn_result = mvn(data = transform_data, multivariateOutlierMethod = "quan", showNewData = TRUE)

par(mfrow=c(1,2))
# test if made changes have resulted in multivariave normality
res_f = mvn(data = mvn_result$newData, multivariatePlot = "qq", multivariateOutlierMethod = "quan")


# test if made changes have resulted in multivariave normality
cat("Multivariate Normality Test: \n")
res_f$multivariateNormality
```

**Multiple Linear Model using Original variables**

```
#=======================================
#             Linear Model
#=======================================

#Fit a linear model for the response variable- happiness score
multlm = lm(formula = Ladder ~., data = data.frame(mvn_result$newData))


#=======================================
#             Model Assumtions
#=======================================

par(mfrow=c(2,2))

# Check whether the assumptions of this model are satisfied.
plot(multlm)


# Output VIF table, to check multi colinearity
cat("VIF Scores are: \n")
round(vif(multlm),3)

# fit the model with all the variables except LogGDP
multlm.new = lm(formula = Ladder ~. -LogGDP , data = data.frame(mvn_result$newData))
par(mfrow=c(2,2))
# Check whether the assumptions of this model are satisfied.
plot(multlm.new)


# Report the `Adjusted R-squared'
multlm.summary = summary(multlm.new)
cat("Adjusted R Squared is: ")
multlm.summary$adj.r.squared
```

**Multiple Linear Model using Principal Components**

```
#==============================
#       Data Manupilation
#==============================

X= mvn_result$newData[,2:10]      # set X as our transformed data
n = nrow(X)

#==============================
#       Cov vs Corr Matrix
#==============================

# sample covariance matrix of X:

S= cov(X)
S
round(S, 3)
# the scales are quite different, suggesting that we may be better off using R (Corrolation Matrix) to


# obtain correlation matrix
R = cor(X)

#==============================
#        Sample PCs entries
#==============================
Z=X

# Obtain the sample mean vector
x.bar = apply(X,2,mean)


# now fill in the entries by calculating sample PCs
for(i in 1:9){
Z[,i] = (X[,i]-x.bar[i])/sqrt(diag(S)[i])
}

#==================================
#    Eigen Values and Vectors
#==================================

# obtain eigenvalues and eigenvectors of R
Val.new = eigen(R)$values


Vec.new = eigen(R)$vectors
rownames(Vec.new) = colnames(X)
colnames(Vec.new) = c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6", "PC7", "PC8", "PC9")



#==============================
#        Sample PCs Values
```

```r
#============================

# Obtain sample PC values:

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:n){
    W.new[j,i] = Vec.new[,i] %*% Z[j,]   # no need to center when using normalized PCCs
    }}




#=======================================
#    How many components should we keep?
#=======================================

# screeplot: Proportion of variation explained by each PC:
screeplot(prcomp(Z),npcs = 9, type = "lines") # suggests keeping the first 4 or 5 PCs.

#============================================================================
# Regression with all standardized PCs as the explanatory variables
#============================================================================

# Fit the model and check which PCs are significant, by checking the P - values
PC.model.new.1 = lm(data.frame(X)$LogGDP ~  W.new[,1] + W.new[,2] + W.new[,3] + W.new[,4] + W.new[,5])
res = summary(PC.model.new.1)
res$coefficients

cat("Adjusted R Squared is: ")
res$adj.r.squared

# Fit a new model, with the correct number of PCs
PC.model.new = lm(data.frame(X)$LogGDP ~  W.new[,1] + W.new[,2] + W.new[,5])
cat("Adjusted R Squared is: ")
summary(PC.model.new)$adj.r.squared



# check the importance of each variable in standardized PCs:

round(Vec.new[,1],3)   # LogGDP, Social, HLE contribute to first principal component more than other va

round(Vec.new[,2],3)   # "Freedom", "Generosity", "Positive" and "gini" contribute to the second princi

round(Vec.new[,5],3)   # Dur dominates the third PC  (perhaps vs stress)

round(Vec.new[,4],3)   # "Generosity", "Corruption" and "gini" contribute to the fifth principal compon
```