Homework 6

Richard Albright
ISYE6501
Spring 2018

2/16/2019

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)!)

The following is an analysis of the original 15 factor model, the 6 factor model as determined by significant p-values from the lm() function, and results from a model constructed using Principal Component Analysis.

Read in the CSV

```r
data <-
  read.table(
    "/Users/ralbright/Dropbox/ISYE6501/week3/homework/uscrime.txt",
    header=TRUE,
    sep="\t"
  )
```

Head:

```r
table <- xtable(head(data))
print(table, type='latex', comment=FALSE, scalebox='0.75')
```

	M	So	Ed	Po1	Po2	LF	M.F	Pop	NW	U1	U2	Wealth	Ineq	Prob	Time	Crime
1	15.10	1	9.10	5.80	5.60	0.51	95.00	33	30.10	0.11	4.10	3940	26.10	0.08	26.20	791
2	14.30	0	11.30	10.30	9.50	0.58	101.20	13	10.20	0.10	3.60	5570	19.40	0.03	25.30	1635
3	14.20	1	8.90	4.50	4.40	0.53	96.90	18	21.90	0.09	3.30	3180	25.00	0.08	24.30	578
4	13.60	0	12.10	14.90	14.10	0.58	99.40	157	8.00	0.10	3.90	6730	16.70	0.02	29.90	1969
5	14.10	0	12.10	10.90	10.10	0.59	98.50	18	3.00	0.09	2.00	5780	17.40	0.04	21.30	1234
6	12.10	0	11.00	11.80	11.50	0.55	96.40	25	4.40	0.08	2.90	6890	12.60	0.03	21.00	682

Tail:

```r
table <- xtable(tail(data))
print(table, type='latex', comment=FALSE, scalebox='0.75')
```

	M	So	Ed	Po1	Po2	LF	M.F	Pop	NW	U1	U2	Wealth	Ineq	Prob	Time	Crime
42	14.10	0	10.90	5.60	5.40	0.52	96.80	4	0.20	0.11	3.70	4890	17.00	0.09	12.20	542
43	16.20	1	9.90	7.50	7.00	0.52	99.60	40	20.80	0.07	2.70	4960	22.40	0.05	32.00	823
44	13.60	0	12.10	9.50	9.60	0.57	101.20	29	3.60	0.11	3.70	6220	16.20	0.03	30.00	1030
45	13.90	1	8.80	4.60	4.10	0.48	96.80	19	4.90	0.14	5.30	4570	24.90	0.06	32.60	455
46	12.60	0	10.40	10.60	9.70	0.60	98.90	40	2.40	0.08	2.50	5930	17.10	0.05	16.70	508
47	13.00	0	12.10	9.00	9.10	0.62	104.90	3	2.20	0.11	4.00	5880	16.00	0.05	16.10	849

Summary:

```
table <- xtable(summary(data))
print(table, type='latex', comment=FALSE, scalebox='0.4')
```

	M	So	Ed	Po1	Po2	LF	M.F	Pop
X	Min. :11.90	Min. :0.0000	Min. : 8.70	Min. : 4.50	Min. : 4.100	Min. :0.4800	Min. : 93.40	Min. : 3.00
X.1	1st Qu.:13.00	1st Qu.:0.0000	1st Qu.: 9.75	1st Qu.: 6.25	1st Qu.: 5.850	1st Qu.:0.5305	1st Qu.: 96.45	1st Qu.: 10.00
X.2	Median :13.60	Median :0.0000	Median :10.80	Median : 7.80	Median : 7.300	Median :0.5600	Median : 97.70	Median : 25.00
X.3	Mean :13.86	Mean :0.3404	Mean :10.56	Mean : 8.50	Mean : 8.023	Mean :0.5612	Mean : 98.30	Mean : 36.62
X.4	3rd Qu.:14.60	3rd Qu.:1.0000	3rd Qu.:11.45	3rd Qu.:10.45	3rd Qu.: 9.700	3rd Qu.:0.5930	3rd Qu.: 99.20	3rd Qu.: 41.50
X.5	Max. :17.70	Max. :1.0000	Max. :12.20	Max. :16.60	Max. :15.700	Max. :0.6410	Max. :107.10	Max. :168.00

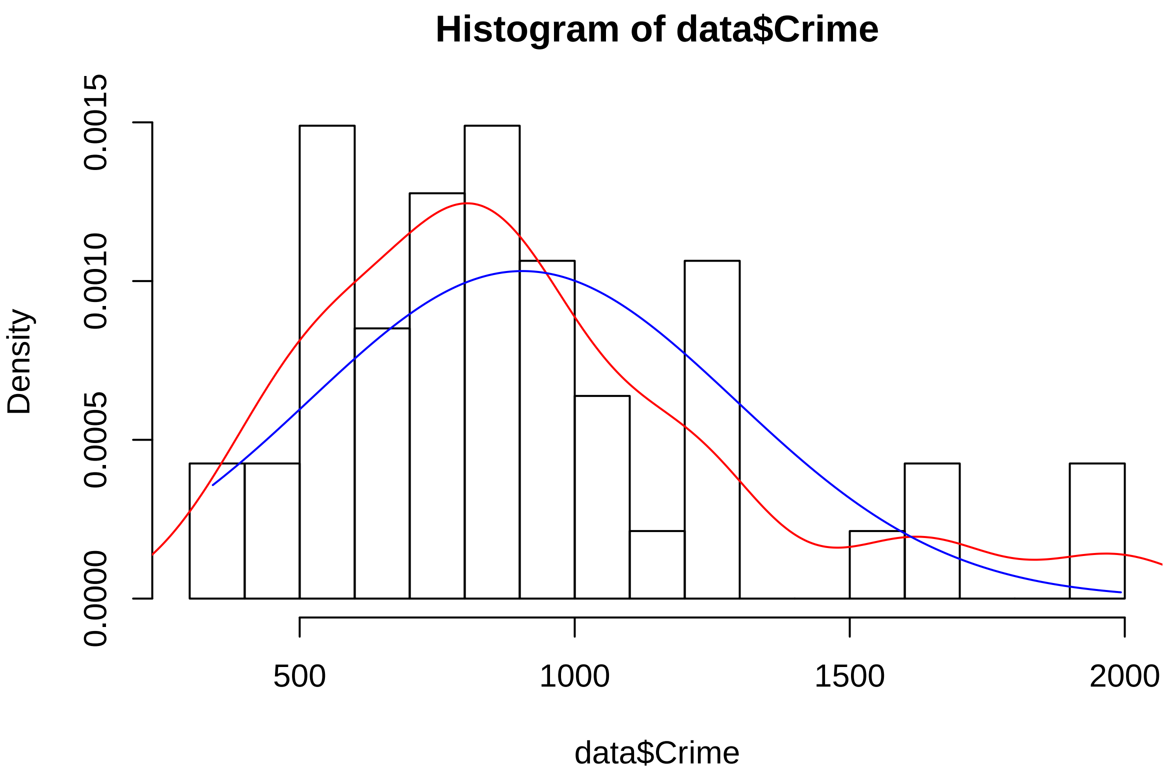
	NW	U1	U2	Wealth	Ineq	Prob	Time	Crime
X	Min. : 0.20	Min. :0.07000	Min. :2.000	Min. :2880	Min. :12.60	Min. :0.00690	Min. :12.20	Min. : 342.0
X.1	1st Qu.: 2.40	1st Qu.:0.08050	1st Qu.:2.750	1st Qu.:4595	1st Qu.:16.55	1st Qu.:0.03270	1st Qu.:21.60	1st Qu.: 658.5
X.2	Median : 7.60	Median :0.09200	Median :3.400	Median :5370	Median :17.60	Median :0.04210	Median :25.80	Median : 831.0
X.3	Mean :10.11	Mean :0.09547	Mean :3.398	Mean :5254	Mean :19.40	Mean :0.04709	Mean :26.60	Mean : 905.1
X.4	3rd Qu.:13.25	3rd Qu.:0.10400	3rd Qu.:3.850	3rd Qu.:5915	3rd Qu.:22.75	3rd Qu.:0.05445	3rd Qu.:30.45	3rd Qu.:1057.5
X.5	Max. :42.30	Max. :0.14200	Max. :5.800	Max. :6890	Max. :27.60	Max. :0.11980	Max. :44.00	Max. :1993.0

Example analysis from http://www.statsci.org/data/general/uscrime.html

Testing our data set for outliers using grubbs.test

Lets 1st plot a histogram of our Crime Response variable vs its density and a overlay of the normal distribution.

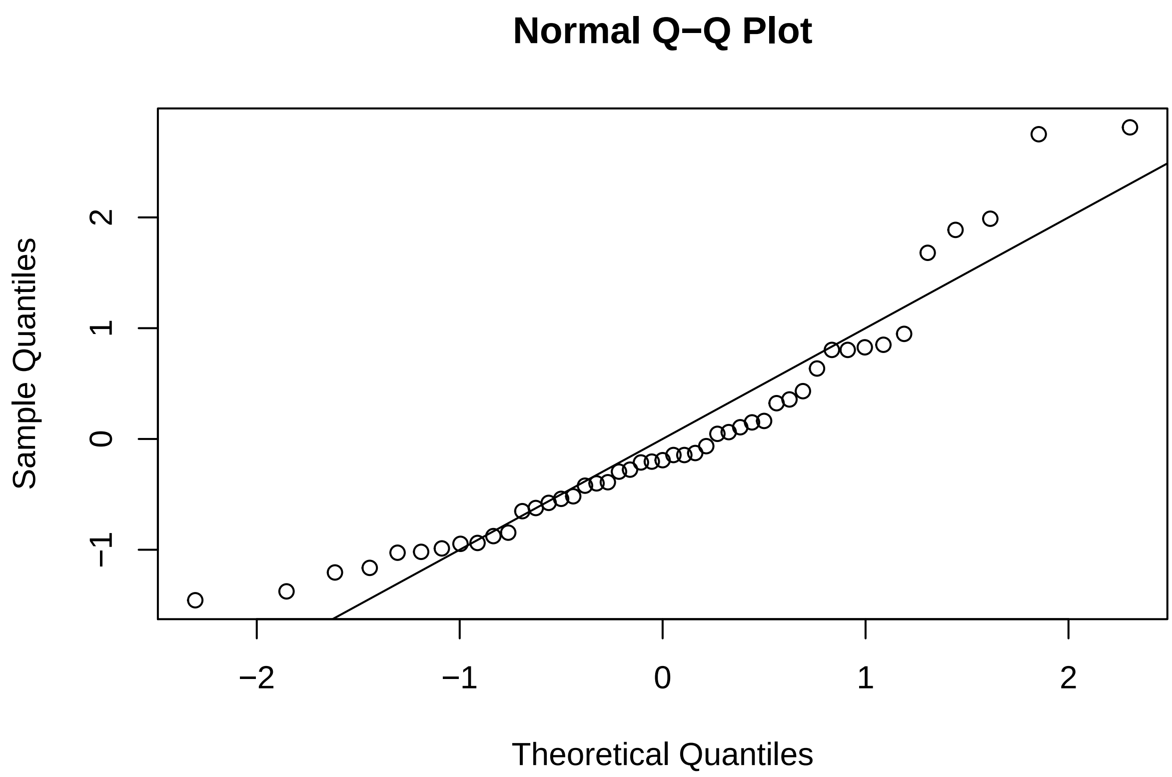
```
hist(data$Crime, freq=F, breaks=12)
lines(density(data$Crime), col="red")
lines(seq(min(data$Crime), max(data$Crime)), dnorm(seq(min(data$Crime), max(data$Crime)),mean(data$Crim
```



The left tail seems to indicate there may be some outliers in our data set.

The plot of the scaled Crime Response Variable using qqnorm also looks like.

```
scaled_crime = scale(data$Crime)
qqnorm(scaled_crime)
abline(0,1)
```



Which seems to indicate that there may outliers in both tails.

Lets take a look at a box plot of our Crime response variable as well.

```
boxplot(data$Crime, main="Crime", boxwex=0.1)
```


```
possible_outliers <- boxplot.stats(data$Crime)$out
possible_outliers
```

```
## [1] 1969 1674 1993
```

The boxplot points to possible outliers in the upper tail. Output from boxplot.stats indicates that the 3 possible outliers are 1969, 1674, & 1993. We will now use the grubbs.test function to test for the outliers from the data set.

We will use the 1st 2 tests of the The grubbs.test function below (taken directly from the R Documentation).

First test (10) is used to detect if the sample dataset contains one outlier, statistically different than the other values. Test is based by calculating score of this outlier G (outlier minus mean and divided by sd) and comparing it to appropriate critical values. Alternative method is calculating ratio of variances of two datasets - full dataset and dataset without outlier. The obtained value called U is bound with G by simple formula.

Second test (11) is used to check if lowest and highest value are two outliers on opposite tails of sample. It is based on calculation of ratio of range to standard deviation of the sample.

We will loop through the 1st two test types on the Crime column.

```
tests <- c(10, 11)
for(test in tests) {
  for(truth in c(TRUE,FALSE)) {
    gtest <- grubbs.test(as.vector(data$Crime), type=test, opposite=truth)
    print(paste('Grubbs Test Type:', test, collapse=' '))
    print(gtest)
  }
}
```

```
## [1] "Grubbs Test Type: 10"
##
##  Grubbs test for one outlier
##
## data:  as.vector(data$Crime)
## G = 1.45590, U = 0.95292, p-value = 1
## alternative hypothesis: lowest value 342 is an outlier
##
## [1] "Grubbs Test Type: 10"
##
##  Grubbs test for one outlier
##
## data:  as.vector(data$Crime)
## G = 2.81290, U = 0.82426, p-value = 0.07887
## alternative hypothesis: highest value 1993 is an outlier
##
## [1] "Grubbs Test Type: 11"
##
##  Grubbs test for two opposite outliers
##
## data:  as.vector(data$Crime)
## G = 4.26880, U = 0.78103, p-value = 1
## alternative hypothesis: 342 and 1993 are outliers
##
## [1] "Grubbs Test Type: 11"
##
##  Grubbs test for two opposite outliers
##
## data:  as.vector(data$Crime)
## G = 4.26880, U = 0.78103, p-value = 1
## alternative hypothesis: 342 and 1993 are outliers
```

Using a 95% confidence interval, We accept the null hypothesis that there are not any outliers in our Crime reponse variable.

We will perform a linear regression using the lm() function using the last column Crime vs its predictor columns.

```
lm.crime <- lm(Crime~., data=data, names=names(data))
summary(lm.crime,correlation=FALSE)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = data, names = names(data))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -5984.28760  1628.31837  -3.675 0.000893 ***
## M              87.83017    41.71387   2.106 0.043443 *
## So             -3.80345   148.75514  -0.026 0.979765
## Ed            188.32431    62.08838   3.033 0.004861 **
## Po1           192.80434   106.10968   1.817 0.078892 .
## Po2          -109.42193   117.47754  -0.931 0.358830
## LF           -663.82615  1469.72882  -0.452 0.654654
## M.F            17.40686    20.35384   0.855 0.398995
## Pop            -0.73301     1.28956  -0.568 0.573845
## NW              4.20446     6.48089   0.649 0.521279
## U1          -5827.10272  4210.28904  -1.384 0.176238
## U2            167.79967    82.33596   2.038 0.050161 .
## Wealth          0.09617     0.10367   0.928 0.360754
## Ineq           70.67210    22.71652   3.111 0.003983 **
## Prob        -4855.26582  2272.37462  -2.137 0.040627 *
## Time           -3.47902     7.16528  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 209.1 on 31 degrees of freedom
## Multiple R-squared:  0.8031, Adjusted R-squared:  0.7078
## F-statistic: 8.429 on 15 and 31 DF,  p-value: 0.0000003539
```

The R-squared and adjusted R-squared from our model fitting the entire data set is 0.8030868 and 0.7078062.

Lets calculate the AIC and BIC of our initial model.

```
aic1 = AIC(lm.crime)
aic1
```

```
## [1] 650.0291
```

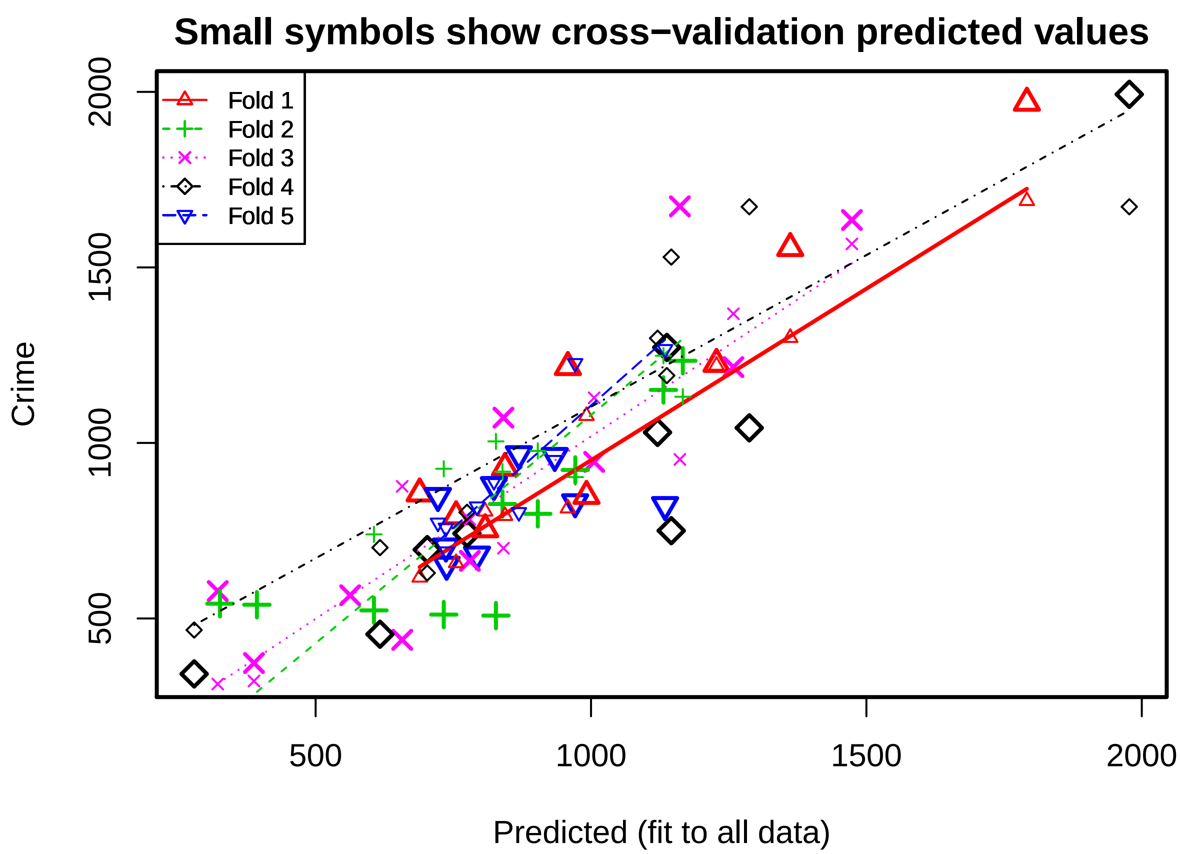
```
bic1= BIC(lm.crime)
bic1
```

```
## [1] 681.4816
```

We'll then perform a K-Fold cross validation on our initial model using 5 folds.

```r
lm.crime.cv <- cv.lm(data, lm.crime, m=5)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value       Pr(>F)
## M          1   55084   55084    1.26       0.2702
## So         1   15370   15370    0.35       0.5575
## Ed         1  905668  905668   20.72 0.0000772205 ***
## Po1        1 3076033 3076033   70.38 0.0000000018 ***
## Po2        1  153024  153024    3.50       0.0708 .
## LF         1   61134   61134    1.40       0.2459
## M.F        1  111000  111000    2.54       0.1212
## Pop        1   42649   42649    0.98       0.3309
## NW         1   14197   14197    0.32       0.5728
## U1         1    7065    7065    0.16       0.6904
## U2         1  269663  269663    6.17       0.0186 *
## Wealth     1   34748   34748    0.79       0.3795
## Ineq       1  547423  547423   12.52       0.0013 **
## Prob       1  222620  222620    5.09       0.0312 *
## Time       1   10304   10304    0.24       0.6307
## Residuals 31 1354946   43708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```




```
##
## fold 1
```

```
## Observations in test set: 9
##               1    4    8   9  18      20   23    32   47
## Predicted   755 1791 1362 689 844 1227.84  958 807.8  992
## cvpred      658 1690 1300 617 792 1220.22  814 804.9 1077
## Crime       791 1969 1555 856 929 1225.00 1216 754.0  849
## CV residual 133  279  255 239 137    4.78  402 -50.9 -228
##
## Sum of squares = 453204    Mean square = 50356    n = 9
##
## fold 2
## Observations in test set: 10
##                5   13   15  17   25    34    39     40    42   46
## Predicted   1167  733  903 393  606 971.5 839.3 1131.5 326.3  827
## cvpred      1132  926  977 152  740 902.7 918.1 1248.5  62.3 1004
## Crime       1234  511  798 539  523 923.0 826.0 1151.0 542.0  508
## CV residual  102 -415 -179 387 -217  20.3 -92.1  -97.5 479.7 -496
##
## Sum of squares = 906384    Mean square = 90638    n = 10
##
## fold 3
## Observations in test set: 10
##                  2   3   11   14   16   22   28    31   33      38
## Predicted   1473.7 322 1161  780 1006  657 1258 388.0  841 562.693
## cvpred      1566.9 313  953  782 1129  876 1368 321.7  700 566.231
## Crime       1635.0 578 1674  664  946  439 1216 373.0 1072 566.000
## CV residual   68.1 265  721 -118 -183 -437 -152  51.3  372  -0.231
##
## Sum of squares = 997216    Mean square = 99722    n = 10
##
## fold 4
## Observations in test set: 9
##               19    21   26   27   29    30     36   44   45
## Predicted   1146 774.9 1977  279 1287 702.7 1137.6 1121  617
## cvpred      1529 802.3 1673  467 1673 629.6 1191.9 1298  702
## Crime        750 742.0 1993  342 1043 696.0 1272.0 1030  455
## CV residual -779 -60.3  320 -125 -630  66.4   80.1 -268 -247
##
## Sum of squares = 1269688    Mean square = 141076    n = 9
##
## fold 5
## Observations in test set: 9
##                6     7    10    12  24    35   37  41   43
## Predicted    793 934.2 736.5 722.0 869 737.8  971 824 1134
## cvpred       819 950.9 758.1 772.5 802 690.5 1227 891 1267
## Crime        682 963.0 705.0 849.0 968 653.0  831 880  823
## CV residual -137  12.1 -53.1  76.5 166 -37.5 -396 -11 -444
##
## Sum of squares = 410109    Mean square = 45568    n = 9
##
## Overall (Sum over all 9 folds)
##    ms
## 85885
```

Then let's calculate our r^2 for our K-fold cross validated model.

```r
sse1 = attr(lm.crime.cv, 'ms') * nrow(data)
sst1 = sum((data$Crime - mean(data$Crime)) ^ 2)
rsquared1 = 1 - sse1/sst1
rsquared1
```

```
## [1] 0.413
```

We find that the best predictors after performing a linear regression are M, Ed, Po1, U2, Ineq, and Prob. Our initial model's adjusted R-squared accounts for approximately 41.336% of the variance of the data set.

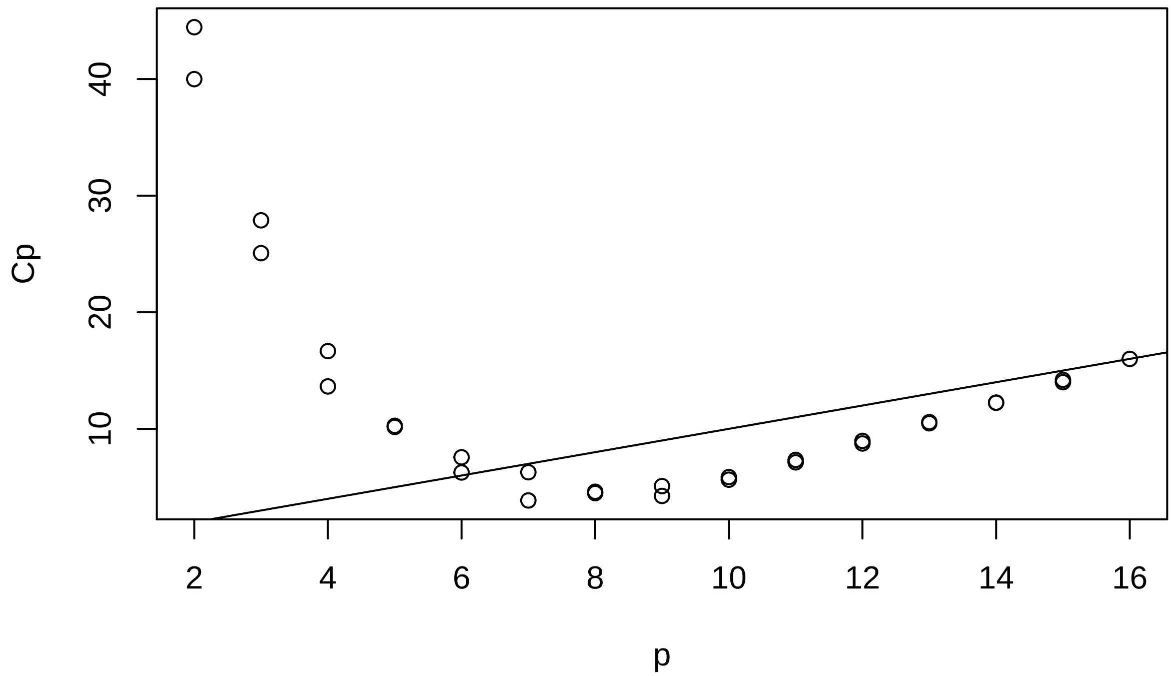
The leaps functions is an all subsets regression function that attempts to find the best predictors for use in a linear regression model. This can be used as an alternative to a stepwise AIC (which does stepwise regression). We can then run our predictors through the leaps functions to verify if in fact our predictors are the best ones to use (Information about leaps here: http://www2.hawaii.edu/~taylor/z632/Rbestsubsets.pdf). We want to find the combination of number of p predictors is closest in value to Mallows C_p Statistic ($p=C_p$) (https://en.wikipedia.org/wiki/Mallows's_Cp).

```r
leaps.crime <- leaps(data[,1:15],data$Crime,nbest=2, names=names(data[,1:15]))
```

```r
leaps.tab <- data.frame(p=leaps.crime$size,Cp=leaps.crime$Cp)
round(leaps.tab,2)
```

```
##     p    Cp
## 1   2 40.00
## 2   2 44.45
## 3   3 25.07
## 4   3 27.89
## 5   4 13.64
## 6   4 16.67
## 7   5 10.16
## 8   5 10.26
## 9   6  6.26
## 10  6  7.56
## 11  7  3.86
## 12  7  6.28
## 13  8  4.49
## 14  8  4.61
## 15  9  4.24
## 16  9  5.09
## 17 10  5.64
## 18 10  5.86
## 19 11  7.13
## 20 11  7.34
## 21 12  8.75
## 22 12  8.97
## 23 13 10.48
## 24 13 10.58
## 25 14 12.24
## 26 14 12.25
## 27 15 14.00
## 28 15 14.20
## 29 16 16.00
```

```r
plot(leaps.tab)
abline(0,1)
```



We can see from the chart that using 6 predictors gives you the best linear regression model (The 1st point where the AB line crosses a scatter point from left to right). This agrees with what was identified as significant in our initial models K-fold cross validation. Now let's generate a linear regression model using only these factors as identified as significant in our initial K-fold cross validated lm model.

```
lm.crime2 <- lm(Crime~M+Ed+Po1+U2+Ineq+Prob,data=data)
summary(lm.crime2)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -470.7  -78.4  -19.7  133.1  556.2
##
## Coefficients:
##             Estimate Std. Error t value      Pr(>|t|)
## (Intercept)  -5040.5      899.8   -5.60 0.00000171527 ***
## M              105.0       33.3    3.15        0.0031 **
## Ed             196.5       44.8    4.39 0.00008072016 ***
## Po1            115.0       13.8    8.36 0.00000000026 ***
## U2              89.4       40.9    2.18        0.0348 *
## Ineq            67.7       13.9    4.85 0.00001879377 ***
## Prob         -3801.8     1528.1   -2.49        0.0171 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201 on 40 degrees of freedom
## Multiple R-squared:  0.766,  Adjusted R-squared:  0.731
## F-statistic: 21.8 on 6 and 40 DF,  p-value: 0.0000000000342
```

Let's calculate the AIC and BIC of our improved model.

```r
aic2 = AIC(lm.crime2)
aic2
```

```
## [1] 640
```

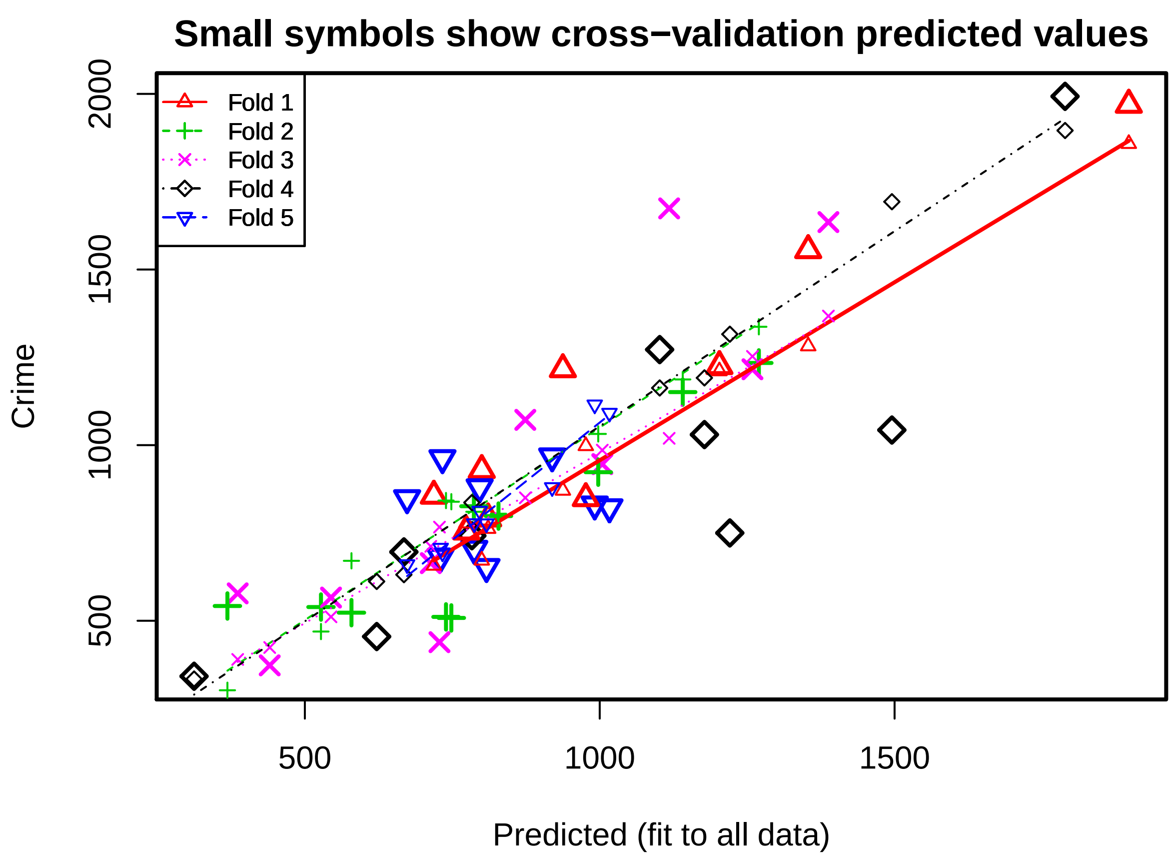
```r
bic2 = BIC(lm.crime2)
bic2
```

```
## [1] 655
```

We'll now perform a cross validation of our improved model using 5 folds.

```r
lm.crime2.cv <- cv.lm(data, lm.crime2, m=5)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value         Pr(>F)
## M          1   55084   55084    1.37        0.24914
## Ed         1  725967  725967   18.02        0.00013 ***
## Po1        1 3173852 3173852   78.80 0.000000000053 ***
## U2         1  217386  217386    5.40        0.02534 *
## Ineq       1  848273  848273   21.06 0.000043385425 ***
## Prob       1  249308  249308    6.19        0.01711 *
## Residuals 40 1611057   40276
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```




```
##
```

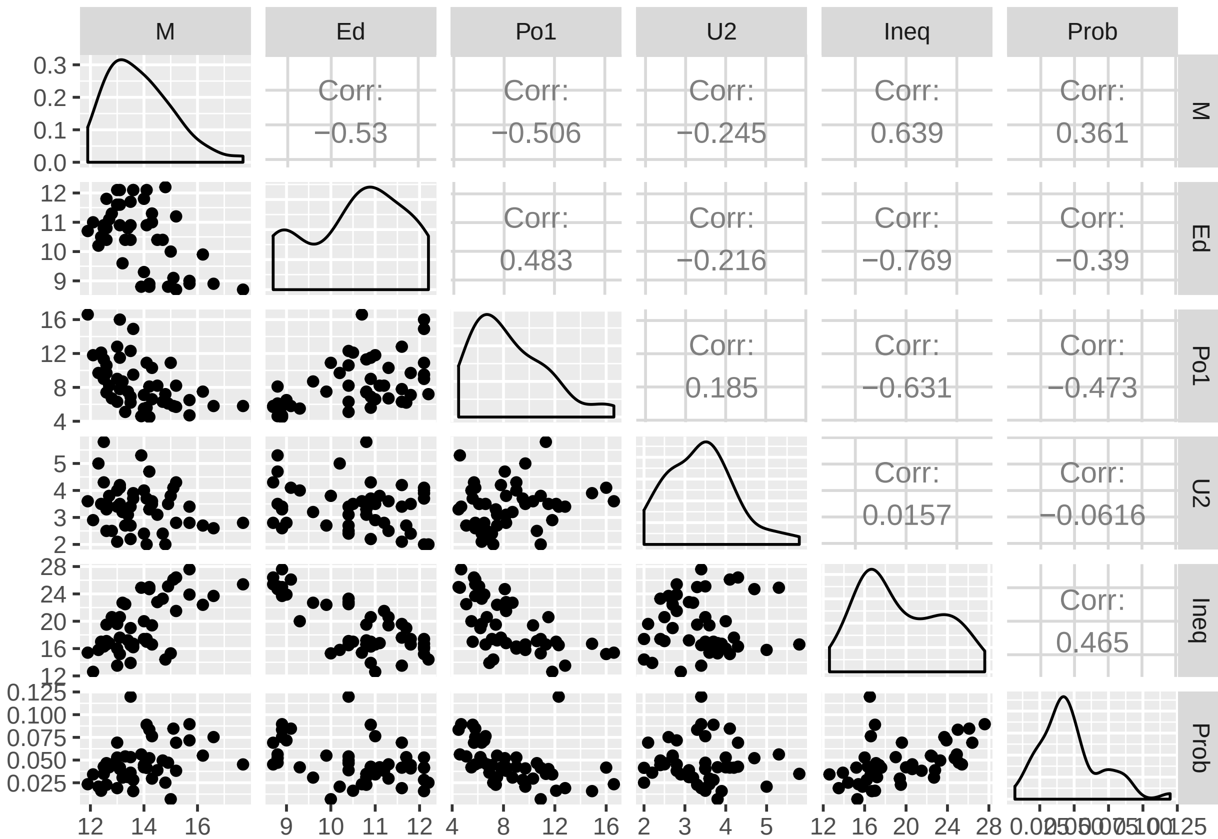
```
## fold 1
## Observations in test set: 9
##                 1    4    8   9  18     20   23    32   47
## Predicted   810.8 1897 1354 719 800 1203.0  938 773.7  976
## cvpred      762.1 1858 1282 657 672 1210.8  871 777.6  998
## Crime       791.0 1969 1555 856 929 1225.0 1216 754.0  849
## CV residual  28.9  111  273 199 257   14.2  345 -23.6 -149
##
## Sum of squares = 335463    Mean square = 37274    n = 9
##
## fold 2
## Observations in test set: 10
##                5   13     15    17   25   34    39   40  42   46
## Predicted   1270  739 828.34 527.4  579  998 786.7 1141 369  748
## cvpred      1337  842 804.73 469.3  671 1032 810.3 1187 302  839
## Crime       1234  511 798.00 539.0  523  923 826.0 1151 542  508
## CV residual -103 -331  -6.73  69.7 -148 -109  15.7  -36 240 -331
##
## Sum of squares = 327423    Mean square = 32742    n = 10
##
## fold 3
## Observations in test set: 10
##                2   3   11    14     16   22     28    31   33    38
## Predicted   1388 386 1118 713.6 1004.4  728 1259.0 440.4  874 544.4
## cvpred      1368 390 1019 711.8  985.8  767 1252.6 423.8  850 511.2
## Crime       1635 578 1674 664.0  946.0  439 1216.0 373.0 1072 566.0
## CV residual  267 188  655 -47.8  -39.8 -328  -36.6 -50.8  222  54.8
##
## Sum of squares = 702726    Mean square = 70273    n = 10
##
## fold 4
## Observations in test set: 9
##               19    21     26     27   29    30   36   44   45
## Predicted   1221 783.3 1789.1 312.20 1495 668.0 1102 1178  622
## cvpred      1316 836.4 1895.7 334.15 1693 631.2 1163 1191  612
## Crime        750 742.0 1993.0 342.00 1043 696.0 1272 1030  455
## CV residual -566 -94.4   97.3   7.85 -650  64.8  109 -161 -157
##
## Sum of squares = 827924    Mean square = 91992    n = 9
##
## fold 5
## Observations in test set: 9
##              6   7    10  12    24   35   37    41   43
## Predicted   730 733 787.3 673 919.4  808  992 796.4 1017
## cvpred      707 694 776.8 660 879.7  777 1115 812.6 1091
## Crime       682 963 705.0 849 968.0  653  831 880.0  823
## CV residual -25 269 -71.8 189  88.3 -124 -284  67.4 -268
##
## Sum of squares = 294201    Mean square = 32689    n = 9
##
## Overall (Sum over all 9 folds)
##    ms
## 52931
```

```r
sse2 = attr(lm.crime2.cv, 'ms') * nrow(data)
sst2 = sum((data$Crime - mean(data$Crime)) ^ 2)
rsquared2 = 1 - sse2/sst2
rsquared2
```

```
## [1] 0.638
```

Now that we have our r^2 for our model from Question 8.2, we'll run a Principal Component Analysis on our original data set and see how that differs from just using the best p values from the lm() function.

```r
# ggpairs() shows correlation
# prcomp() is the pca function
ggpairs(data, columns=c("M", "Ed", "Po1", "U2", "Ineq", "Prob"))
```



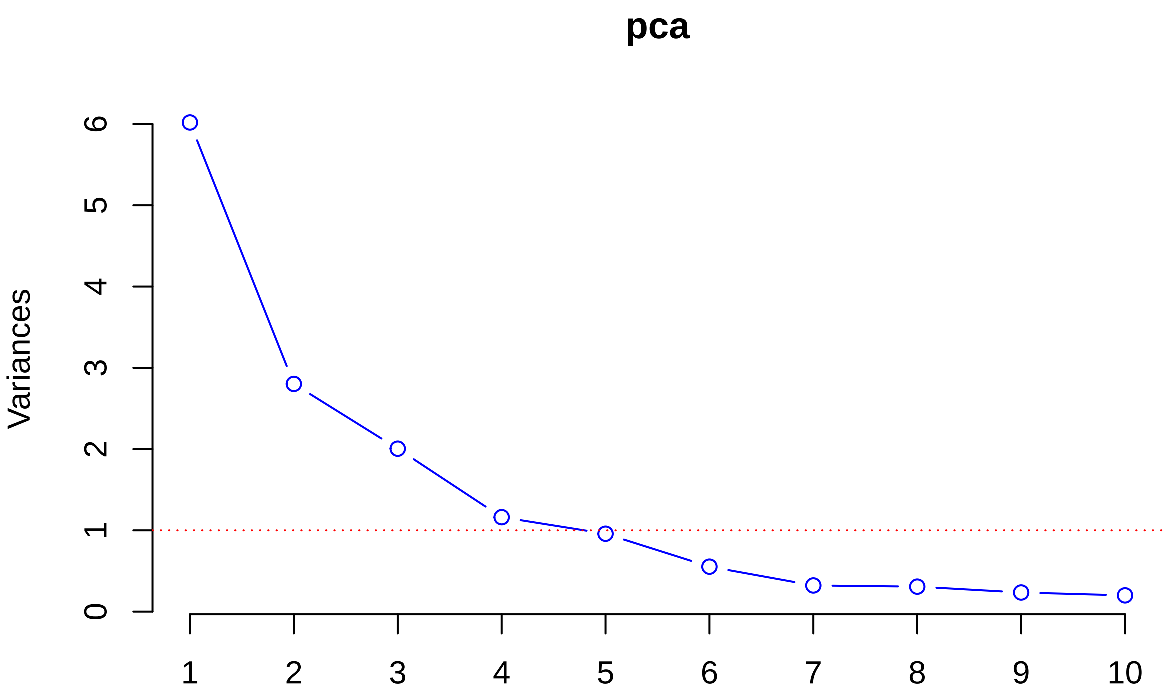

```r
pca <- prcomp(data[,1:15], scale=TRUE)
summary(pca)
```

```
## Importance of components:
##                          PC1   PC2   PC3    PC4    PC5    PC6    PC7
## Standard deviation     2.453 1.674 1.416 1.0781 0.9789 0.7438 0.5673
## Proportion of Variance 0.401 0.187 0.134 0.0775 0.0639 0.0369 0.0215
## Cumulative Proportion  0.401 0.588 0.722 0.7992 0.8631 0.9000 0.9214
##                           PC8    PC9   PC10   PC11    PC12    PC13   PC14
## Standard deviation     0.5544 0.4849 0.4471 0.4191 0.35804 0.26333 0.2418
## Proportion of Variance 0.0205 0.0157 0.0133 0.0117 0.00855 0.00462 0.0039
## Cumulative Proportion  0.9419 0.9576 0.9709 0.9826 0.99117 0.99579 0.9997
##                           PC15
## Standard deviation     0.06793
```

```
## Proportion of Variance 0.00031
## Cumulative Proportion  1.00000
```

Using the scree test we only want to include the first i Principal Components where their eigenvalues (variance) at >= 1 (the Kaiser criterion).

```r
# plot the variances of the components
screeplot(pca, type="lines", col="blue")
abline(h=1,lty=3, col="red")
```



This results in our selection of the 1st 5 components

```r
#get 1st i components pca$x[,1:i]
pc <- pca$x[,1:5]
# data of only our principal components
data_pc <- cbind(pc, Crime=data[,16])
# 6th column is our response value
lm.crime_pca = lm(Crime~PC1+PC2+PC3+PC4+PC5, data=as.data.frame(data_pc))
summary(lm.crime_pca)
```

```
##
## Call:
## lm(formula = Crime ~ PC1 + PC2 + PC3 + PC4 + PC5, data = as.data.frame(data_pc))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -420.8 -185.0   12.2  146.2  447.9
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)    905.1       35.6   25.43 < 0.0000000000000002 ***
## PC1             65.2       14.7    4.45            0.0000651 ***
## PC2            -70.1       21.5   -3.26               0.0022 **
## PC3             25.2       25.4    0.99               0.3272
```

```
## PC4             69.4       33.4    2.08              0.0437 *
## PC5           -229.0       36.8   -6.23           0.0000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 244 on 41 degrees of freedom
## Multiple R-squared:  0.645,  Adjusted R-squared:  0.602
## F-statistic: 14.9 on 5 and 41 DF,  p-value: 0.0000000245
```

Now let's calculate the PCA Model's AIC and BIC values.

```
aic3 = AIC(lm.crime_pca)
aic3
```

```
## [1] 658
```

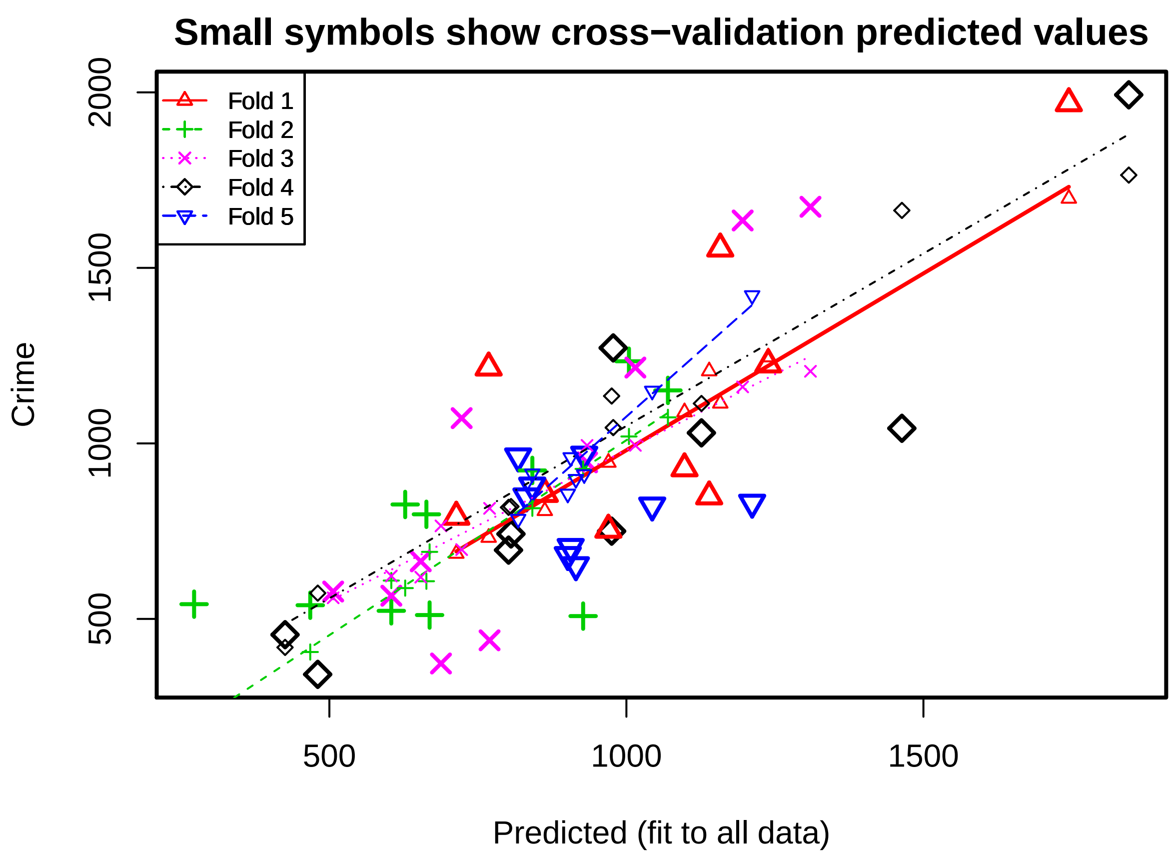
```
bic3 = BIC(lm.crime_pca)
bic3
```

```
## [1] 671
```

We'll now perform a cross validation of our PCA model using 5 folds.

```
lm.crime_pca.cv <- cv.lm(as.data.frame(data_pc), lm.crime_pca, m=5)
```

```
## Analysis of Variance Table
##
## Response: Crime
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## PC1        1 1177568 1177568   19.78 0.0000651 ***
## PC2        1  633037  633037   10.63    0.0022 **
## PC3        1   58541   58541    0.98    0.3272
## PC4        1  257832  257832    4.33    0.0437 *
## PC5        1 2312556 2312556   38.84 0.0000002 ***
## Residuals 41 2441394   59546
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```




```
##
## fold 1
## Observations in test set: 9
##                1    4    8     9   18     20   23   32   47
## Predicted    714 1745 1158 862.7 1098 1238.8  768  970 1139
## cvpred       686 1698 1114 807.9 1089 1245.7  732  945 1206
## Crime        791 1969 1555 856.0  929 1225.0 1216  754  849
## CV residual 105  271  441  48.1 -160  -20.7  484 -191 -357
##
## Sum of squares = 706357    Mean square = 78484    n = 9
##
## fold 2
## Observations in test set: 10
##                 5   13  15  17    25  34  39     40  42   46
## Predicted    1004  669 663 468 604.2 842 628 1069.9 272  927
## cvpred       1020  691 607 406 609.3 815 588 1074.2 185  927
## Crime        1234  511 798 539 523.0 923 826 1151.0 542  508
## CV residual  214 -180 191 133 -86.3 108 238   76.8 357 -419
##
## Sum of squares = 517100    Mean square = 51710    n = 10
##
## fold 3
## Observations in test set: 10
##                 2     3   11    14    16   22   28   31   33    38
## Predicted    1196 506.4 1310 653.8 933.8  770 1015  688  723 604.3
## cvpred       1161 560.1 1205 618.9 994.2  815  994  765  697 622.2
## Crime        1635 578.0 1674 664.0 946.0  439 1216  373 1072 566.0
```

```
## CV residual  474  17.9  469  45.1 -48.2 -376  222 -392  375 -56.2
##
## Sum of squares = 936438    Mean square = 93644    n = 10
##
## fold 4
## Observations in test set: 9
##               19  21   26   27   29   30   36     44    45
## Predicted    975 806 1846  480 1464  802  978 1126.3 425.5
## cvpred      1135 820 1764  573 1664  818 1045 1113.6 418.8
## Crime        750 742 1993  342 1043  696 1272 1030.0 455.0
## CV residual -385 -78  229 -231 -621 -122  227  -83.6  36.2
##
## Sum of squares = 719985    Mean square = 79998    n = 9
##
## fold 5
## Observations in test set: 9
##                6   7   10    12    24   35   37    41   43
## Predicted    901 818  906 831.7 929.0  915 1212 841.5 1043
## cvpred       856 785  960 886.8 911.7  898 1422 913.8 1150
## Crime        682 963  705 849.0 968.0  653  831 880.0  823
## CV residual -174 178 -255 -37.8  56.3 -245 -591 -33.8 -327
##
## Sum of squares = 648385    Mean square = 72043    n = 9
##
## Overall (Sum over all 9 folds)
##    ms
## 75069
```

```
sse_pca = attr(lm.crime_pca.cv, 'ms') * nrow(data_pc)
sst3 = sum((data$Crime - mean(data$Crime)) ^ 2)
rsquared_pca = 1 - sse_pca/sst3
rsquared_pca
```

```
## [1] 0.487
```

The leaps analysis performed above also confirms that leaving M in as a predictor results in a better model, even though our K-fold validation indicates that M was not significant. Our AIC for all 3 models is 650.029, 640.166, and 657.703. The AIC for all 3 models indicates the 2nd model is the best. The BIC of all 3 models is 681.482, 654.967, and 670.654. The 2nd model of 654.967 is much better than our 1st model's BIC of 681.482, and our 3rd model's BIC of 670.654. Our R-squared values for all 3 models is 0.413, 0.638, and 0.487. This also confirms our 2nd model is likely to be the best. The unsupercvised PCA model does perform better than the initial 15 factor model, and is created without using the response variable.

Let's see how good the PCA model is at predicting the point provided in the homework.

The following is our test point.

```
test_point <- 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)
```

In order to convert our PCA model back to its original coordinate space, we need to get the betas of our model and convert them to alphas using our PCA model's eigenvectors.

Let's get our beta intercept and the beta coefficients for the model.

```
beta_coef = lm.crime_pca$coefficients[-1]
beta_intercept = lm.crime_pca$coefficients[1]
```

To convert our betas back into alpha's we need to matrix multiply the 5 pricipal components eigenvectors used in our model (the rotation variable in our object) by our beta coefficients.

```
alphas = pca$rotation[,1:5] %*% beta_coef
```

We also need the means and stand deviations of our original data set to do our conversion back to our original coordinates.

```
means = sapply(data[,1:15], mean)
stddevs = sapply(data[,1:15], sd)
```

The original coefficients are the alphas divided by the standard deviations.

```
original_coefs = alphas / stddevs
original_coefs
```

```
##              [,1]
## M         48.3737
## So        79.0192
## Ed        17.8312
## Po1       39.4848
## Po2       39.8589
## LF      1886.9458
## M.F       36.6937
## Pop        1.5466
## NW         9.5374
## U1       159.0115
## U2        38.2993
## Wealth     0.0372
## Ineq       5.5403
## Prob   -1523.5214
## Time       3.8388
```

To get the original coordinates intercept, we subtract the sum of our alphas * the means / standard deviations from our beta intercept.

```
original_intercept <- beta_intercept - sum(alphas * means / stddevs)
original_intercept
```

```
## (Intercept)
##       -5934
```

We now hove our PCA model converted back to the original intercept and coefficients, so we can now make a prediction based on our test point.

```
crime_prediction_pca <- original_intercept + sum(original_coefs * test_point)
crime_prediction_pca
```

```
## (Intercept)
##        1389
```

Our resulting predicted crime rate for the provided point using our PCA model is 1388.926.