Homework 6

This week was devoted to PCA and its' importance in regression. The following document provides solutions for all the tasks from Homework 6.

Task 9.1

QUESTION: Using the same crime data set uscrime.txt as in Question 8.2, apply Principal Component Analysis and then create a regression model using the first few principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Question 8.2. You can use the R function prcomp for PCA. (Note that to first scale the data, you can include scale. = TRUE to scale as part of the PCA function. Don't forget that, to make a prediction for the new city, you'll need to unscale the coefficients (i.e., do the scaling calculation in reverse)!)

```r
#New city data
prediction_data <- data.frame('M' = 14.0,
'So' = 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,
'Prob' = 0.04,
'Time' = 39.0
)
prediction_data
```

```
##    M So Ed Po1  Po2   LF M.F Pop  NW   U1  U2 Wealth Ineq Prob Time
## 1 14  0 10  12 15.5 0.64  94 150 1.1 0.12 3.6   3200 20.1 0.04   39
```

SOLUTION:

```r
#loading the libraries
library(knitr)
library(readr)
library(tidyverse)
library(plyr)
library(dplyr)
library(ggpubr)
library(stats)
library(fBasics)
library(stats4)
library(AICcmodavg)
library(flexmix)
library(psych)
```

This task involves many statistical libraries such as stats, stats4, fBasics, flexmix and AICcmodavg.

Analysing the data points

```r
#Loading the data from a txt file
data <- read.table("C:/Users/1/Documents/data 5.1/uscrime.txt", header = TRUE)
kable(head(data, 10))
```

M	So	Ed	Po1	Po2	LF	M.F	Pop	NW	U1	U2	Wealth	Ineq	Prob	Time	Crime
15.1	1	9.1	5.8	5.6	0.510	95.0	33	30.1	0.108	4.1	3940	26.1	0.084602	26.2011	791
14.3	0	11.3	10.3	9.5	0.583	101.2	13	10.2	0.096	3.6	5570	19.4	0.029599	25.2999	1635
14.2	1	8.9	4.5	4.4	0.533	96.9	18	21.9	0.094	3.3	3180	25.0	0.083401	24.3006	578
13.6	0	12.1	14.9	14.1	0.577	99.4	157	8.0	0.102	3.9	6730	16.7	0.015801	29.9012	1969
14.1	0	12.1	10.9	10.1	0.591	98.5	18	3.0	0.091	2.0	5780	17.4	0.041399	21.2998	1234
12.1	0	11.0	11.8	11.5	0.547	96.4	25	4.4	0.084	2.9	6890	12.6	0.034201	20.9995	682
12.7	1	11.1	8.2	7.9	0.519	98.2	4	13.9	0.097	3.8	6200	16.8	0.042100	20.6993	963
13.1	1	10.9	11.5	10.9	0.542	96.9	50	17.9	0.079	3.5	4720	20.6	0.040099	24.5988	1555
15.7	1	9.0	6.5	6.2	0.553	95.5	39	28.6	0.081	2.8	4210	23.9	0.071697	29.4001	856
14.0	0	11.8	7.1	6.8	0.632	102.9	7	1.5	0.100	2.4	5260	17.4	0.044498	19.5994	705

We have 15 predictors and 1 response column. This is the description for each column:

Variable	Description
M	percentage of males aged 14вЂ“24 in total state population
So	indicator variable for a southern state
Ed	mean years of schooling of the population aged 25 years or over
Po1	per capita expenditure on police protection in 1960
Po2	per capita expenditure on police protection in 1959
LF	labour force participation rate of civilian urban males in the age-group 14-24
M.F	number of males per 100 females
Pop	state population in 1960 in hundred thousands
NW	percentage of nonwhites in the population
U1	unemployment rate of urban males 14вЂ“24
U2	unemployment rate of urban males 35вЂ“39
Wealth	wealth: median value of transferable assets or family income
Ineq	income inequality: percentage of families earning below half the median income
Prob	probability of imprisonment: ratio of number of commitments to number of offenses
Time	average time in months served by offenders in state prisons before their first release
Crime	crime rate: number of offenses per 100,000 population in 1960

```r
#Checking the number of missing values in every column
sum(is.na(data$Crime))
```

```
## [1] 0
```

There is no any missing values in the data. So it is ready for the analysis.

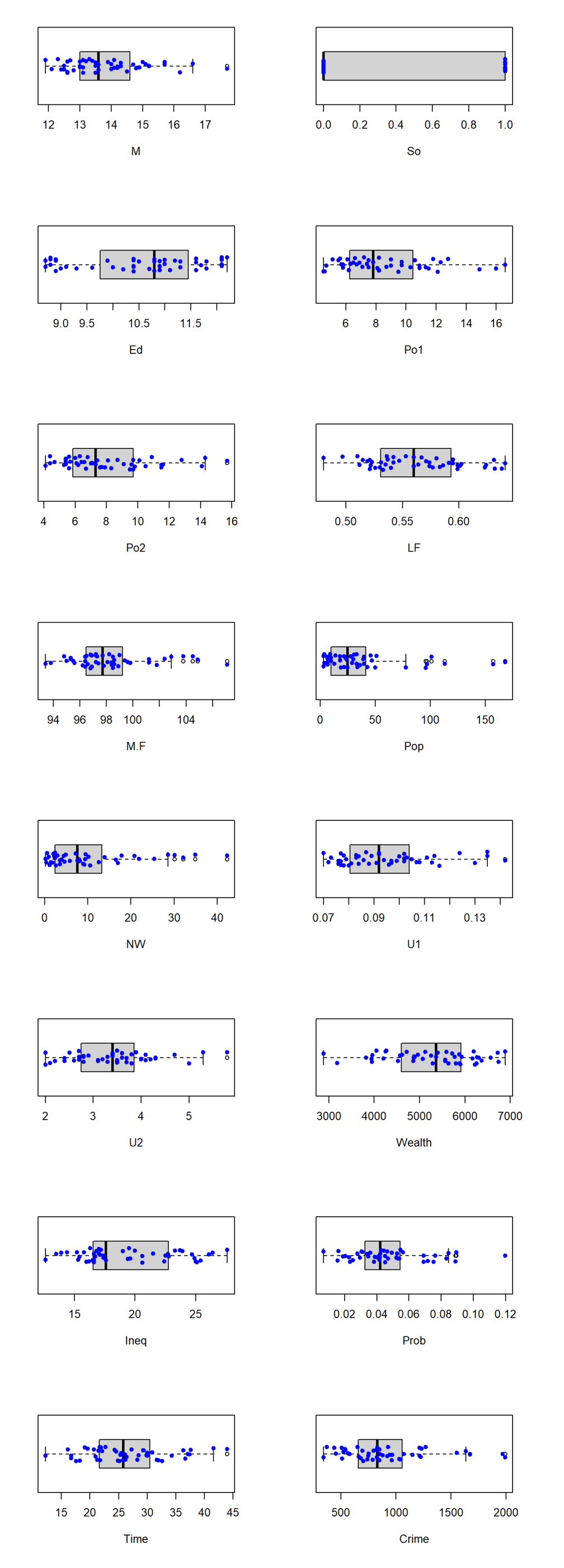
Let's explore the dataset in more details.

```r
#Exploring every variable
statistics <- basicStats(data)
kable(t(statistics[c(3,4,5,6,7,8,13,14),]))
```

	Minimum	Maximum	1. Quartile	3. Quartile	Mean	Median	Variance	Stdev
M	11.9000	17.700000	13.000000	14.60000	13.857447	13.6000	1.579454e+00	1.256763
So	0.0000	1.000000	0.000000	1.00000	0.340426	0.0000	2.294170e-01	0.478975
Ed	8.7000	12.200000	9.750000	11.45000	10.563830	10.8000	1.251489e+00	1.118700
Po1	4.5000	16.600000	6.250000	10.45000	8.500000	7.8000	8.832174e+00	2.971897
Po2	4.1000	15.700000	5.850000	9.70000	8.023404	7.3000	7.818353e+00	2.796132
LF	0.4800	0.641000	0.530500	0.59300	0.561191	0.5600	1.633000e-03	0.040412
M.F	93.4000	107.100000	96.450000	99.20000	98.302128	97.7000	8.683256e+00	2.946737
Pop	3.0000	168.000000	10.000000	41.50000	36.617021	25.0000	1.449415e+03	38.071188
NW	0.2000	42.300000	2.400000	13.25000	10.112766	7.6000	1.057377e+02	10.282882
U1	0.0700	0.142000	0.080500	0.10400	0.095468	0.0920	3.250000e-04	0.018029
U2	2.0000	5.800000	2.750000	3.85000	3.397872	3.4000	7.132560e-01	0.844545
Wealth	2880.0000	6890.000000	4595.000000	5915.00000	5253.829787	5370.0000	9.310502e+05	964.909442
Ineq	12.6000	27.600000	16.550000	22.75000	19.400000	17.6000	1.591696e+01	3.989606
Prob	0.0069	0.119804	0.032701	0.05445	0.047091	0.0421	5.170000e-04	0.022737
Time	12.1996	44.000400	21.600350	30.45075	26.597921	25.8006	5.022408e+01	7.086895
Crime	342.0000	1993.000000	658.500000	1057.50000	905.085106	831.0000	1.495854e+05	386.762697

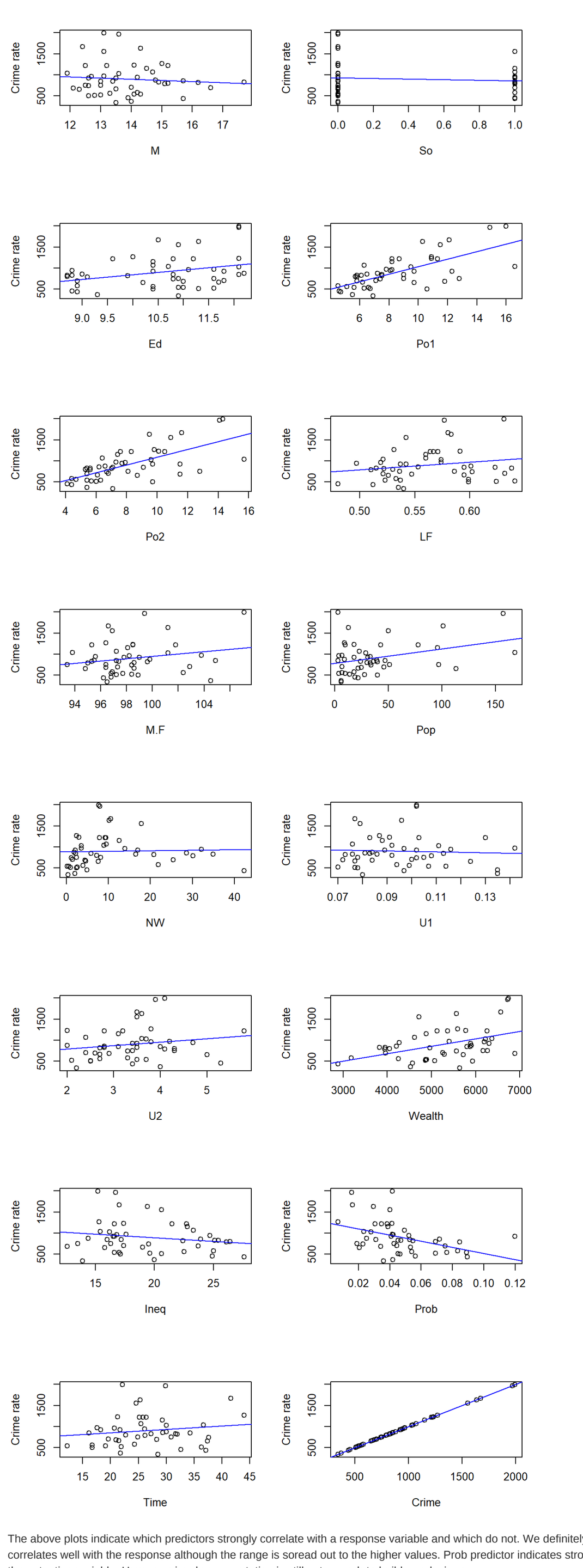
Most of the predictors have a good variability which is sufficient for our analysis. The Crime rate is between 342 and 1993. While predicting the crime rate on our prediction data we should build the crime rate between this two values or at least with a little difference with these maximum and minimum values.

```r
#Plotting every variable in boxplots
par(mfrow = c(2,2))
for (i in 1:ncol(data)) {
  boxplot(data[,i], horizontal = TRUE, xlab = colnames(data)[i])
  stripchart(data[,i], method = 'jitter', pch = 19, add = TRUE, col = 'blue')
}
```



The boxplots show that variables are well distributed without clustering around one range. However there is a So variable with only two values c(1,0) which probably means TRUE and FALSE. Some variables have outliying values however we will consider all of them as a real occasional observations.

```r
#Plotting correlation plots of every predictor with a response
par(mfrow = c(2,2))
for (i in 1:ncol(data)) {
  plot(data[,i], data$Crime, xlab = colnames(data)[i], ylab = 'Crime rate')
  abline(lm(data$Crime ~ data[,i], data = data), col = "blue")
}
```



The above plots indicate which predictors strongly correlate with a response variable and which do not. We definitely see that Po1 and Po2 correlates well with the response although the range is soread out to the higher values. Prob predictor indicates strong negative relationship with the outputing variable. However visual representation is still not enough to build conclusions on.

Fitting the regression model without PCA

Let's discuss how to find a best fit and what is the indicator of it. Best fit minimizes Sum of Squared Errors defined by parameters a0 and a1. In the case of using pca it is more complicated to compare these two models as they are performed on two different datasets. So the comparative analysis will be performed using SSE and R squared. Firstly let's fit our model without using PCA and let's see the results for further comparison. As we saw from the previous analysis we need to fit a model using certain parameters: M, Po1, Ed, U1, U2, Ineq, and Prob. This model performed really well in comparison to when we used all variables.

```r
#Fitting the regression model
better_model <- lm(Crime~M + Po1 + Ed + U1 + U2 + Ineq + Prob,
                   data)
summary(better_model)
```

```
##
## Call:
## lm(formula = Crime ~ M + Po1 + Ed + U1 + U2 + Ineq + Prob, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -520.76 -105.67    9.53  136.28  519.37
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5095.55     896.90  -5.681 1.43e-06 ***
## M             106.78      33.18   3.218   0.0026 **
## Po1           105.96      15.72   6.738 4.91e-08 ***
## Ed            218.45      48.33   4.520 5.62e-05 ***
## U1          -3542.35    3021.94  -1.172   0.2482
## U2            158.82      71.89   2.209   0.0331 *
## Ineq           66.33      13.92   4.767 2.61e-05 ***
## Prob        -3730.85    1522.21  -2.451   0.0188 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 199.8 on 39 degrees of freedom
## Multiple R-squared:  0.7738, Adjusted R-squared:  0.7332
## F-statistic: 19.06 on 7 and 39 DF,  p-value: 8.805e-11
```

Let's see how this same model predicts the response on our prediction_data:

```r
#Find the prediction response
predict(better_model, prediction_data)
```

```
##        1
## 1186.075
```

Response is in the range of our observed responses and even is close to the upper quartile. Let's see what will happen after performing PCA.

Fitting regression model using PCA

```r
#Performing Principal component analysis
my_pca <- prcomp(data[,1:15], scale = TRUE)
summary(my_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377 0.56729
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688 0.02145
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996 0.92142
##                            PC8     PC9    PC10    PC11    PC12    PC13   PC14
## Standard deviation     0.55444 0.48493 0.44708 0.41915 0.35804 0.26333 0.2418
## Proportion of Variance 0.02049 0.01568 0.01333 0.01171 0.00855 0.00462 0.0039
## Cumulative Proportion  0.94191 0.95759 0.97091 0.98263 0.99117 0.99579 0.9997
##                           PC15
## Standard deviation     0.06793
## Proportion of Variance 0.00031
## Cumulative Proportion  1.00000
```

```r
#Finding the eigenvectors of all principal components
my_pca$rotation
```

```
##                PC1         PC2           PC3         PC4         PC5
## M      -0.30371194  0.06280357  0.1724199946 -0.02035537 -0.35832737
## So     -0.33088129 -0.15837219  0.0155433104  0.29247181 -0.12061130
## Ed      0.33962148  0.21461152  0.0677396249  0.07974375 -0.02442839
## Po1     0.30863412 -0.26981761  0.0506458161  0.33325059 -0.23527680
## Po2     0.31099285 -0.26396300  0.0530651173  0.35192809 -0.20473383
## LF      0.17617757  0.31943042  0.2715301768 -0.14326529 -0.39407588
## M.F     0.11638221  0.39434428 -0.2031621598  0.01048029 -0.57877443
## Pop     0.11307836 -0.46723456  0.0770210971 -0.03210513 -0.08317034
## NW     -0.29358647 -0.22801119  0.0788156621  0.23925971 -0.36079387
## U1      0.04050137  0.00807439 -0.6590290980 -0.18279096 -0.13136873
## U2      0.01812228 -0.27971336 -0.5785006293 -0.06889312 -0.13499487
## Wealth  0.37970331 -0.07718862  0.0100647664  0.11781752  0.01167683
## Ineq   -0.36579778 -0.02752240 -0.0002944563 -0.08066612 -0.21672823
## Prob   -0.25888661  0.15831708 -0.1176726436  0.49303389  0.16562829
## Time   -0.02062867 -0.38014836  0.2235664632 -0.54059002 -0.14764767
##                 PC6         PC7         PC8         PC9        PC10        PC11
## M      -0.449132706 -0.15707378 -0.55367691  0.15474793 -0.01443093  0.39446657
## So     -0.100500743  0.19649727  0.22734157 -0.65599872  0.06141452  0.23397868
## Ed     -0.008571367 -0.23943629 -0.14644678 -0.44326978  0.51887452 -0.11821954
## Po1    -0.095776709  0.08011735  0.04613156  0.19425472 -0.14320978 -0.13042001
## Po2    -0.119524780  0.09518288  0.03168720  0.19512072 -0.05929780 -0.13885912
## LF      0.504234275 -0.15931612  0.25513777  0.14393498  0.03077073  0.38532827
## M.F    -0.074501901  0.15548197 -0.05507254 -0.24378252 -0.35323357 -0.28029732
## Pop     0.547098563  0.09046187 -0.59078221 -0.20244830 -0.03970718  0.05849643
## NW      0.051219538 -0.31154195  0.20432828  0.18984178  0.49201966 -0.20695666
## U1      0.017385981 -0.17354115 -0.20206312  0.02069349  0.22765278 -0.17857891
## U2      0.048155286 -0.07526787  0.24369650  0.05576010 -0.04750100  0.47021842
## Wealth -0.154683104 -0.14859424  0.08630649 -0.23196695 -0.11219383  0.31955631
## Ineq    0.272027031  0.37483032  0.07184018 -0.02494384 -0.01390576 -0.18278697
## Prob    0.283535996 -0.56159383 -0.08598908 -0.05306898 -0.42530006 -0.08978385
## Time   -0.148203050 -0.44199877  0.19507812 -0.23551363 -0.29264326 -0.26363121
##               PC12        PC13        PC14          PC15
## M       0.16580189 -0.05142365  0.04901705  0.0051398012
## So     -0.05753357 -0.29368483 -0.29364512  0.0084369230
## Ed      0.47786536  0.19441949  0.03964277 -0.0280052040
## Po1     0.22611207 -0.18592255 -0.09490151 -0.6894155129
## Po2     0.19088461 -0.13454940 -0.08259642  0.7200270100
## LF      0.02705134 -0.27742957 -0.15385625  0.0336823193
## M.F    -0.23925913  0.31624667 -0.04125321  0.0097922075
## Pop    -0.18350385  0.12651689 -0.05326383  0.0001496323
## NW     -0.36671707  0.22901695  0.13227774 -0.0370783671
## U1     -0.09314897 -0.59039450 -0.02335942  0.0111359325
## U2      0.28440496  0.43292853 -0.03985736  0.0073618948
## Wealth -0.32172821 -0.14077972  0.70031840 -0.0025685109
## Ineq    0.43762828 -0.12181090  0.59279037  0.0177570357
## Prob    0.15567100 -0.03547596  0.04761011  0.0293376260
## Time    0.13536989 -0.05738113 -0.04488401  0.0376754405
```

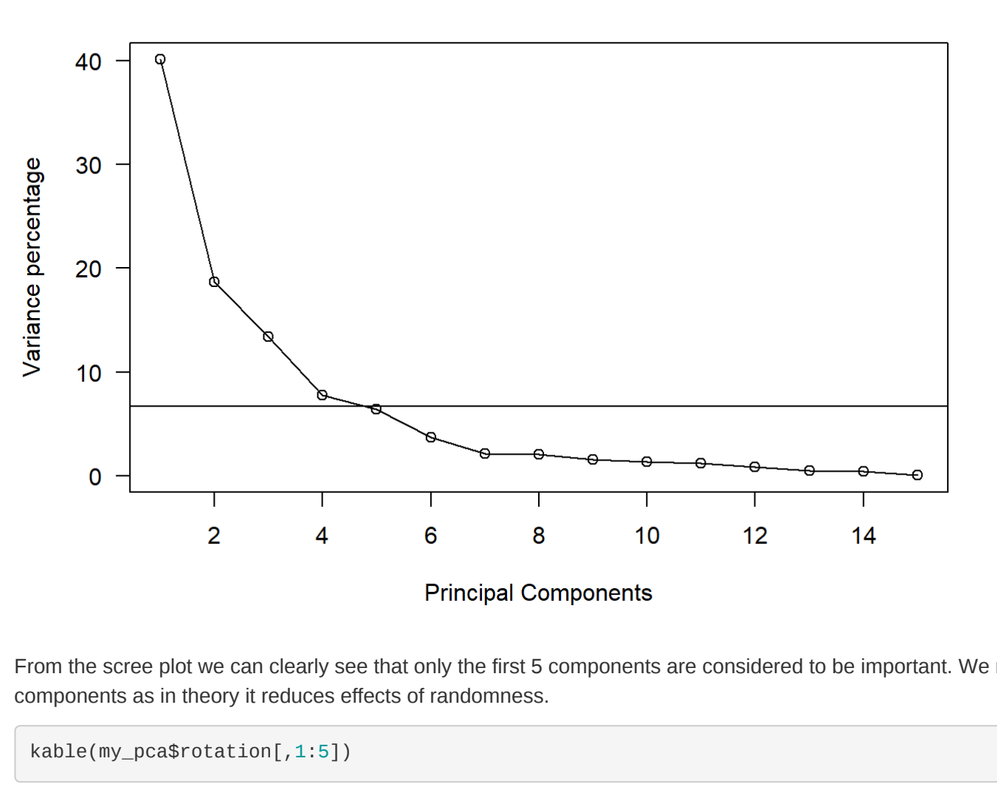
```r
#Let's see the coordinates ranked by importance
summary(my_pca$x)
```

```
##       PC1              PC2                PC3               PC4
##  Min.   :-5.6595   Min.   :-4.76202   Min.   :-3.3514   Min.   :-2.19771
##  1st Qu.:-1.4753   1st Qu.:-1.09557   1st Qu.:-0.8622   1st Qu.:-0.57222
##  Median : 0.4266   Median : 0.05183   Median : 0.2279   Median :-0.05246
##  Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 1.7406   3rd Qu.: 1.25317   3rd Qu.: 1.1088   3rd Qu.: 0.61597
##  Max.   : 4.0566   Max.   : 3.03207   Max.   : 2.3630   Max.   : 3.97985
##       PC5               PC6                PC7                 PC8
##  Min.   :-2.89511   Min.   :-2.4336   Min.   :-1.337729   Min.   :-1.54746
##  1st Qu.:-0.64597   1st Qu.:-0.4682   1st Qu.:-0.409679   1st Qu.:-0.30308
##  Median : 0.04159   Median : 0.0724   Median : 0.009579   Median : 0.02745
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.000000   Mean   : 0.00000
##  3rd Qu.: 0.63264   3rd Qu.: 0.5407   3rd Qu.: 0.264713   3rd Qu.: 0.36137
##  Max.   : 2.11563   Max.   : 1.5231   Max.   : 1.040214   Max.   : 1.05686
##       PC9               PC10               PC11               PC12
##  Min.   :-1.16022   Min.   :-1.07478   Min.   :-1.25026   Min.   :-0.79221
##  1st Qu.:-0.25710   1st Qu.:-0.26610   1st Qu.:-0.24527   1st Qu.:-0.26520
##  Median :-0.01173   Median : 0.06554   Median : 0.01264   Median : 0.03561
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.34146   3rd Qu.: 0.26079   3rd Qu.: 0.18906   3rd Qu.: 0.27995
##  Max.   : 1.07366   Max.   : 0.82841   Max.   : 1.19191   Max.   : 0.59786
##       PC13               PC14               PC15
##  Min.   :-0.72215   Min.   :-0.62323   Min.   :-0.118487
##  1st Qu.:-0.13879   1st Qu.:-0.18170   1st Qu.:-0.045544
##  Median : 0.04775   Median : 0.02648   Median :-0.006267
##  Mean   : 0.00000   Mean   : 0.00000   Mean   : 0.000000
##  3rd Qu.: 0.17023   3rd Qu.: 0.14384   3rd Qu.: 0.033971
##  Max.   : 0.57149   Max.   : 0.62939   Max.   : 0.165961
```

Scree Plot analysis

The scree plot is used to define whether the principal components are meant to be left in the analysis.

```r
#Plotting the scree plot
plot((my_pca$sdev^2)/sum(my_pca$sdev^2) * 100, xlab='Principal Components', ylab='Variance percentage', main="Scree Plot", las = 1)
lines((my_pca$sdev^2)/sum(my_pca$sdev^2) * 100)
abline(h = 1/ncol(data[,1:15])*100) # it is equal to 6.67%
```

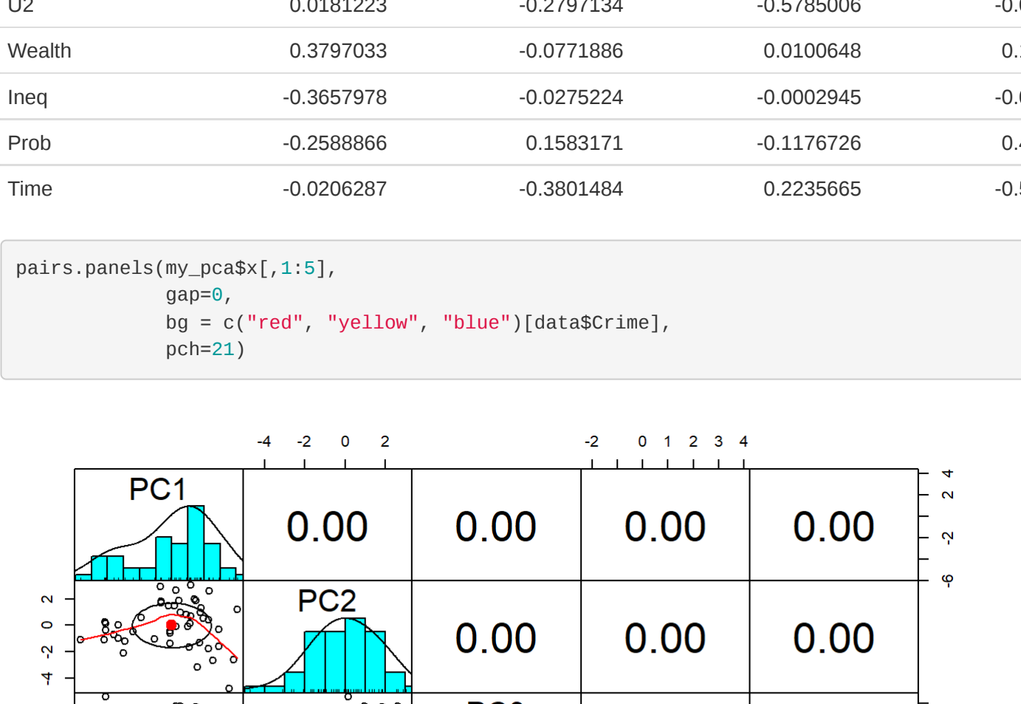


From the scree plot we can clearly see that only the first 5 components are considered to be important. We need to concentrate on these components as in theory it reduces effects of randomness.

```r
kable(my_pca$rotation[,1:5])
```

	PC1	PC2	PC3	PC4	PC5
M	-0.3037119	0.0628036	0.1724200	-0.0203554	-0.3583274
So	-0.3308813	-0.1583722	0.0155433	0.2924718	-0.1206113
Ed	0.3396215	0.2146115	0.0677396	0.0797437	-0.0244284
Po1	0.3086341	-0.2698176	0.0506458	0.3332506	-0.2352768
Po2	0.3109929	-0.2639630	0.0530651	0.3519281	-0.2047338
LF	0.1761776	0.3194304	0.2715302	-0.1432653	-0.3940759
M.F	0.1163822	0.3943443	-0.2031622	0.0104803	-0.5787744
Pop	0.1130784	-0.4672346	0.0770211	-0.0321051	-0.0831703
NW	-0.2935865	-0.2280112	0.0788157	0.2392597	-0.3607939
U1	0.0405014	0.0080744	-0.6590291	-0.1827910	-0.1313687
U2	0.0181223	-0.2797134	-0.5785006	-0.0688931	-0.1349949
Wealth	0.3797033	-0.0771886	0.0100648	0.1178175	0.0116768
Ineq	-0.3657978	-0.0275224	-0.0002945	-0.0806661	-0.2167282
Prob	-0.2588866	0.1583171	-0.1176726	0.4930339	0.1656283
Time	-0.0206287	-0.3801484	0.2235665	-0.5405900	-0.1476477

```r
pairs.panels(my_pca$x[,1:5],
             gap=0,
             bg = c("red", "yellow", "blue")[data$Crime],
             pch=21)
```




```r
#Let's fit the model using these components
pca_data <- as.data.frame(cbind(my_pca$x[,1:5],data$Crime))
pca_model <- lm(V6~., data = pca_data)
summary(pca_model)
```

```
##
## Call:
## lm(formula = V6 ~ ., data = pca_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -420.79 -185.01   12.21  146.24  447.86
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      35.59  25.428  < 2e-16 ***
## PC1            65.22      14.67   4.447 6.51e-05 ***
## PC2           -70.08      21.49  -3.261  0.00224 **
## PC3            25.19      25.41   0.992  0.32725
## PC4            69.45      33.37   2.081  0.04374 *
## PC5          -229.04      36.75  -6.232 2.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 244 on 41 degrees of freedom
## Multiple R-squared:  0.6452, Adjusted R-squared:  0.6019
## F-statistic: 14.91 on 5 and 41 DF,  p-value: 2.446e-08
```

The R-squared dropped slightly for 14% in comparison with a model without pca and reduced variables. 65 % is still considered as a good value.

Specifying PCA model in terms of the original variables

We need to unscale coefficients to perform comparative analysis

```r
#Scaled intercept
sc_int <- pca_model$coefficients[1]
sc_int
```

```
## (Intercept)
##    905.0851
```

```r
#Scaled coefficients
sc_coef <- -pca_model$coefficients[2:6]
sc_coef
```

```
##       PC1       PC2       PC3       PC4       PC5
## -65.21593  70.08312 -25.19408 -69.44603 229.04282
```

```r
#Regression coeficients for xj found by multiplying eigenvectors to scaled pca coefficients
sc_a <- -my_pca$rotation[,1:5]%*%sc_coef
t(sc_a)
```

```
##             M       So       Ed      Po1      Po2      LF      M.F      Pop
## [1,] 60.79435 37.84824 19.94776 117.3449 111.4508 76.2549 108.1266 58.88024
##            NW       U1       U2   Wealth    Ineq      Prob     Time
## [1,] 98.07179 2.866783 32.34551 35.93336 22.1037 -34.64026 27.20502
```

```r
#Unscaling the parameters
unsc_a <- sc_a/my_pca$scale
t(unsc_a)
```

```
##             M       So      Ed      Po1      Po2       LF      M.F      Pop
## [1,] 48.37374 79.01922 17.8312 39.48484 39.85892 1886.946 36.69366 1.546583
##            NW       U1       U2     Wealth     Ineq      Prob     Time
## [1,] 9.537384 159.0115 38.29933 0.03724014 5.540321 -1523.521 3.838779
```

```r
#Unscaling the intercept
unsc_int <- sc_int - sum(sc_a * my_pca$center/my_pca$scale)
unsc_int
```

```
## (Intercept)
##   -5933.837
```

```r
#Find the Sum of squared errors
prediction <- unsc_int + as.matrix(data[,1:15]) %*% unsc_a
sse2 <- sum((prediction - data[,16]) ^ 2)
sse1 <- sum((better_model$residuals) ^ 2)
if (sse2 < sse1) {
  print('The pca model has less sse')
  print(sse2)
} else {
  print('The first model has less sse')
  print(sse1)
}
```

```
## [1] "The first model has less sse"
## [1] 1556227
```

Surprisingly the Sum of Squared errors are higher for the model using PCA.

```r
#Let's perform the prediction on the given data
prediction_data
```

```
##    M So Ed Po1  Po2   LF M.F Pop  NW   U1  U2 Wealth Ineq Prob Time
## 1 14  0 10  12 15.5 0.64  94 150 1.1 0.12 3.6   3200 20.1 0.04   39
```

```r
paste0('New city has a crime rate of ', (unsc_int + as.matrix(prediction_data) %*% unsc_a))
```

```
## [1] "New city has a crime rate of 1388.92569475603"
```

ANSWER:
The purpose of this homework was to perform PCA analysis to reduce the randomness, remove correlation and rank coordinates by importance. By performing Scree plot we found that only first 5 principal components are important for our analysis. These pc-s where used for linear regression model and then parameters were unscaled for further investigation. The results indicate that SSE is higher for the PCA model and the R squared is lower. However, we know that higher percentage of accuracy does not prove better performing model, as there might be a chance of overfitting. The previous model had a value predicted closer to the mean and this time the predicted value for the new city is 1389, farther from the mean.
Scaled LM equation is:
Y = 905.0851 - 65.21593PC1 + 70.08312PC2 - 25.19408PC3 - 69.44603PC4 + 229.04282PC5.