HOMEWORK 6. ISYE 6501

Guillermo de la Hera Casado

September 28th, 2019

Question 9.1. Principal Components Analysis

Exploratory analysis

From the Grubbs test performed in HW3 with this data, we learnt that the value: *1993* could be potentially an outlier with p value = 0.079.

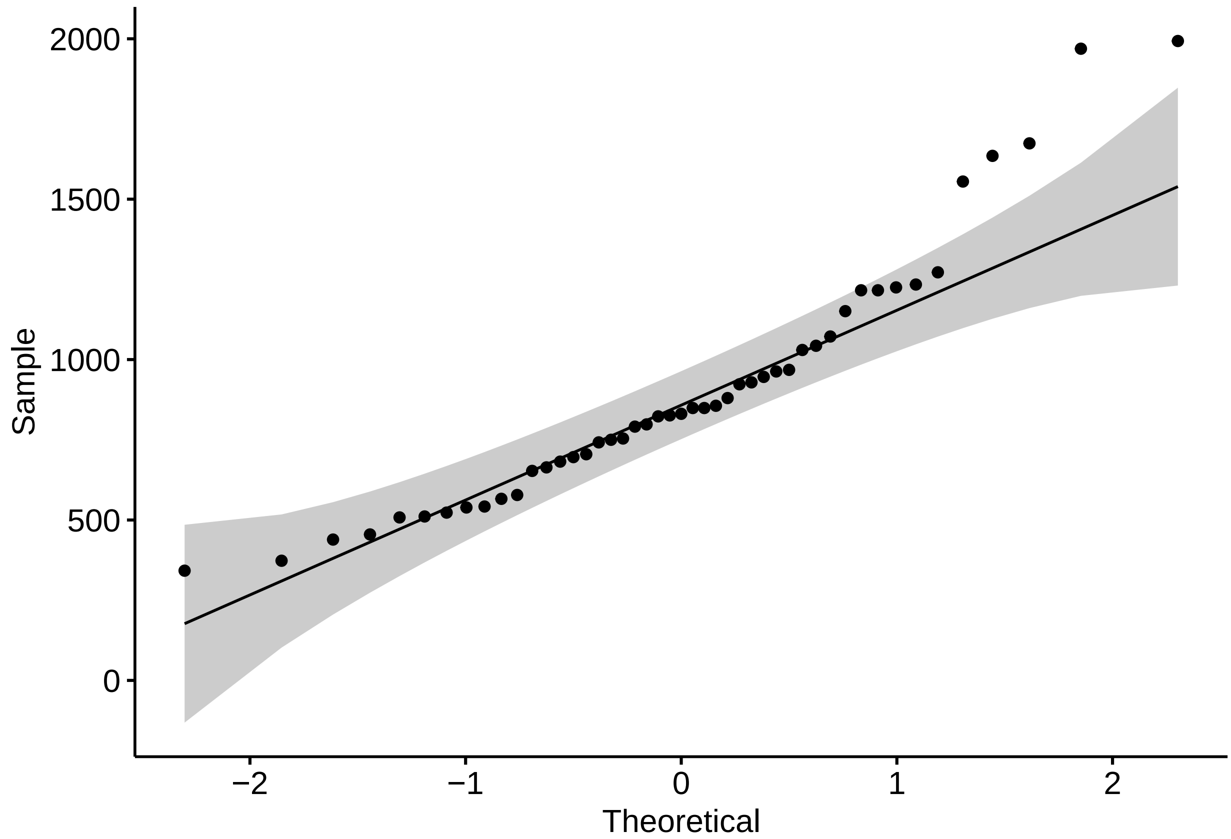
However, since we will use for this homework confidence level = 95%, we cannot reject the null hypothesis [value is not an outlier].

Another check we learnt about is to ensure that the response is normally distributed. Let's apply Q-Q plot to investigate that and see whether a Box-Cox transformation is required:

```
set.seed(101) # Set Seed so that same sample can be reproduced in future also
usCrime <- read.table("uscrime.txt", header = TRUE, sep = "\t")
print(grubbs.test(usCrime$Crime, type = 10, two.sided = FALSE))
```

```
##
##  Grubbs test for one outlier
##
## data:  usCrime$Crime
## G = 2.81287, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1993 is an outlier
```

```
ggqqplot(usCrime$Crime)
```

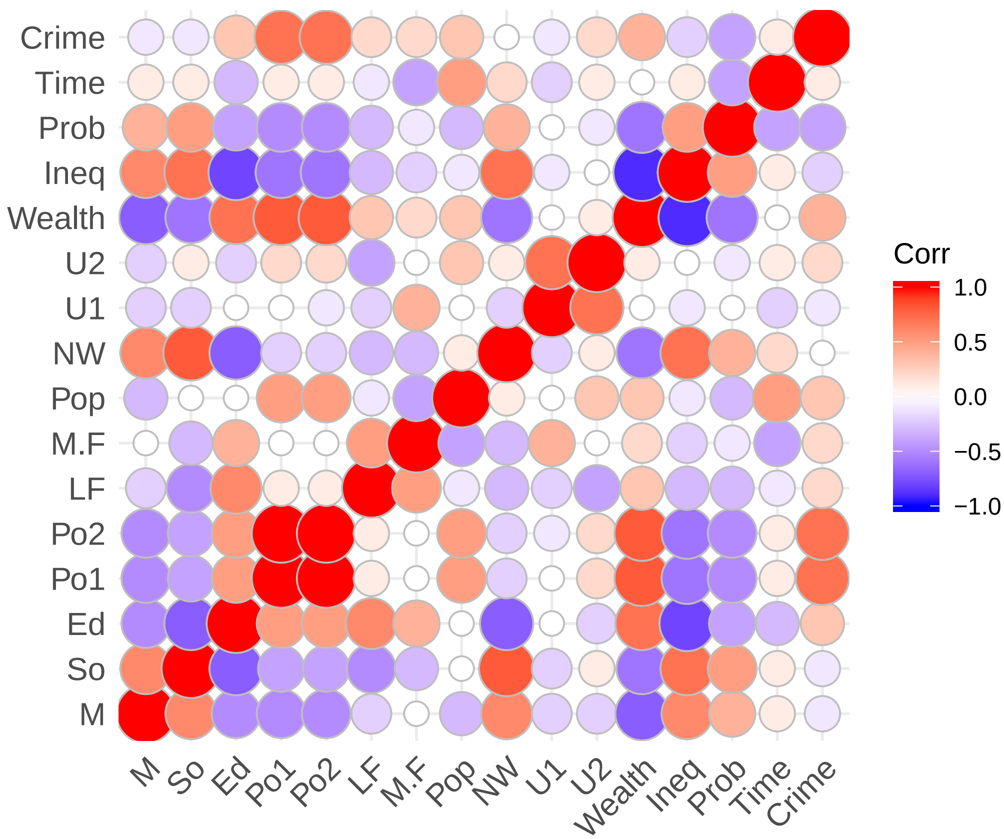


It seems that the majority of data is normally distributed with the exception of the outliers, so we will move ahead without any modification on the data. Before applying PCA, let's investigate the collinearity of the predictors.

From the Correlation Plot we can see the following:

- There is a high positive correlation between Po1 and Po2
- There is a high negative correlation between Wealth and Ineq

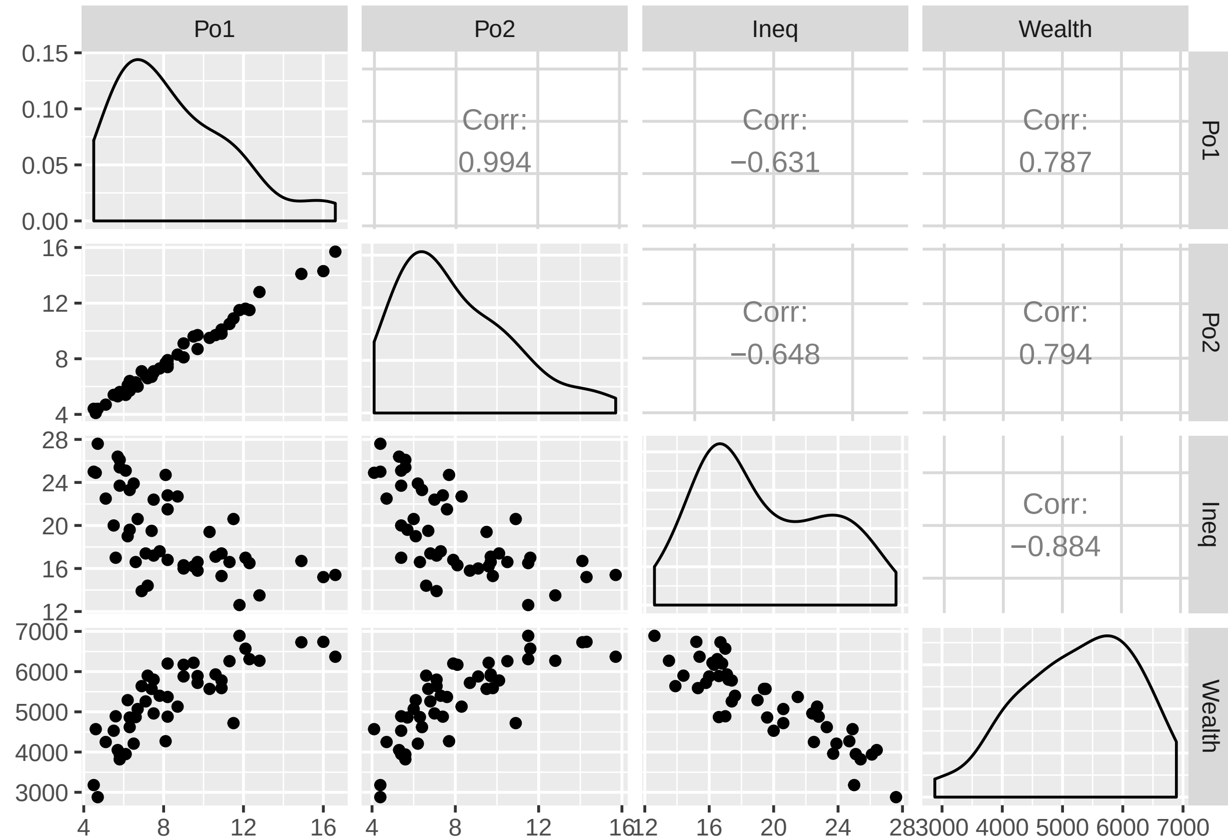
```
corr <- round(cor(usCrime), 1)
ggcorrplot(corr, method = "circle")
```




```
#ggpairs(usCrime)
```

These facts can be confirmed with a ggpairs plot:

```
ggpairs(usCrime, columns = c("Po1","Po2","Ineq", "Wealth"))
```




```
print(head(usCrime[,-c(2,16)]))
```

```
##      M   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4 13.6 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7 0.015801
## 5 14.1 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4 0.041399
## 6 12.1 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6 0.034201
##      Time
## 1 26.2011
## 2 25.2999
## 3 24.3006
## 4 29.9012
## 5 21.2998
## 6 20.9995
```

Apply PCA and create a linear regression model with most relevant Principal Components

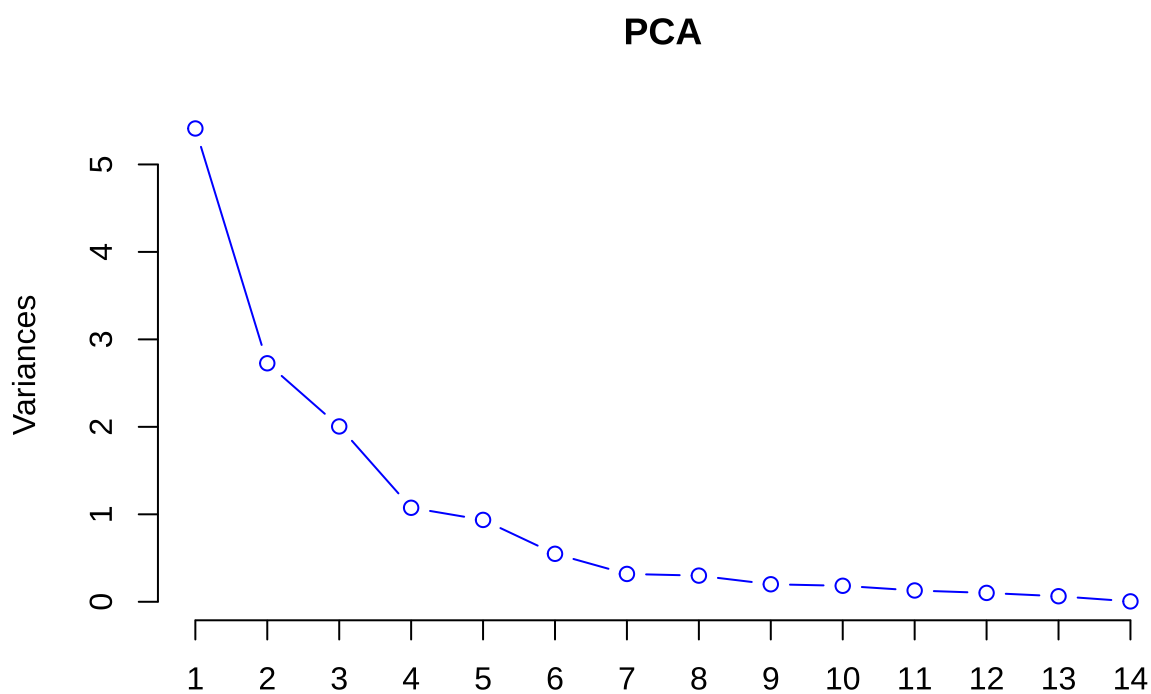
As per stated in office hours, PCA may not work well with binary predictors. Therefore I have removed the second column and the response variable and input the remaining into the PCA model. I have also scaled the data to obtain unit variance:

```
test_point <- data.frame(M = 14.0, Ed = 10.0, Po1 = 12.0, Po2 = 15.5,
                         LF = 0.640, M.F = 94.0, Pop = 150, NW = 1.1,
                         U1 = 0.120, U2 = 3.6, Wealth = 3200, Ineq = 20.1,
```

```r
                          Prob = 0.040, Time = 39.0)
PCA <- prcomp(usCrime[,-c(2, 16)], scale = TRUE, center =TRUE)
test_scaled <- (test_point - PCA$center)/PCA$scale
summary(PCA)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     2.3262 1.6513 1.4158 1.03670 0.96745 0.74049
## Proportion of Variance 0.3865 0.1948 0.1432 0.07677 0.06685 0.03917
## Cumulative Proportion  0.3865 0.5813 0.7244 0.80121 0.86806 0.90723
##                            PC7     PC8    PC9    PC10    PC11    PC12
## Standard deviation     0.56415 0.54675 0.4475 0.42747 0.35945 0.31852
## Proportion of Variance 0.02273 0.02135 0.0143 0.01305 0.00923 0.00725
## Cumulative Proportion  0.92996 0.95132 0.9656 0.97867 0.98790 0.99515
##                           PC13    PC14
## Standard deviation     0.25159 0.06802
## Proportion of Variance 0.00452 0.00033
## Cumulative Proportion  0.99967 1.00000
```

```r
screeplot(PCA, type = "lines", col = "blue",npcs=14)
```



As we can see in the summary (and also confirmed by the Scree Plot):

- the first 4 principal components cover the majority of the variance -> PC1 + PC2 + PC3 + PC4 = 80% cumulative proportion of variance.
- Adding PC5 and PC6 would explain 90% of the variance.

- Adding PC7 to PC14 would provide 10% additional, so not worth it in terms of complexity added.

Trying the model with the 4 most important Principal Components yielded a very low Adjust R Squared result: **0.21** Let's detail here the results if we use **the first 6 Principal Components**:

```
PC <- PCA$x[,1:6]
usCrimePC <- cbind(PC, usCrime[,16])
modelPCA <- lm(V7~., data = as.data.frame(usCrimePC))
summary(modelPCA)
```

```
##
## Call:
## lm(formula = V7 ~ ., data = as.data.frame(usCrimePC))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -399.15 -166.78   15.28  150.91  452.53
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  905.085     36.058  25.101  < 2e-16 ***
## PC1           76.750     15.668   4.898 1.64e-05 ***
## PC2          -57.648     22.072  -2.612   0.0126 *
## PC3           24.313     25.744   0.944   0.3506
## PC4            3.786     35.157   0.108   0.9148
## PC5         -235.831     37.674  -6.260 2.04e-07 ***
## PC6           64.174     49.221   1.304   0.1998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 247.2 on 40 degrees of freedom
## Multiple R-squared:  0.6448, Adjusted R-squared:  0.5915
## F-statistic:  12.1 on 6 and 40 DF,  p-value: 1.036e-07
```

Adjusted R-Squared comes at **0.591**, that is less than:

- what I achieved in Homework 5 by using a linear regression on the training set, with the parameters with <0.1 p-value, so: M, Ed, Ineq, Prob, Po1 and U2 (**0.73**)
- what I achieved in Homework 5 by using cross validation, on parameters with <0.1 p-value, so: M, Ed, Ineq, Prob, Po1, U2 (**0.621**)

The outcome is quite surprising as I expected PCA to deliver a higher Adjusted R-Squared by removing colinearity. The model estimates that PC3, PC4 and PC6 are not statistically significant, with very high p-values. Let's try to adjust the model to only use the following: PC1, PC2, PC5:

```
modelPCA <- lm(V7~PC1+PC2+PC5, data = as.data.frame(usCrimePC))
summary(modelPCA)
```

```
##
## Call:
## lm(formula = V7 ~ PC1 + PC2 + PC5, data = as.data.frame(usCrimePC))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -417.70 -144.59  -19.17  168.14  462.75
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      35.89  25.218  < 2e-16 ***
## PC1            76.75      15.60   4.921 1.31e-05 ***
## PC2           -57.65      21.97  -2.624    0.012 *
## PC5          -235.83      37.50  -6.289 1.39e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 246.1 on 43 degrees of freedom
## Multiple R-squared:  0.6217, Adjusted R-squared:  0.5953
## F-statistic: 23.55 on 3 and 43 DF,  p-value: 3.575e-09
```

```
print(modelPCA$coefficients)
```

```
## (Intercept)         PC1         PC2         PC5
##   905.08511    76.74986   -57.64762  -235.83067
```

The adjusted R-Squared comes quite similar, at **0.595**. Let's then use this model with 3 Principal Components and try to specify it in terms of its original variables, as it's simpler.

Specify the chosen Regression Model in terms of its original variables.

Let's get the Principal Component coefficients explained in terms of original values and make the prediction for our scaled test point:

```
PC1_original <- modelPCA$coefficients[2] %*% PCA$rotation[,1]
PC2_original <- modelPCA$coefficients[3] %*% PCA$rotation[,2]
PC5_original <- modelPCA$coefficients[4] %*% PCA$rotation[,5]
print(PCA$rotation[,1])
```

```
##           M          Ed         Po1         Po2          LF         M.F
## -0.32074046  0.34898975  0.34759618  0.35004235  0.16641342  0.10677158
##         Pop          NW          U1          U2      Wealth        Ineq
##  0.14183140 -0.28869268  0.03516617  0.03246885  0.40799489 -0.38223873
##        Prob        Time
## -0.27281594 -0.01183448
```

```
print(PC1_original)
```

```
##           [,1]     [,2]     [,3]    [,4]     [,5]     [,6]     [,7]
## [1,] -24.61679 26.78492 26.67796 26.8657 12.77221 8.194704 10.88554
##           [,8]     [,9]   [,10]    [,11]     [,12]     [,13]      [,14]
## [1,] -22.15712 2.698999 2.49198 31.31355 -29.33677 -20.93859 -0.9082945
```

```
predicted <- modelPCA$coefficients[1] + rowSums(PC1_original * test_scaled) +
  rowSums(PC2_original * test_scaled) +
  rowSums(PC5_original * test_scaled)
print(predicted)
```

```
## (Intercept)
##    1433.141
```

As we can see, each PC can decomposed on the linear combination of the original variables by using the relevant eigenvector.

The predicted value is: **1433**. If we look at the Crime response distribution, we can see that the value still falls within the upper whisker [Q3, Q3 + 1.5IQR] and it's not considered an outlier.

```
ggplot(data=usCrime, aes(x="", y = usCrime$Crime)) + geom_boxplot()
```