Project: Predicting Countries' Happiness Scores

Abstract

As reported in the World Happiness Report (2017), a country's happiness level is more and more being considered a measure of social progress, and in fact is a goal nations increasingly strive towards. The goal of this report is therefore to predict the Happiness Score of a country after taking into account nine other variables measuring aspects such as GDP, Healthy Life Expectancy, Gini index of household income, Corruption, Generosity and Freedom - that is, the data downloaded from the World Happiness Report. In addition to predicting the happiness score, this report also details the process of handling this multivariate data - from data cleaning and manipulation, to Principal Components Analyis and Regression.

Data Manipulation

Before we can do inference, we must first explore the data to ensure that there are no missing values present, check for normality and check for outliers.

There are 32 missing values in the raw data. Assuming that the data is missing completely at random, since we have no reason to assume that this is not the case, the data can then be imputed using the missMDA package. This package imputes after taking into account the relationship between variables.

To check for normality, as advised in Wichern and Johson (2013), squared generalized distance should be computed and plotted, thereby producing a chi-square plot. Wichern and Johnson also mention checking the marginal normality of variables. Since in linear regression, the predictors are treated as fixed, we will not transform them. The response variable, however, is normal according to the Shapiro-Wilk marginal check and histogram. The chi-square plot, and the d^2 values, have also been used to detect multivariate outliers.

Creating this plot reveals a reasonably normal QQ-plot, save for the top few points which are varying. If roughly half of our computed squared generalized distances fall within $q_{c,p}\,(0.50)$, our distribution can said to be normal (Wichern and Johsnon, 2013). When initially calculated with all the observations retained, we see that 54% of our data is within this region. Although this is close to the desired 50%, we can still drop some of the extreme outliers identified by the plot, so that the normality assumption is not suspect.

To identify the most outlying observations, we can assess both the Chi-Square plot produced, and the d^2 values calculated. As is evident by the plot, the top three outliers visually appear to be Zambia, Yemen and Venezuela. Moreover, as explained on page 191, Wichern and Johnson(2013) use the critical value at $\alpha =$ 0.005 to identify the observations that are most outlying. Using similiar logic, I chose the critical value at $\alpha=$ 0.01, which is 23.20925. This means that, at 10 degrees of freedom, 0.01 of the area under the chi distribution lies to the right of this critical value. Looking at the d^2 values reveals that Somalia, Haiti and Indonesia have distances higher than the critical value. Therefore, we can count these ase outliers and remove them from the data to better adhere to the normality assumption.

The QQ-plot produced after the removal of these outliers adheres more closely to the qq-line than earlier. In addition, roughly half (around 52%) of the d^2 values lie in $q_{c,p}\,(0.50)$. Therefore, we can state that our normality assumption is satisfied.

It should be noted that the revised QQ-plot does contain a few points that visually appear to be outliers. Two of these are Zambia and Yemen, which were visually outlying in the initial QQ-plot also. However, removing them does not greatly impact the QQ-plot or change the proportion of d^2 that are less than or equal to $q_{c,p}\,(0.50)$. Since dropping observations could result in the loss of valuable information, we should

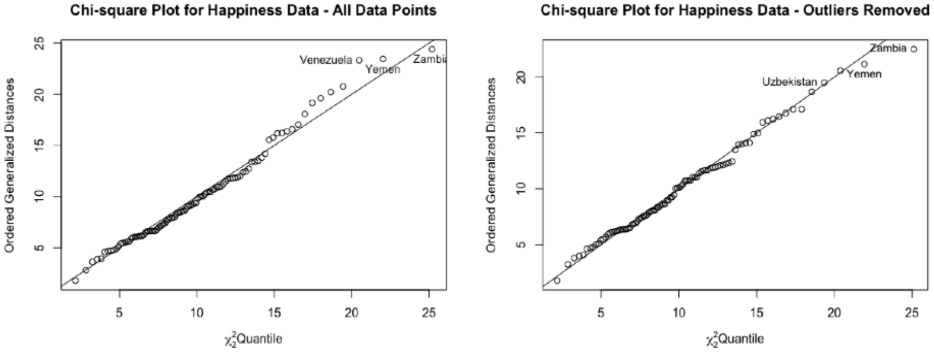


Figure 1: ChiSquare plots before and after Outlier removal

exercise extreme caution. As well, since dropping these observations doesn't greatly impact our assumption, we will leave them in.

Multiple Linear Model Using Original Variables

The Multiple Linear Regression model that we obtain is as follows:

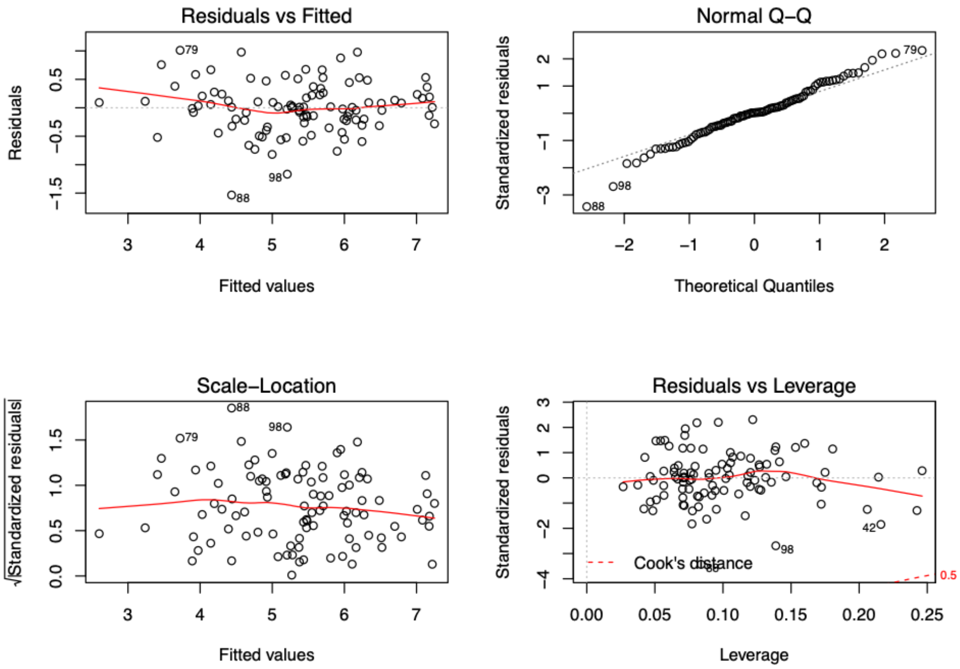
$$Y = -2.19 + 0.35B_1 + 1.63B_2 + 0.02B_3 + \ 1.55B_4 + 0.80_5 - 0.77B_6 + 2.17B_7 + \ 1.20B_8 - 1.25B_9 + \epsilon$$

The Adjusted R^2 value obtained for this model is 0.8185. Before deciding whether or not this value indicates good fit, we must first investigate if the assumptions for linear regression are satisfied. The assumptions are as follows: 1. Linearity 2. Independence of Residuals 3. Equal Variance of Residuals 4. Normal Distribution of Residuals

Plotting the residuals of the model against the fitted values reveals no distinct patterns. For example, we do not see a funnel shape pattern, which suggests equal variances or homoskedacity. We also do not see a linear pattern, and thus the linearity assumption is satisfied also.

Secondly, a histogram of the residuals appears reasonably normal. To further verify this assumption of normality, a QQ-plot can be employed, which will also enable us to detect unusual observations. The QQ-plot produced tapers off at the tails. However, given that it is a small number of points that taper off, the normality assumption should still be satisfied. However, just to ensure that this is true, the Shapiro Wilk test can be applied on the residuals. The p-value calculated is not significant at 0.07869, and therefore our assumption of normality is satisfied.

We can also check the data for multicollinearity. The two highest values are 4.70 for LogGDP and HLE for 4.31, which suggests that the data is not very highly correlated, but we should be wary of this.



Multiple Linear Model Using Principal Components

In order to avoid multicollinearity, the Principal Components can be used for a linear regression model instead of the original variables.

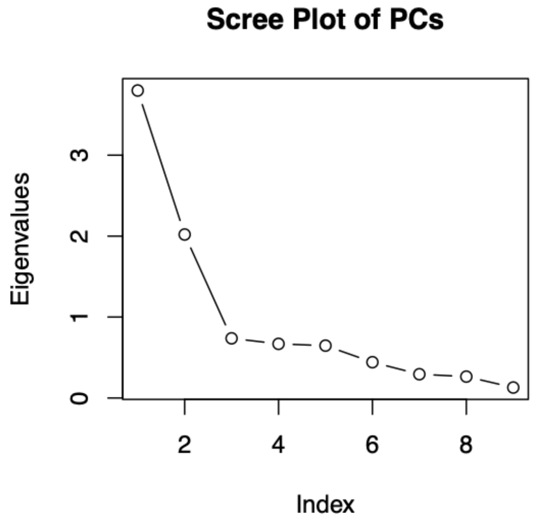
After calculating the Principal Components for this data, fitting a linear model using these Components, and then dropping those that were not siginificant, the model obtained is as follows:

$$\text{Model 2:} \quad Y = 5.39922 + 0.48681B_1 - 0.19684B_2 - 0.21789B_3 + \epsilon$$

Three out of the total nine Principal Components have been used to form this model: PC1, PC4 and PC5. The other six were detected as inisignificant in the summary of the full model and thus not included. Further, the Adjusted R^2 obtained for this model is 0.8185.

To calculate the Principal Components, the Correlation matrix was used instead of the Covariance matrix. This is because of the difference in units in the original variables. For example, the units for Healthy Life Expectancy, are quite different from the units for Generosity. Further, the correlation matrix of the data is equal to the covariance matrix of the standardized observations. Standardization of the measurements ensures that they have equal weight in the analysis, and as explained in the textbook, this is especially important when the units are quite different.

Moreover, in order to decide how many of th 9 principal components should be retained, a scree plot was created, and the proportion of variance explained (PVE) by each component was calculated. As depicted in the scree plot, 2-5 components should be retained. The PVE values reveal that 42.5% of the sample variation in the data is explained by the first component, 22.6% by the second, 8.5% by the third, 7.3% by the fourth, and 6.8% by the sixth. The total percent variation explained by all of these is roughly 88%, and we want the components to account for 70%-90% of variation. Thus, retaining 2-5 components is appropriate.



As stated in Wicham and Johnson(2013), the component coefficients and the correlations should both be examined when interpreting principal components. Looking at the correlations for Component 1 and 5 (which were the ones retained in our linear model), it is evident that, except for 'Generosity' all variables contribute to the first Principal Component. Further, the fourth component is about 'Negative' and 'Generosity', and the fifth is about 'Generosity', 'Negative' and 'Corruption'. We could also look at the correlation between the Components and the Variables.

Appendix

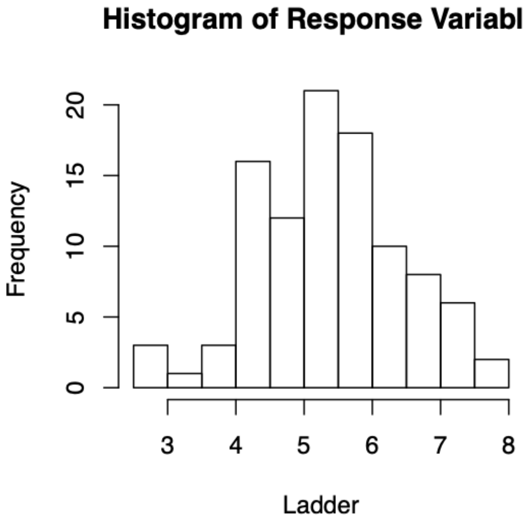
```
##NOTE: Code has been personalized with my initials - hn

############################ PART 1:DATA MANIPULATION #############################

####### Select random sample of 100 #########
set.seed(7736)

myDAT.hn <- sample(RAWDAT.hn$country,100)

#Data frame of random sample of 100 countries:
DAT.hn <- RAWDAT.hn %>% filter(country %in% myDAT.hn)

#check if the filter function worked as we wanted it to:
check.overlap1.hn = setdiff(myDAT.hn,DAT.hn$country)
check.overlap2.hn =setdiff(DAT.hn$country,myDAT.hn) #Yes, it worked

#Make format more convenient to deal with:
rownames(DAT.hn) <- DAT.hn$country
DAT.hn <- DAT.hn[,-which(colnames(DAT.hn)=='country')]

#Summary Stats:
#str(DAT.hn)
```

```
hist(DAT.hn$Ladder,main="Histogram of Response Variable",xlab="Ladder")
#Response variable ladder is normally distributed

## Missing Values ##
total.missing.hn = sum(is.na(DAT.hn))  #32 missing values

#Missing value imputation using MissMDA library
ncpdim.hn <- estim_ncpPCA(DAT.hn)
imputedDAT.hn <- imputePCA(DAT.hn, ncp = ncpdim.hn$ncp)

#This is the imputed DF we will work with:
DAT.hn <- as.data.frame(imputedDAT.hn$completeObs)

#Double Check that everything is as it should be by checking summary stats:

sumstat.head.hn = head(DAT.hn)
#sumstat.str.hn = str(DAT.hn)
sumstat.dim.hn =dim(DAT.hn)
sum.missing.hn = sum(is.na(DAT.hn)) #should see 0
#All good - we are now ready to move forward

#######  Assessing the Assumption of Normality #######

#marginal test for response variable:
DAT.hn <- as.data.frame(DAT.hn)
shapiro.test(DAT.hn$Ladder)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  DAT.hn$Ladder
## W = 0.98697, p-value = 0.4356
```

```
#pvalue:0.4356, therefore normal

##We can plot the pairs of variables:
#scatter.normality.hn <- pairs(DAT.hn)
#however,hard to see all the points so we will use the ChiSq plot instead
#Chisq plot is also the recommended multivariate test of normality + outliers

#Sample Mean and Covariance Matrix:
x.bar.hn <- apply(DAT.hn,2,mean)
sigma.hn <- as.matrix(cov(DAT.hn))

#Generalized Square Distance
DAT.hn <- as.matrix(DAT.hn)
sigma.hn <- as.matrix(sigma.hn)

diff <- DAT.hn - matrix(x.bar.hn,nrow=nrow(DAT.hn),ncol=ncol(DAT.hn),byrow=TRUE)

d2.hn <- diag(diff %*% solve(sigma.hn) %*% t(diff)) #These are D^2 Values

#Obtain quantiles:
quant.hn <- qchisq(((1:nrow(DAT.hn))-0.5)/nrow(DAT.hn), df=10)
```

```
#Create Chi Square QQ Plot:
#chisq.plot.hn <- plot(quant.hn, sort(d2.hn),
                      #xlab = expression(paste(chi[2]^2, "Quantile")),
                      #ylab = "Ordered Generalized Distances",
                      #main = "Chi-square Plot for Happiness Data")
#abline(a=0, b=1)

#Label outlyinging points by running following function and clicking on a point to label:
label.pts.hn = identify(quant.hn,sort(d2.hn),labels = rownames(DAT.hn),n=1)
```



```
#The plot looks reasonably normal, other than three points,
#identified to be: Zambia. Yemen, and Venezuela.

half.dist.hn = qchisq(.50,10)
prop.half.dist.hn = length(d2.hn[which(d2.hn < qchisq(.50,10))])
percentInsideHalfDist <- prop.half.dist.hn/nrow(DAT.hn)

#around 50% of our data should be within the 50% probability contour. Right now it is 54%
#Normality assumption may be suspect here. Therefore we shall drop extreme outliers

#Critical value used to drop Outliers:
crit.val.hn = qchisq(.99,10)
#Compare this value with the d^2 value. if d^2 > crit.val.hn, then drop corresponding obs.

drop <- c("Haiti","Somalia","Indonesia")
DAT.hn <- DAT.hn[-which(rownames(DAT.hn) %in% drop),]

#Check marginal probability of response variable to ensure normality:
DAT.hn <- as.data.frame(DAT.hn)
#shapiro.test(DAT.hn$Ladder) # p-value = 0.4651 therefore normal

#repeat chi-sq plot process:
```

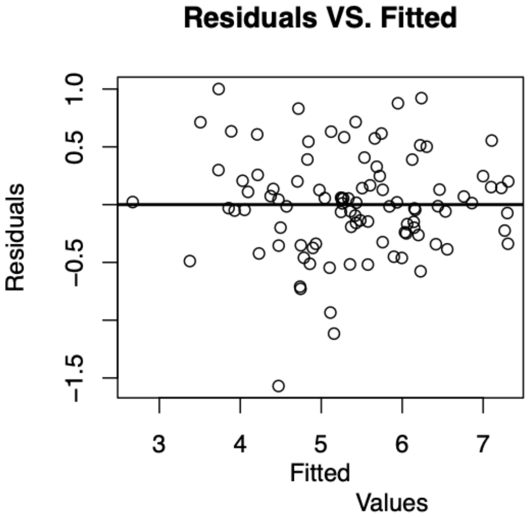
```
x.bar.hn <- apply(DAT.hn,2,mean)
sigma.hn <- as.matrix(cov(DAT.hn))
#Generalized Square Distance
DAT.hn <- as.matrix(DAT.hn)
sigma.hn <- as.matrix(sigma.hn)

diff <- DAT.hn - matrix(x.bar.hn,nrow=nrow(DAT.hn),ncol=ncol(DAT.hn),byrow=TRUE)

d2.hn2 <- diag(diff %*% solve(sigma.hn) %*% t(diff)) #These are D^2 Values

#Obtain quantiles:
quant.hn <- qchisq(((1:nrow(DAT.hn))-0.5)/nrow(DAT.hn), df=10)

#Create Chi Square QQ Plot:

#chisq.plot.hn <- plot(quant.hn, sort(d2.hn2),
                     #xlab = expression(paste(chi[2]^2, "Quantile")),
                     #ylab = "Ordered Generalized Distances",
                     #main = "Chi-square Plot for Happiness Data")
#abline(a=0, b=1)
#identify(quant.hn,sort(d2.hn2),labels = rownames(DAT.hn),n=1)

prop.half.dist2.hn <- length(d2.hn2[which(d2.hn2 < qchisq(.50,10))])


############################ PART 2: MLR with Original Vars ############################

DAT.hn <- as.data.frame(DAT.hn)
mod.hn <- lm(DAT.hn$Ladder ~ DAT.hn$LogGDP +
               DAT.hn$Social + DAT.hn$HLE + DAT.hn$Freedom+
               DAT.hn$Generosity + DAT.hn$Corruption +
               DAT.hn$Positive + DAT.hn$Negative + DAT.hn$gini)

summary(mod.hn)
```

```
##
## Call:
## lm(formula = DAT.hn$Ladder ~ DAT.hn$LogGDP + DAT.hn$Social +
##     DAT.hn$HLE + DAT.hn$Freedom + DAT.hn$Generosity + DAT.hn$Corruption +
##     DAT.hn$Positive + DAT.hn$Negative + DAT.hn$gini)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56959 -0.24877  0.01064  0.20612  1.00015
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.18914    0.87061  -2.514 0.013760 *
## DAT.hn$LogGDP      0.35363    0.08939   3.956 0.000155 ***
## DAT.hn$Social      1.62849    0.61068   2.667 0.009133 **
## DAT.hn$HLE         0.01820    0.01241   1.467 0.146034
## DAT.hn$Freedom     1.54886    0.53805   2.879 0.005025 **
## DAT.hn$Generosity  0.80287    0.40520   1.981 0.050705 .
## DAT.hn$Corruption -0.76803    0.38219  -2.010 0.047577 *
## DAT.hn$Positive    2.17374    0.68411   3.177 0.002057 **
## DAT.hn$Negative    1.20095    0.71175   1.687 0.095126 .
## DAT.hn$gini       -1.25454    0.62461  -2.009 0.047689 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4602 on 87 degrees of freedom
## Multiple R-squared:  0.8355, Adjusted R-squared:  0.8185
## F-statistic:  49.1 on 9 and 87 DF,  p-value: < 2.2e-16
```

```
#rounded coefficients:
rounded.coef.hn = round(coef(mod.hn),2)

#Multicollinearity:
vif(mod.hn)
```

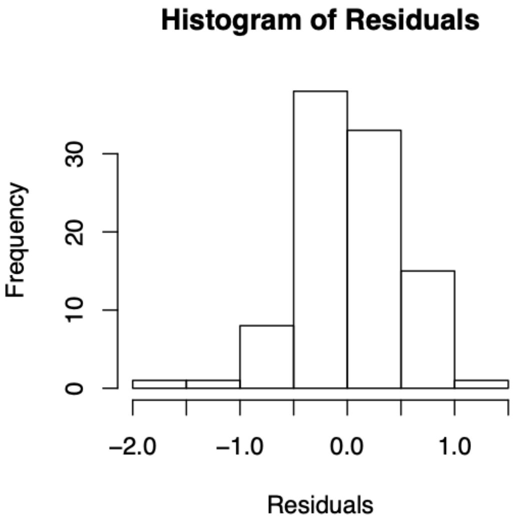
```
##     DAT.hn$LogGDP     DAT.hn$Social        DAT.hn$HLE    DAT.hn$Freedom
##          4.838356          2.827582          4.540402          2.123877
## DAT.hn$Generosity DAT.hn$Corruption   DAT.hn$Positive   DAT.hn$Negative
##          1.370443          1.578114          2.065065          1.731290
##       DAT.hn$gini
##          1.988819
```

```
#Adjusted R^2: 0.8185

###Residuals VS Fitted Values plot for identify equal variance assumption:
resid.plot.hn = plot(fitted(mod.hn),resid(mod.hn),
                     main="Residuals VS. Fitted",xlab="Fitted
                     Values",ylab="Residuals")+abline(h = 0, lwd = 2)
```



```
#No distinct patter, therefore homoskedacity of residuals

#Histograms and QQ-plots to assess Normality Assumption:
```

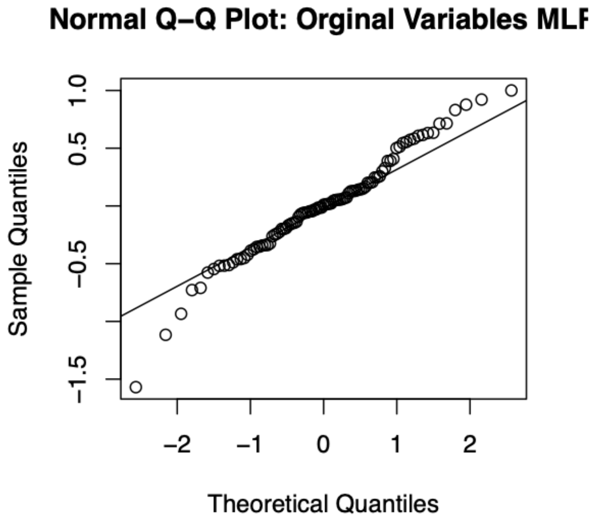
```
hist.resid.hn = hist(resid(mod.hn),main="Histogram of Residuals",xlab="Residuals")
```



```
#Looks reasonably normal

#QQ-Plot to confirm normality:
#Little tapering at tails but only a few points
#To ensure normality assumption is met we can
#confirm using the Shapiro Wilk test on the residuals:
shapiro.test(resid(mod.hn))
```

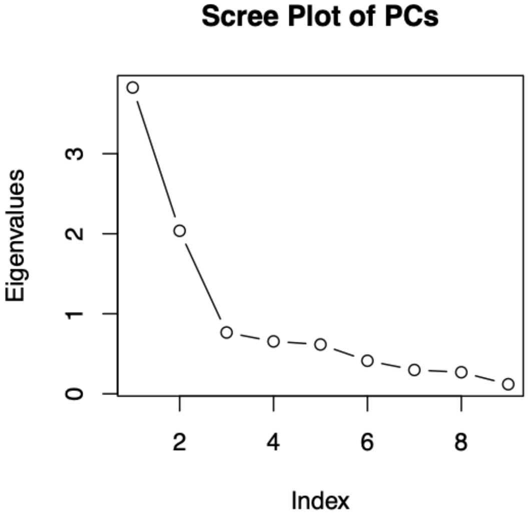
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(mod.hn)
## W = 0.97648, p-value = 0.07869
```

```
#P-value:0.07869 -  assumption of normality satisfied!

qqnorm(resid(mod.hn), main = "Normal Q-Q Plot: Orginal Variables MLR")
qqline(resid(mod.hn))
```



```
################################ PART 3: MLR with PC #############################


#### First we need to obtain the Principal Components ####

#Obtain matrix without the Response variable
DAT.PC <- as.matrix(DAT.hn[,2:10])

std.hn <- apply(DAT.PC,2,sd)
xbar.hn <- apply(DAT.PC, 2, mean)


#We can standardized the matrix. This will help us decide if the Correlation matrix is correct.
Z.PC <- DAT.PC

for(i in 1:ncol(Z.PC)){
  Z.PC[,i] <- (DAT.PC[,i] - xbar.hn[i])/std.hn[i]
}

R.hn = cor(DAT.PC)
sigma = cov(Z.PC)
#These two matrices are the same.
#This is helpful to check that our calculations are correct


#obtain eigenvectors and values
eigenmat.hn <- eigen(R.hn)
eigenvals.hn <- eigenmat.hn$values
eigenvecs.hn <- eigenmat.hn$vectors

colnames(eigenvecs.hn) <- c("PC1","PC2","PC3","PC4","PC5","PC6","PC7","PC8","PC9")
rownames(eigenvecs.hn) = colnames(DAT.PC)
```

```
#^this is our matrix of eigenvectors

#Now we want to actually calculate the sample PCs:

PCvals.hn <- DAT.PC
colnames(PCvals.hn) = c("PC1","PC2","PC3","PC4","PC5","PC6","PC7","PC8","PC9")

for(i in 1:ncol(PCvals.hn)){
  for(j in 1:nrow(DAT.PC)){
    PCvals.hn[j,i] = eigenvecs.hn[,i] %*% Z.PC[j,]
  }}

#So now we should have our PC values

#How many PC's should we retain?
#screeplot: We can retain 2 to 5 PCs

scree.plot.hn = plot(eigenvals.hn, type="b",main="Scree Plot of PCs",ylab="Eigenvalues")
```



```
PVE <- round(eigenvals.hn/sum(eigenvals.hn),3)
PVE
```

```
## [1] 0.425 0.226 0.085 0.073 0.068 0.046 0.033 0.030 0.013
```

```
#First PC explains 42.5% of the variance, and PC2 explains 22.6%


######## Now we can do our Principal Components Regression #######

PC.model.hn = lm(DAT.hn$Ladder ~  PCvals.hn[,1] + PCvals.hn[,2] +
                   PCvals.hn[,3] + PCvals.hn[,4] + PCvals.hn[,5]+
                   PCvals.hn[,6] + PCvals.hn[,7] + PCvals.hn[,8] + PCvals.hn[,9])
summary(PC.model.hn)
```

```
##
## Call:
## lm(formula = DAT.hn$Ladder ~ PCvals.hn[, 1] + PCvals.hn[, 2] +
##     PCvals.hn[, 3] + PCvals.hn[, 4] + PCvals.hn[, 5] + PCvals.hn[,
##     6] + PCvals.hn[, 7] + PCvals.hn[, 8] + PCvals.hn[, 9])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.56959 -0.24877  0.01064  0.20612  1.00015
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.39922    0.04673 115.547  < 2e-16 ***
## PCvals.hn[, 1]  0.48681    0.02401  20.275  < 2e-16 ***
## PCvals.hn[, 2]  0.05188    0.03291   1.576 0.118548
## PCvals.hn[, 3]  0.01555    0.05366   0.290 0.772678
## PCvals.hn[, 4] -0.19684    0.05805  -3.391 0.001051 **
## PCvals.hn[, 5] -0.21789    0.05985  -3.641 0.000461 ***
## PCvals.hn[, 6]  0.04951    0.07310   0.677 0.500039
## PCvals.hn[, 7] -0.06425    0.08606  -0.747 0.457355
## PCvals.hn[, 8]  0.06664    0.09054   0.736 0.463739
## PCvals.hn[, 9]  0.18745    0.13581   1.380 0.171047
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4602 on 87 degrees of freedom
## Multiple R-squared:  0.8355, Adjusted R-squared:  0.8185
## F-statistic:  49.1 on 9 and 87 DF,  p-value: < 2.2e-16
```

```
# Adjusted R-squared:  0.8247 ;  p-value: < 2.2e-16

# Significant PCs: PC1, PC4 and PC5, so let's drop the rest:
PC.model2.hn = lm(DAT.hn$Ladder ~  PCvals.hn[,1] + PCvals.hn[,4] + PCvals.hn[,5])
summary(PC.model2.hn)
```

```
##
## Call:
## lm(formula = DAT.hn$Ladder ~ PCvals.hn[, 1] + PCvals.hn[, 4] +
##     PCvals.hn[, 5])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.59958 -0.24379 -0.00836  0.30383  1.14552
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.39922    0.04674  115.53  < 2e-16 ***
## PCvals.hn[, 1]  0.48681    0.02401   20.27  < 2e-16 ***
## PCvals.hn[, 4] -0.19684    0.05806   -3.39 0.001027 **
## PCvals.hn[, 5] -0.21789    0.05986   -3.64 0.000448 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4603 on 93 degrees of freedom
## Multiple R-squared:  0.8241, Adjusted R-squared:  0.8184
## F-statistic: 145.2 on 3 and 93 DF,  p-value: < 2.2e-16
```

```
#Adjusted R-squared:  0.8184 ; p-value: < 2.2e-16


#PCs included in model:

round(eigenvecs.hn[,1],3) #all vars contribute to the first PC except for Generosity
```

```
##     LogGDP     Social        HLE    Freedom Generosity Corruption   Positive
##      0.438      0.422      0.429      0.313      0.060     -0.269      0.298
##   Negative       gini
##     -0.364     -0.227
```

```
round(eigenvecs.hn[,4],3) #Negative is the major contributor, and then Generosity
```

```
##     LogGDP     Social        HLE    Freedom Generosity Corruption   Positive
##     -0.323      0.221     -0.270     -0.206      0.474      0.167     -0.138
##   Negative       gini
##     -0.633     -0.245
```

```
round(eigenvecs.hn[,5],3) #Generosity  followed by Negative
```

```
##     LogGDP     Social        HLE    Freedom Generosity Corruption   Positive
##     -0.185     -0.197     -0.256      0.044     -0.634     -0.428      0.054
##   Negative       gini
##     -0.513      0.086
```

```
#Remaining PCs:
r2.hn = round(eigenvecs.hn[,2],3) #Generosity, Gini and Freedom are most important here
r3.hn = round(eigenvecs.hn[,3],3) #Corruption contributes most, followed by Positive
r6.hn = round(eigenvecs.hn[,6],3) #Freedom and Gini are most important here
r7.hn = round(eigenvecs.hn[,7],3) #Positive and Freedom contribute most
r8.hn = round(eigenvecs.hn[,8],3) #Most of them contribute, with Social dominating
r9.hn = round(eigenvecs.hn[,9],3) #LogGDP and HLE
```