Homework-6

Question 9.1

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)!)

```
rm(list=ls())

library(GGally)
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(DAAG)
```

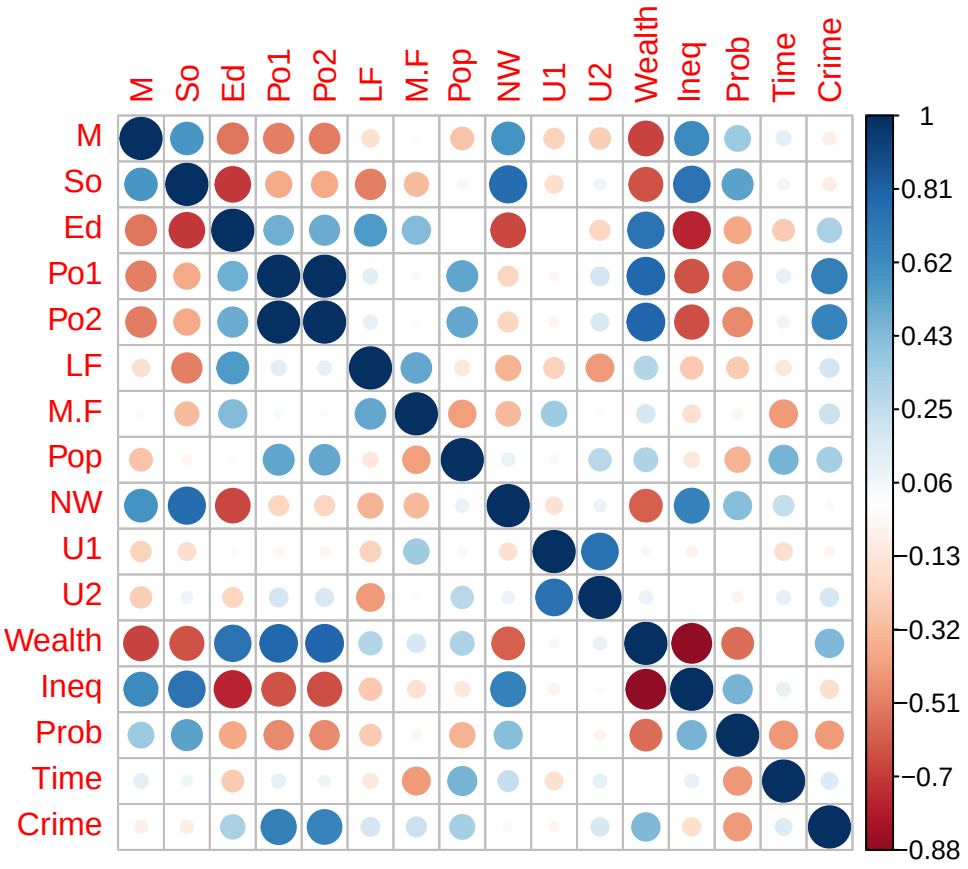
```
## Loading required package: lattice
```

```
# load data
data = read.table("C:/Users/Admin/Desktop/MM/Homework 6/uscrime.txt",
                  stringsAsFactors = FALSE,
                  header = TRUE)
head(data)
```

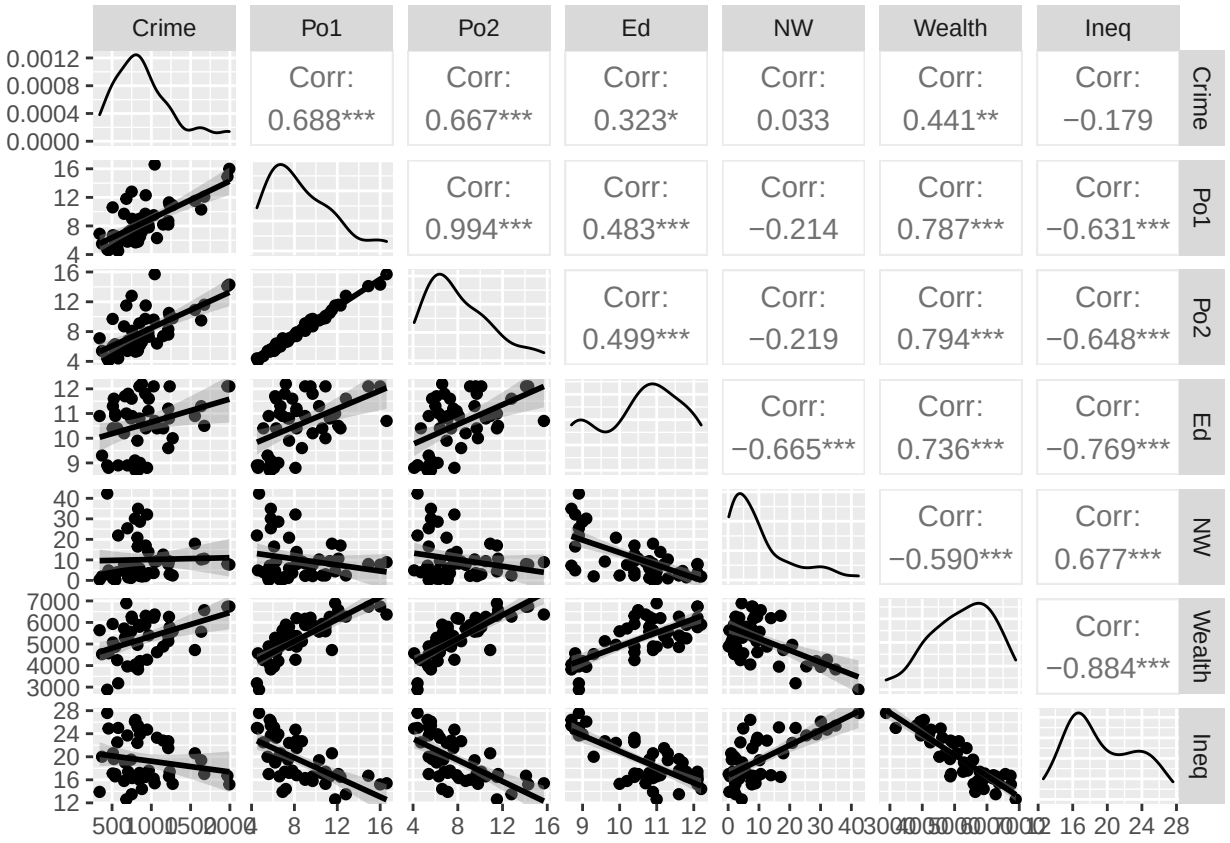
```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq     Prob
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1 0.084602
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4 0.029599
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0 0.083401
## 4 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
## 5 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
## 6 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
##      Time Crime
```

```
## 1 26.2011   791
## 2 25.2999  1635
## 3 24.3006   578
## 4 29.9012  1969
## 5 21.2998  1234
## 6 20.9995   682
```

```
# visualizing correlated variables
corr_matrix = cor(data)
corrplot(corr_matrix, method='circle', is.corr = FALSE)
```



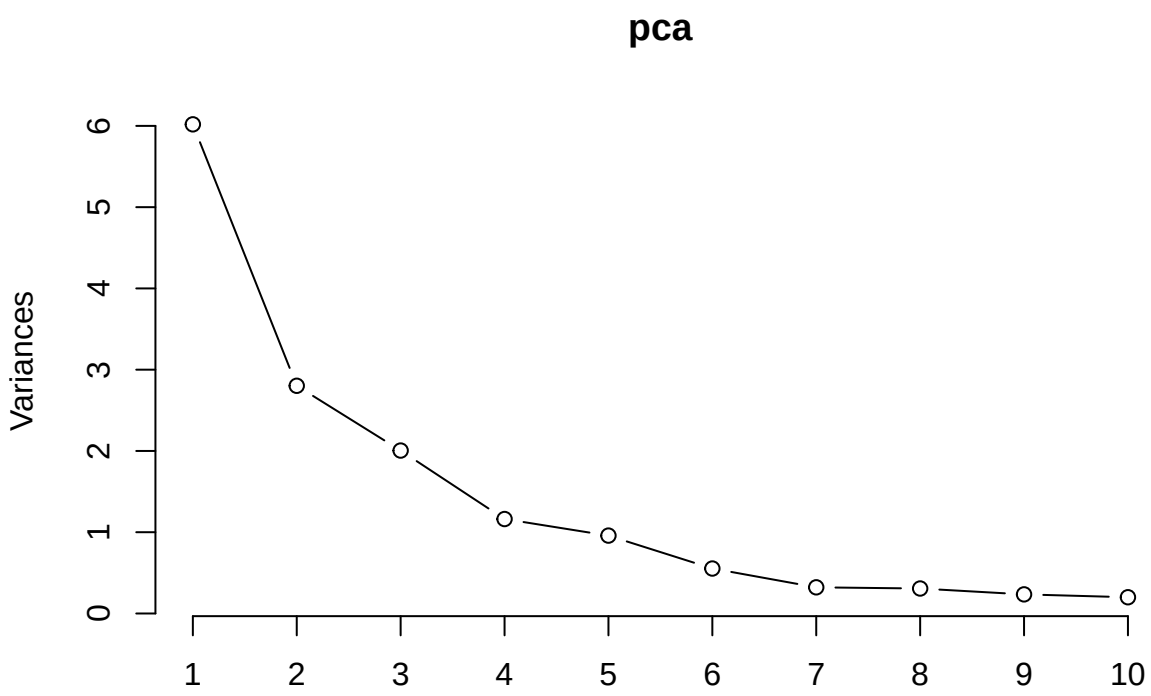

```
ggpairs(data, columns = c('Crime', 'Po1', 'Po2', 'Ed', 'NW', 'Wealth', 'Ineq'),
        lower=list(continuous='smooth'))
```




```
# pca
pca = prcomp(data[,1:15], scale.=TRUE)
summary(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
```

```
screeplot(pca, type='lines')
```




```
# building with 6 principal components
pca.data = cbind(pca$x[,1:6], data['Crime'])
model = lm(Crime~., data=pca.data)
summary(model)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = pca.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -377.15 -172.23   25.81  132.10  480.38
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      35.35  25.604  < 2e-16 ***
## PC1            65.22      14.56   4.478 6.14e-05 ***
## PC2           -70.08      21.35  -3.283  0.00214 **
## PC3            25.19      25.23   0.998  0.32409
## PC4            69.45      33.14   2.095  0.04252 *
## PC5          -229.04      36.50  -6.275 1.94e-07 ***
## PC6           -60.21      48.04  -1.253  0.21734
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 242.3 on 40 degrees of freedom
```

```
## Multiple R-squared:  0.6586, Adjusted R-squared:  0.6074
## F-statistic: 12.86 on 6 and 40 DF,  p-value: 4.869e-08
```

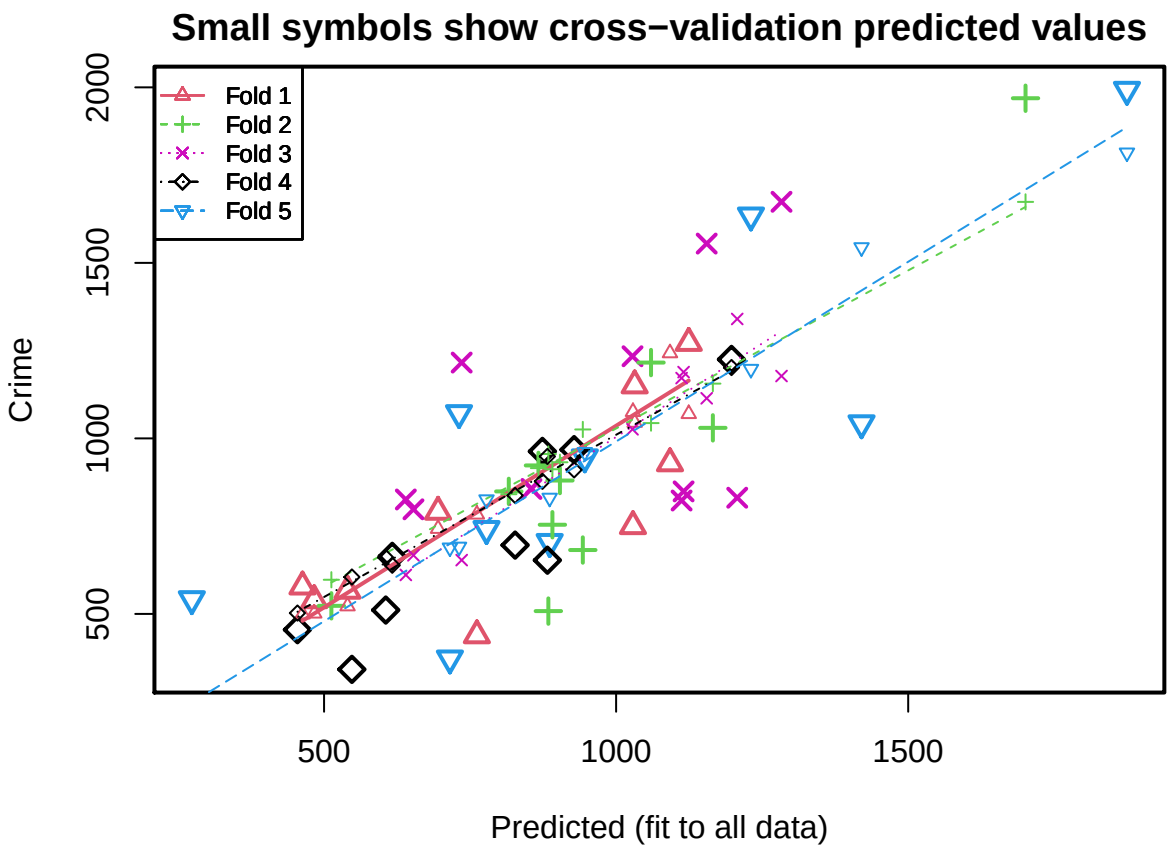
```
SST = sum((data$Crime - mean(data$Crime))^2)
SSR = sum(model$residuals^2)
```

Model with 6 components yield R^2= 0.6586023 .

```
cvmodel = cv.lm(pca.data, model, m=5)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## PC1        1 1177568 1177568   20.05 6.1e-05 ***
## PC2        1  633037  633037   10.78  0.0021 **
## PC3        1   58541   58541    1.00  0.3241
## PC4        1  257832  257832    4.39  0.0425 *
## PC5        1 2312556 2312556   39.38 1.9e-07 ***
## PC6        1   92261   92261    1.57  0.2173
## Residuals 40 2349133   58728
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Warning in cv.lm(pca.data, model, m = 5):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 9
##                 1     3    17   18   19   22   36  38   40
## Predicted   695.2 463.1 483.2 1092 1029  762 1124 540 1032
## cvpred      742.7 498.9 501.5 1243 1076  783 1070 522 1045
## Crime       791.0 578.0 539.0  929  750  439 1272 566 1151
## CV residual  48.3  79.1  37.5 -314 -326 -344  202  44  106
##
## Sum of squares = 387320    Mean square = 43036    n = 9
##
## fold 2
## Observations in test set: 10
##                4    6     12    25   28   32     34  41   44   46
## Predicted   1701  943 816.27 512.5 1060  891 867.01 904 1166  884
## cvpred      1674 1025 840.99 597.3 1044  912 915.34 932 1156  956
## Crime       1969  682 849.00 523.0 1216  754 923.00 880 1030  508
## CV residual  295 -343   8.01 -74.3  172 -158   7.66 -52 -126 -448
##
## Sum of squares = 484414    Mean square = 48441    n = 10
##
## fold 3
## Observations in test set: 10
##                5    8     9   11  15   23   37  39   43   47
## Predicted   1028 1155 855.02 1283 653  736 1207 640 1112 1116
## cvpred      1026 1114 861.14 1178 667  653 1340 611 1172 1189
```

```
## Crime       1234 1555 856.00 1674 798 1216  831 826  823  849
## CV residual  208  441  -5.14  496 131  563 -509 215 -349 -340
##
## Sum of squares = 1360587    Mean square = 136059    n = 10
##
## fold 4
## Observations in test set: 9
##                 7   13    14     20    24   27   30   35    45
## Predicted   873.8  606 616.7 1197.5 928.1  548  827  882 454.5
## cvpred      877.7  661 638.6 1201.9 910.3  605  837  949 502.4
## Crime       963.0  511 664.0 1225.0 968.0  342  696  653 455.0
## CV residual  85.3 -150  25.4   23.1  57.7 -263 -141 -296 -47.4
##
## Sum of squares = 213008    Mean square = 23668    n = 9
##
## fold 5
## Observations in test set: 9
##                2   10    16    21   26   29   31   33  42
## Predicted   1231  886 946.4 778.1 1875 1420  715  731 274
## cvpred      1198  830 960.7 825.8 1814 1544  689  691 221
## Crime       1635  705 946.0 742.0 1993 1043  373 1072 542
## CV residual  437 -125 -14.7 -83.8  179 -501 -316  381 321
##
## Sum of squares = 845402    Mean square = 93934    n = 9
##
## Overall (Sum over all 9 folds)
##    ms
## 70016
```

```
SSR.cv = attr(cvmodel, 'ms')*nrow(pca.data)
```

5-fold cross validation of model yield $R^2 = 0.522$.

```
# building with 4 significant principal components
pca.data = cbind(pca$x[,1:6][,c(1, 2, 4, 5)], data['Crime'])
model = lm(Crime~., data=pca.data)
summary(model)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = pca.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -401.9 -181.5  -33.9  124.5  465.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    905.1       35.6   25.43  < 2e-16 ***
## PC1             65.2       14.7    4.45  6.3e-05 ***
## PC2            -70.1       21.5   -3.26   0.0022 **
## PC4             69.4       33.4    2.08   0.0435 *
## PC5           -229.0       36.7   -6.23  1.8e-07 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 244 on 42 degrees of freedom
## Multiple R-squared:  0.637,  Adjusted R-squared:  0.602
## F-statistic: 18.4 on 4 and 42 DF,  p-value: 8.38e-09
```

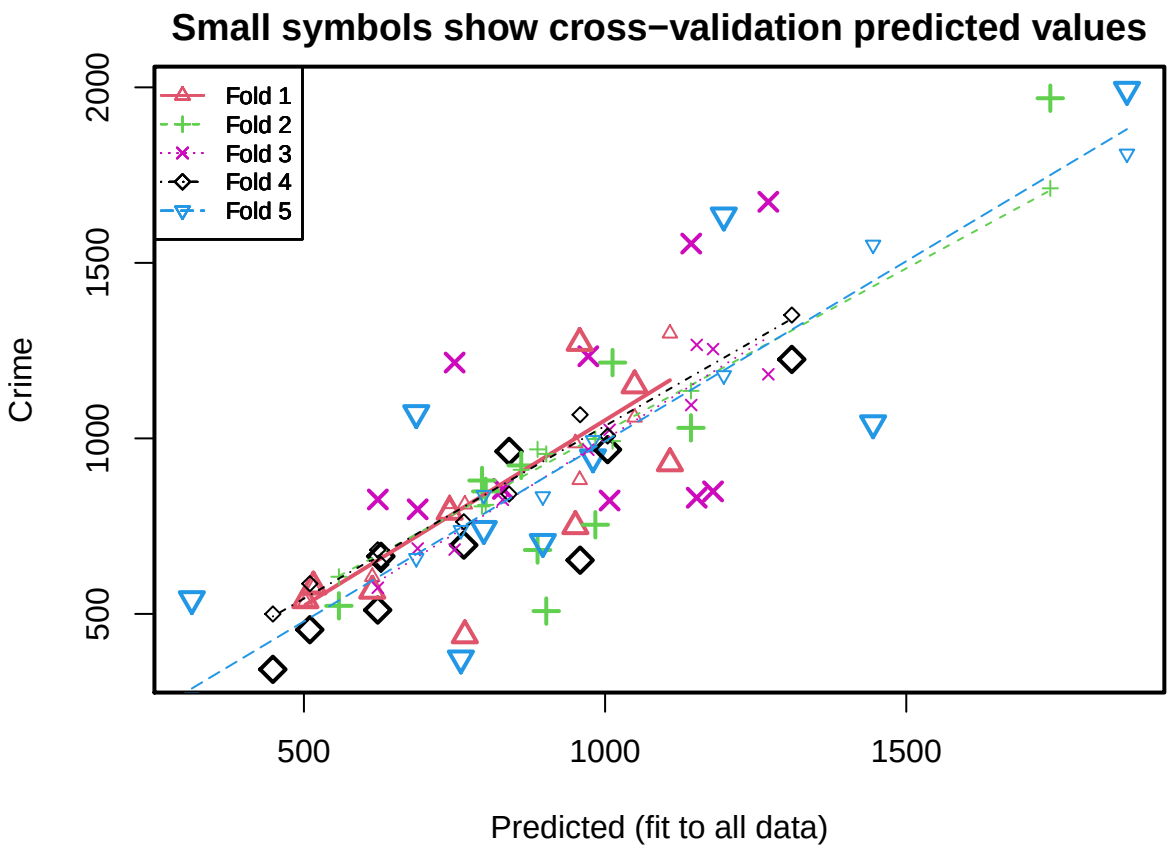
```
SSR = sum(model$residuals^2)
```

Model with 4 components yield $R^2 = 0.637$.

```
cvmodel = cv.lm(pca.data, model, m=5)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## PC1        1 1177568 1177568   19.78 6.3e-05 ***
## PC2        1  633037  633037   10.64  0.0022 **
## PC4        1  257832  257832    4.33  0.0435 *
## PC5        1 2312556 2312556   38.85 1.8e-07 ***
## Residuals 42 2499935   59522
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Warning in cv.lm(pca.data, model, m = 5):
##
##  As there is >1 explanatory variable, cross-validation
##  predicted values for a fold are not a linear function
##  of corresponding overall predicted values.  Lines that
##  are shown for the different folds are approximate
```




```
##
## fold 1
## Observations in test set: 9
##                 1      3     17   18   19   22   36    38     40
## Predicted   741.9 515.75 502.45 1108  951  767  958 613.5 1049.0
## cvpred      813.4 575.89 535.45 1299  987  812  882 605.7 1059.8
## Crime       791.0 578.00 539.00  929  750  439 1272 566.0 1151.0
## CV residual -22.4   2.11   3.55 -370 -237 -373  390 -39.7   91.2
##
## Sum of squares = 494899    Mean square = 54989    n = 9
##
## fold 2
## Observations in test set: 10
##                4    6    12    25   28   32    34    41   44   46
## Predicted   1739  888 801.8 558.1 1012  984 860.8 795.7 1143  902
## cvpred      1712  969 810.6 605.9  992  998 910.5 806.3 1135  956
## Crime       1969  682 849.0 523.0 1216  754 923.0 880.0 1030  508
## CV residual  257 -287  38.4 -82.9  224 -244  12.5  73.7 -105 -448
##
## Sum of squares = 483339    Mean square = 48334    n = 10
##
## fold 3
## Observations in test set: 10
##                5    8   9   11  15   23   37  39   43   47
## Predicted    972 1143 830 1271 689  750 1152 623 1007 1179
## cvpred       968 1095 825 1182 686  683 1266 575 1025 1254
```

```
## Crime       1234 1555 856 1674 798 1216  831 826  823  849
## CV residual  266  460  31  492 112  533 -435 251 -202 -405
##
## Sum of squares = 1279050    Mean square = 127905    n = 10
##
## fold 4
## Observations in test set: 9
##                7   13    14   20     24   27    30   35   45
## Predicted   841  623 627.9 1310 1004.2  448 765.5  959  510
## cvpred      841  683 640.7 1351 1006.9  500 762.4 1067  586
## Crime       963  511 664.0 1225  968.0  342 696.0  653  455
## CV residual 122 -172  23.3 -126  -38.9 -158 -66.4 -414 -131
##
## Sum of squares = 280413    Mean square = 31157    n = 9
##
## fold 5
## Observations in test set: 9
##                 2   10    16    21   26   29   31   33  42
## Predicted   1197  897 979.9 798.6 1866 1445  761  687 314
## cvpred      1180  835 994.8 838.3 1811 1551  739  659 261
## Crime       1635  705 946.0 742.0 1993 1043  373 1072 542
## CV residual  455 -130 -48.8 -96.3  182 -508 -366  413 281
##
## Sum of squares = 911165    Mean square = 101241    n = 9
##
## Overall (Sum over all 9 folds)
##    ms
## 73380
```

```
SSR.cv = attr(cvmodel, 'ms')*nrow(pca.data)
```

5-fold cross validation of model yield $R^2 = 0.499$.

Not using PCA seemed to yield better models that predicts and explains the variability of the data better.

```
# reverse scaling
beta = model$coefficients
alpha.scaled = pca$rotation[,c(1, 2, 4, 5)] %*% beta[2:5]

alpha = alpha.scaled/sapply(data[,1:15], sd)
intercept = beta[1] - sum(alpha.scaled*sapply(data[,1:15], mean)/
                          sapply(data[,1:15], sd))

# Coefficients of unscaled variables
alpha
```

```
##             [,1]
## M         44.917
## So        78.202
## Ed        16.306
## Po1       39.055
## Po2       39.381
## LF      1717.665
```

```
## M.F       38.431
## Pop        1.496
## NW         9.344
## U1      1079.963
## U2        55.557
## Wealth     0.037
## Ineq       5.542
## Prob   -1393.132
## Time       3.044
```

```
# Unscaled Intercept
intercept
```

```
## (Intercept)
##       -6064
```