Question 8.1

I think a regression model would be appropriate for the marketing department to understand the correlation between the amount of money they spend on advertising and revenue returns. A simple linear regression model with appropriate advertisement spending would be one predictor variable with revenue as the response variable.

- Some other predictors that can correlate with revenue are product price, customer satisfaction, and inventory.
- As for advertisement spending variables, some response variables can be product sales volume and transaction volumn.

Question 8.2

The first step in fitting a linear model is loading the data and quickly studying it. I consider the response's density as well as the relationships between predictors and the relationship between predictors and the response. In most cases, we only want to include the factors that account for a significant portion of the response variable and avoid predictors that are associated with one another. From reading the data, we know that Crime is a response, and other variables are predictors. We use the entire dataset to build a regression model which is then used for prediction. We're not choosing between models (so validation isn't needed) and we're not bothering to estimate model quality (so test data isn't needed).

From the predicted model, we get a result of

```
## 155.4349
```

Which is unexpected. The estimate is significantly lower than the next-lowest city's crime rate. Since none of the test data point's factor values fall outside the range of the other data points, that cannot be the cause. I purposely picked this data point to serve as an example. The complete model we used above has a large number of unimportant components. We'll go back and get an estimate using only the important variables. We'll attempt to use all the variables when p=0.1.

This time we get

```
## 1304.245
```

This seems like a more reasonable prediction, now that the insignificant factors are gone.

Now that we can use this data to calculate R-squared for model, model2, and cross-validation. So, this shows that including the insignificant factors overfits compared to removing them, and even the fitted model is probably overfitted. That's not so surprising, since we started with just 47 data points and we have 15 factors to predict from. The ratio of data points to factors is about 3:1, and it's usually good to have 10:1 or more.

The results are displayed after using the linear regression function glm() from the R stats package with all of the supplied data. glm() is a more-general function for regression. The values in the "Estimate" column represent the coefficients of the regression model for each predictor, and the values in the

"Pr(>|t|)" column represent the p-value, indicating the predictors' significance to the model. We determine a crime rate of 155 using the model on the provided point data, which is incredibly low.

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *  
## So          -3.803e+00  1.488e+02  -0.026 0.979765    
## Ed           1.883e+02  6.209e+01   3.033 0.004861 ** 
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .  
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830    
## LF          -6.638e+02  1.470e+03  -0.452 0.654654    
## M.F          1.741e+01  2.035e+01   0.855 0.398995    
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845    
## NW           4.204e+00  6.481e+00   0.649 0.521279    
## U1          -5.827e+03  4.210e+03  -1.384 0.176238    
## U2           1.678e+02  8.234e+01   2.038 0.050161 .  
## Wealth       9.617e-02  1.037e-01   0.928 0.360754    
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 ** 
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *  
## Time        -3.479e+00  7.165e+00  -0.486 0.630708    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the adjusted R-squared penalizes for having too many predictors, the summary reveals a significant discrepancy between the R-squared value and the R-squared value, indicating that the model is overfitting. Additionally, we can observe that many predictors have high p-values, indicating that there is a minimal link between them and the crime rate. Getting rid of every predictor with a p-value higher than 0.8, was selected because Po1 was identified by the software as a significant predictor.

Rerunning glm() with the revised reduced function results in the summary depicted in Figure 3, and running with the data point results in a predicted crime rate of 1304, which is considerably more realistic. All of the p-values are low, and the difference between the R-squared value and the adjusted R-squared value has greatly narrowed, as can be shown.

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 ** 
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *  
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -2.488  0.01711 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We may compare the R2 values, which are 0.803 vs. 0.766, to assess how well the two models fit each other. The R2 value for the model summary is 80%. This indicates that our model accounts for almost 80% of the entire variance in crime. Although the performance of the training set is not necessarily a reliable predictor of model quality, this model may appear to be a decent one. We should carry out some sort of cross validation or at the very least test the model on an independent dataset to better gauge the model's quality. At first appearance, model 1 appears to be superior; however, when we calculate the R2 values for each model using the cross-validation regression function cv.lm() with five folds, we find that model 1's value is 0.413 while model 2's is 0.638. Model 2 is the superior model overall and is less overfit than Model 1 according to these data, but Model 2 is still overfit, proving that the original R2 estimate of 0.766 was way too optimistic.

Untitled2

2022-09-28

```r
rm(list = ls())


# Setting the random number generator seed so that results are reproducible
set.seed(1)

#First, Read in the data

dat <- read.table("/Users/xiaofanjiao/Desktop/uscrime.txt", stringsAsFactors = FALSE, header = TRUE)

head(dat)
```

```
##      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
```

```r
model <- lm( Crime ~ ., data = dat)

#Summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = dat)
##
## 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) -5.984e+03  1.628e+03  -3.675 0.000893 ***
```

```
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -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: 3.539e-07
```

```
#Create the test datapoint manually using dataframe

test <-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,

#Predict the crime rate for test data point
pred_model <- predict(model, test)
pred_model
```

```
##        1
## 155.4349
```

Use just the singificant factors to get an estimate. We'll try using all of the factors with p<=0.1.

```
model2 <- lm( Crime ~  M + Ed + Po1 + U2 + Ineq + Prob, data = dat)

#Summary of the model
summary(model2)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
```

```
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -2.488  0.01711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 40 degrees of freedom
## Multiple R-squared:  0.7659, Adjusted R-squared:  0.7307
## F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

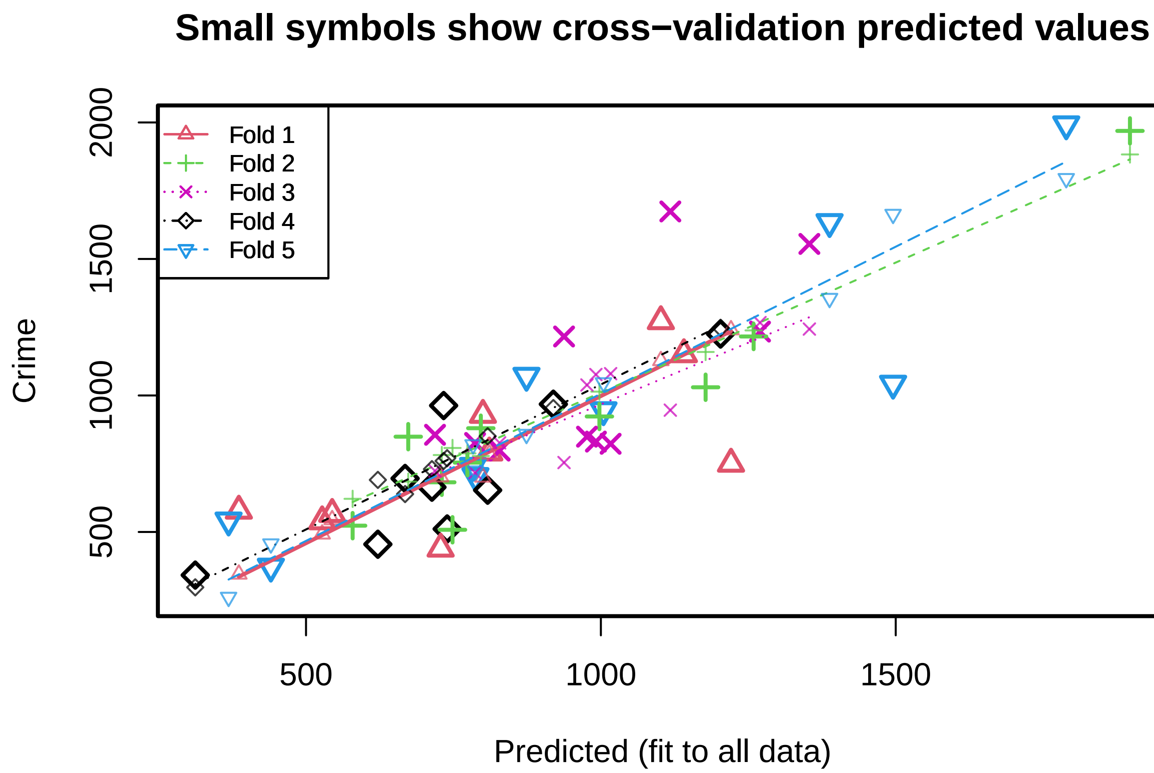
```r
#Predict on our test observation
pred_model2 <- predict(model2, test)
pred_model2
```

```
##        1
## 1304.245
```

```r
# Install the DAAG package, which has cross-validation functions
#install.packages("DAAG")
library(DAAG)

# do 5-fold cross-validation
c <- cv.lm(dat,model2,m=5)
```

```
## Warning in cv.lm(dat, model2, 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
## Predicted    810.825487 386.1368 527.3659 800.0046 1220.6767  728.3110 1101.7167
## cvpred       785.364736 345.3417 492.2016 700.5751 1240.2916  701.5126 1127.3318
## Crime        791.000000 578.0000 539.0000 929.0000  750.0000  439.0000 1272.0000
## CV residual    5.635264 232.6583  46.7984 228.4249 -490.2916 -262.5126  144.6682
##                     38         40
## Predicted    544.37325 1140.79061
## cvpred       544.69903 1168.21107
## Crime        566.00000 1151.00000
## CV residual   21.30097  -17.21107
##
## Sum of squares = 439507.2    Mean square = 48834.14    n = 9
##
## fold 2
## Observations in test set: 10
##                       4         6        12        25          28        32
## Predicted    1897.18657 730.26589 673.3766 579.06379 1259.00338 773.68402
## cvpred       1882.73805 781.75573 684.3525 621.37453 1238.31917 788.03429
## Crime        1969.00000 682.00000 849.0000 523.00000 1216.00000 754.00000
## CV residual    86.26195 -99.75573 164.6475 -98.37453  -22.31917 -34.03429
##                     34       41        44        46
## Predicted     997.54981 796.4198 1177.5973  748.4256
## cvpred       1013.92532 778.0437 1159.3155  807.6968
```

```
## Crime        923.00000 880.0000 1030.0000  508.0000
## CV residual  -90.92532 101.9563 -129.3155 -299.6968
##
## Sum of squares = 181038.4    Mean square = 18103.83    n = 10
##
## fold 3
## Observations in test set: 10
##                      5         8         9        11        15        23
## Predicted   1269.84196 1353.5532 718.7568 1117.7702 828.34178  937.5703
## cvpred      1266.79544 1243.1763 723.5331  946.1309 826.28548  754.2511
## Crime       1234.00000 1555.0000 856.0000 1674.0000 798.00000 1216.0000
## CV residual  -32.79544  311.8237 132.4669  727.8691 -28.28548  461.7489
##                     37        39        43        47
## Predicted    991.5623 786.6949 1016.5503  976.4397
## cvpred      1076.5799 717.0989 1079.7748 1038.3321
## Crime        831.0000 826.0000  823.0000  849.0000
## CV residual -245.5799 108.9011 -256.7748 -189.3321
##
## Sum of squares = 1033612    Mean square = 103361.1    n = 10
##
## fold 4
## Observations in test set: 9
##                     7        13        14        20        24        27
## Predicted   733.3799  739.3727 713.56395 1202.9607 919.39117 312.20470
## cvpred      759.9655  770.2015 730.05546 1247.8616 953.72478 297.19321
## Crime       963.0000  511.0000 664.00000 1225.0000 968.00000 342.00000
## CV residual 203.0345 -259.2015 -66.05546  -22.8616  14.27522  44.80679
##                     30        35        45
## Predicted   668.01610  808.0296  621.8592
## cvpred      638.87118  850.6961  690.6802
## Crime       696.00000  653.0000  455.0000
## CV residual  57.12882 -197.6961 -235.6802
##
## Sum of squares = 213398.5    Mean square = 23710.94    n = 9
##
## fold 5
## Observations in test set: 9
##                      2        10        16        21        26        29
## Predicted   1387.8082 787.27124 1004.3984 783.27334 1789.1406 1495.4856
## cvpred      1355.7097 723.66781 1046.8197 819.71145 1794.6456 1663.6272
## Crime       1635.0000 705.00000  946.0000 742.00000 1993.0000 1043.0000
## CV residual  279.2903 -18.66781 -100.8197 -77.71145  198.3544 -620.6272
##                    31        33       42
## Predicted   440.4394  873.8469 368.7031
## cvpred      456.5736  857.7052 260.9211
## Crime       373.0000 1072.0000 542.0000
## CV residual -83.5736  214.2948 281.0789
##
## Sum of squares = 650990    Mean square = 72332.23    n = 9
##
## Overall (Sum over all 9 folds)
##       ms
## 53586.08
```

```r
c
```

```
##       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
## 7  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
## 8  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
## 9  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
## 10 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
## 11 12.4  0 10.5 12.1 11.6 0.580  96.6 101 10.6 0.077 3.5   6570 17.0 0.016201
## 12 13.4  0 10.8  7.5  7.1 0.595  97.2  47  5.9 0.083 3.1   5800 17.2 0.031201
## 13 12.8  0 11.3  6.7  6.0 0.624  97.2  28  1.0 0.077 2.5   5070 20.6 0.045302
## 14 13.5  0 11.7  6.2  6.1 0.595  98.6  22  4.6 0.077 2.7   5290 19.0 0.053200
## 15 15.2  1  8.7  5.7  5.3 0.530  98.6  30  7.2 0.092 4.3   4050 26.4 0.069100
## 16 14.2  1  8.8  8.1  7.7 0.497  95.6  33 32.1 0.116 4.7   4270 24.7 0.052099
## 17 14.3  0 11.0  6.6  6.3 0.537  97.7  10  0.6 0.114 3.5   4870 16.6 0.076299
## 18 13.5  1 10.4 12.3 11.5 0.537  97.8  31 17.0 0.089 3.4   6310 16.5 0.119804
## 19 13.0  0 11.6 12.8 12.8 0.536  93.4  51  2.4 0.078 3.4   6270 13.5 0.019099
## 20 12.5  0 10.8 11.3 10.5 0.567  98.5  78  9.4 0.130 5.8   6260 16.6 0.034801
## 21 12.6  0 10.8  7.4  6.7 0.602  98.4  34  1.2 0.102 3.3   5570 19.5 0.022800
## 22 15.7  1  8.9  4.7  4.4 0.512  96.2  22 42.3 0.097 3.4   2880 27.6 0.089502
## 23 13.2  0  9.6  8.7  8.3 0.564  95.3  43  9.2 0.083 3.2   5130 22.7 0.030700
## 24 13.1  0 11.6  7.8  7.3 0.574 103.8   7  3.6 0.142 4.2   5400 17.6 0.041598
## 25 13.0  0 11.6  6.3  5.7 0.641  98.4  14  2.6 0.070 2.1   4860 19.6 0.069197
## 26 13.1  0 12.1 16.0 14.3 0.631 107.1   3  7.7 0.102 4.1   6740 15.2 0.041698
## 27 13.5  0 10.9  6.9  7.1 0.540  96.5   6  0.4 0.080 2.2   5640 13.9 0.036099
## 28 15.2  0 11.2  8.2  7.6 0.571 101.8  10  7.9 0.103 2.8   5370 21.5 0.038201
## 29 11.9  0 10.7 16.6 15.7 0.521  93.8 168  8.9 0.092 3.6   6370 15.4 0.023400
## 30 16.6  1  8.9  5.8  5.4 0.521  97.3  46 25.4 0.072 2.6   3960 23.7 0.075298
## 31 14.0  0  9.3  5.5  5.4 0.535 104.5   6  2.0 0.135 4.0   4530 20.0 0.041999
## 32 12.5  0 10.9  9.0  8.1 0.586  96.4  97  8.2 0.105 4.3   6170 16.3 0.042698
## 33 14.7  1 10.4  6.3  6.4 0.560  97.2  23  9.5 0.076 2.4   4620 23.3 0.049499
## 34 12.6  0 11.8  9.7  9.7 0.542  99.0  18  2.1 0.102 3.5   5890 16.6 0.040799
## 35 12.3  0 10.2  9.7  8.7 0.526  94.8 113  7.6 0.124 5.0   5720 15.8 0.020700
## 36 15.0  0 10.0 10.9  9.8 0.531  96.4   9  2.4 0.087 3.8   5590 15.3 0.006900
## 37 17.7  1  8.7  5.8  5.6 0.638  97.4  24 34.9 0.076 2.8   3820 25.4 0.045198
## 38 13.3  0 10.4  5.1  4.7 0.599 102.4   7  4.0 0.099 2.7   4250 22.5 0.053998
## 39 14.9  1  8.8  6.1  5.4 0.515  95.3  36 16.5 0.086 3.5   3950 25.1 0.047099
## 40 14.5  1 10.4  8.2  7.4 0.560  98.1  96 12.6 0.088 3.1   4880 22.8 0.038801
## 41 14.8  0 12.2  7.2  6.6 0.601  99.8   9  1.9 0.084 2.0   5900 14.4 0.025100
## 42 14.1  0 10.9  5.6  5.4 0.523  96.8   4  0.2 0.107 3.7   4890 17.0 0.088904
## 43 16.2  1  9.9  7.5  7.0 0.522  99.6  40 20.8 0.073 2.7   4960 22.4 0.054902
## 44 13.6  0 12.1  9.5  9.6 0.574 101.2  29  3.6 0.111 3.7   6220 16.2 0.028100
## 45 13.9  1  8.8  4.6  4.1 0.480  96.8  19  4.9 0.135 5.3   4570 24.9 0.056202
## 46 12.6  0 10.4 10.6  9.7 0.599  98.9  40  2.4 0.078 2.5   5930 17.1 0.046598
## 47 13.0  0 12.1  9.0  9.1 0.623 104.9   3  2.2 0.113 4.0   5880 16.0 0.052802
##       Time Crime Predicted    cvpred fold
## 1  26.2011   791  810.8255  785.3647    1
## 2  25.2999  1635 1387.8082 1355.7097    5
## 3  24.3006   578  386.1368  345.3417    1
```

```
## 4  29.9012  1969 1897.1866 1882.7381    2
## 5  21.2998  1234 1269.8420 1266.7954    3
## 6  20.9995   682  730.2659  781.7557    2
## 7  20.6993   963  733.3799  759.9655    4
## 8  24.5988  1555 1353.5532 1243.1763    3
## 9  29.4001   856  718.7568  723.5331    3
## 10 19.5994   705  787.2712  723.6678    5
## 11 41.6000  1674 1117.7702  946.1309    3
## 12 34.2984   849  673.3766  684.3525    2
## 13 36.2993   511  739.3727  770.2015    4
## 14 21.5010   664  713.5639  730.0555    4
## 15 22.7008   798  828.3418  826.2855    3
## 16 26.0991   946 1004.3984 1046.8197    5
## 17 19.1002   539  527.3659  492.2016    1
## 18 18.1996   929  800.0046  700.5751    1
## 19 24.9008   750 1220.6767 1240.2916    1
## 20 26.4010  1225 1202.9607 1247.8616    4
## 21 37.5998   742  783.2733  819.7114    5
## 22 37.0994   439  728.3110  701.5126    1
## 23 25.1989  1216  937.5703  754.2511    3
## 24 17.6000   968  919.3912  953.7248    4
## 25 21.9003   523  579.0638  621.3745    2
## 26 22.1005  1993 1789.1406 1794.6456    5
## 27 28.4999   342  312.2047  297.1932    4
## 28 25.8006  1216 1259.0034 1238.3192    2
## 29 36.7009  1043 1495.4856 1663.6272    5
## 30 28.3011   696  668.0161  638.8712    4
## 31 21.7998   373  440.4394  456.5736    5
## 32 30.9014   754  773.6840  788.0343    2
## 33 25.5005  1072  873.8469  857.7052    5
## 34 21.6997   923  997.5498 1013.9253    2
## 35 37.4011   653  808.0296  850.6961    4
## 36 44.0004  1272 1101.7167 1127.3318    1
## 37 31.6995   831  991.5623 1076.5799    3
## 38 16.6999   566  544.3733  544.6990    1
## 39 27.3004   826  786.6949  717.0989    3
## 40 29.3004  1151 1140.7906 1168.2111    1
## 41 30.0001   880  796.4198  778.0437    2
## 42 12.1996   542  368.7031  260.9211    5
## 43 31.9989   823 1016.5503 1079.7748    3
## 44 30.0001  1030 1177.5973 1159.3155    2
## 45 32.5996   455  621.8592  690.6802    4
## 46 16.6999   508  748.4256  807.6968    2
## 47 16.0997   849  976.4397 1038.3321    3
```

```r
# We can calculate the R-squared values directly.
# R-squared = 1 - SSEresiduals/SSEtotal

# total sum of squared differences between data and its mean
SStot <- sum((dat$Crime - mean(dat$Crime))^2)

# for model, model2, and cross-validation, calculated SEres

SSres_model <- sum(model$residuals^2)
```

```r
SSres_model2 <- sum(model2$residuals^2)

SSres_c <- attr(c,"ms")*nrow(dat) # mean squared error, times number of data points, gives sum of squar

# Calculate R-squareds for model, model2, cross-validation

1 - SSres_model/SStot # initial model with insignificant factors
```

```
## [1] 0.8030868
```

```r
1 - SSres_model2/SStot # model2 without insignificant factors
```

```
## [1] 0.7658663
```

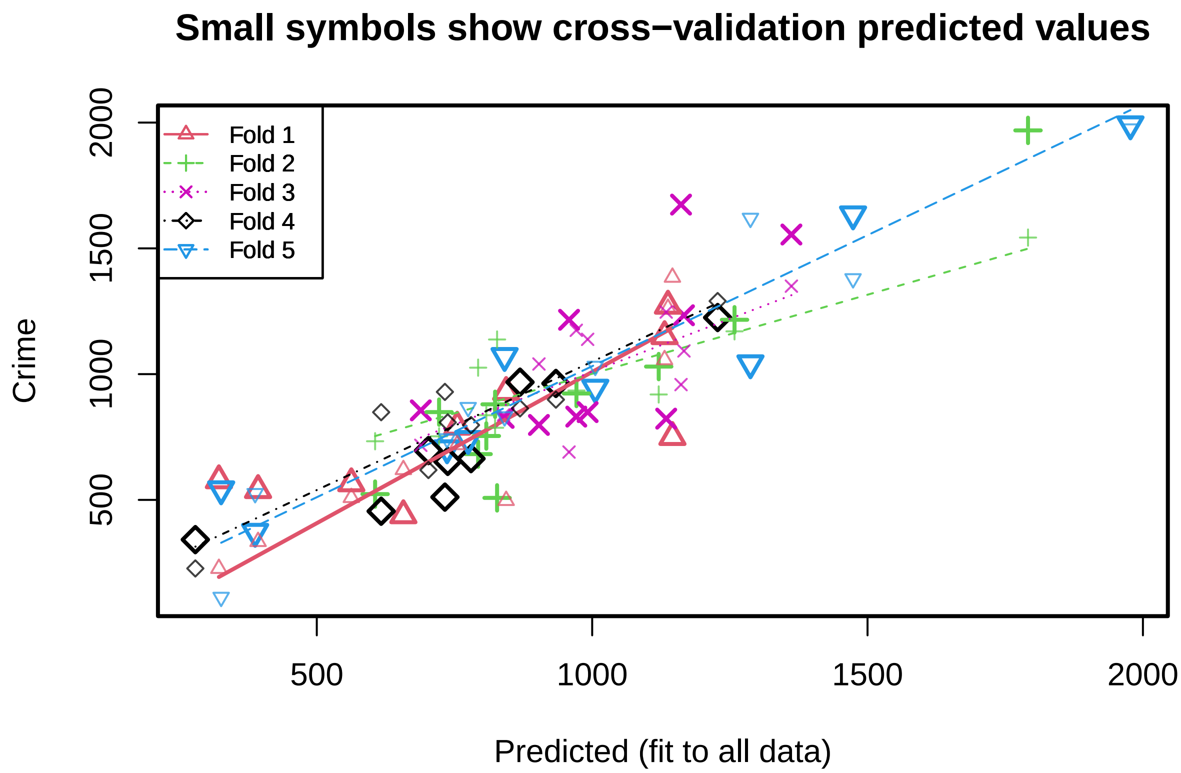
```r
1 - SSres_c/SStot # cross-validated
```

```
## [1] 0.6339817
```

```r
# We can also try cross-validation on the first, 15-factor model

cfirst <- cv.lm(dat,model,m=5)
```

```
## Warning in cv.lm(dat, 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
## Predicted   755.03222 322.2615 393.3633 843.8072 1145.7379  657.2092 1137.61711
## cvpred      719.48189 227.3811 334.2928 497.4904 1384.9349  620.1834 1261.61602
## Crime       791.00000 578.0000 539.0000 929.0000  750.0000  439.0000 1272.00000
## CV residual  71.51811 350.6189 204.7072 431.5096 -634.9349 -181.1834   10.38398
##                   38         40
## Predicted   562.6934 1131.45326
## cvpred      509.0826 1057.08701
## Crime       566.0000 1151.00000
## CV residual  56.9174   93.91299
##
## Sum of squares = 804290.7    Mean square = 89365.64    n = 9
##
## fold 2
## Observations in test set: 10
##                     4         6        12        25          28        32
## Predicted   1791.3619  792.9301 722.04080  605.8824 1258.48423 807.81667
## cvpred      1542.8663 1025.6864 752.84607  733.1797 1170.10415 836.60938
## Crime       1969.0000  682.0000 849.00000  523.0000 1216.00000 754.00000
## CV residual  426.1337 -343.6864  96.15393 -210.1797   45.89585 -82.60938
##                   34        41        44        46
## Predicted   971.45581 823.74192 1120.8227  827.3543
## cvpred      934.62797 786.74042  919.1066 1137.6778
```

```
## Crime       923.00000 880.00000 1030.0000  508.0000
## CV residual -11.62797  93.25958  110.8934 -629.6778
##
## Sum of squares = 779686.2    Mean square = 77968.62    n = 10
##
## fold 3
## Observations in test set: 10
##                     5         8         9        11        15        23
## Predicted   1166.6840 1361.7468 688.8682 1161.3291  903.3541  957.9918
## cvpred      1092.1924 1349.7715 717.0401  958.3058 1040.2775  690.2073
## Crime       1234.0000 1555.0000 856.0000 1674.0000  798.0000 1216.0000
## CV residual  141.8076  205.2285 138.9599  715.6942 -242.2775  525.7927
##                    37       39        43        47
## Predicted    971.1513 839.2864 1134.4172  991.7629
## cvpred      1174.2195 838.1895 1246.7022 1138.2873
## Crime        831.0000 826.0000  823.0000  849.0000
## CV residual -343.2195 -12.1895 -423.7022 -289.2873
##
## Sum of squares = 1310071    Mean square = 131007.1    n = 10
##
## fold 4
## Observations in test set: 9
##                     7        13        14          20       24       27
## Predicted   934.16366  732.6412  780.0401 1227.83873 868.9805 279.4772
## cvpred      898.53488  929.2776  797.4106 1290.40739 863.7702 227.4408
## Crime       963.00000  511.0000  664.0000 1225.00000 968.0000 342.0000
## CV residual  64.46512 -418.2776 -133.4106  -65.40739 104.2298 114.5592
##                    30        35        45
## Predicted   702.69454  737.7888  616.8983
## cvpred      618.72406  808.0845  848.6350
## Crime       696.00000  653.0000  455.0000
## CV residual  77.27594 -155.0845 -393.6350
##
## Sum of squares = 410147.4    Mean square = 45571.93    n = 9
##
## fold 5
## Observations in test set: 9
##                     2        10          16        21          26        29
## Predicted   1473.6764 736.50802 1005.65694  774.8506 1977.37067 1287.3917
## cvpred      1379.5108 743.27567 1031.35676  867.6315 1975.12567 1619.8299
## Crime       1635.0000 705.00000  946.00000  742.0000 1993.00000 1043.0000
## CV residual  255.4892 -38.27567  -85.35676 -125.6315   17.87433 -576.8299
##                    31        33       42
## Predicted    388.0334  840.9992 326.3324
## cvpred       525.4791  830.6871 112.9800
## Crime        373.0000 1072.0000 542.0000
## CV residual -152.4791  241.3129 429.0200
##
## Sum of squares = 688401.1    Mean square = 76489.01    n = 9
##
## Overall (Sum over all 9 folds)
##       ms
## 84948.87
```

```r
SSres_cfirst <- attr(cfirst,"ms")*nrow(dat) # mean squared error, times number of data points, gives su

1 - SSres_cfirst/SStot # cross-validated
```

```
## [1] 0.419759
```

```r
# glm() is a more-general function for regression.

g <- glm(Crime ~ . , data=dat, family="gaussian")
summary(g)
```

```
##
## Call:
## glm(formula = Crime ~ ., family = "gaussian", data = dat)
##
## Deviance 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) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 43707.93)
##
##     Null deviance: 6880928  on 46  degrees of freedom
## Residual deviance: 1354946  on 31  degrees of freedom
## AIC: 650.03
##
## Number of Fisher Scoring iterations: 2
```

```r
g2 <- glm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob , data=dat, family="gaussian")
summary(g2)
```

```
##
```

```
## Call:
## glm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, family = "gaussian",
##     data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -470.68   -78.41   -19.68   133.12   556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## M             105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    1528.10  -2.488  0.01711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 40276.42)
##
##     Null deviance: 6880928  on 46  degrees of freedom
## Residual deviance: 1611057  on 40  degrees of freedom
## AIC: 640.17
##
## Number of Fisher Scoring iterations: 2
```

```r
library(boot)

cg <- cv.glm(dat,g,K=5) # note that here, K is the number of folds
cg2 <- cv.glm(dat,g2,K=5)

# mean squared error is cg$delta[1]
# depending on random seed, this could be different;

1 - cg$delta[1]*nrow(dat)/SStot
```

```
## [1] 0.4792911
```

```r
1 - cg2$delta[1]*nrow(dat)/SStot
```

```
## [1] 0.6719836
```