ISYE Homework 5

2024-09-24

R Markdown

Question 8.1 Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

I used to work for Social Security so analyzing factors that influence the time taken to process claims is something in which a linear regression model would be appropriate

Predictors:

Complexity of the Claim: A measure of how complicated the claim is, potentially affecting processing time.

Experience Level of the Employee: The number of years an employee has been working in claims processing could influence efficiency.

Number of Claims in the Queue: The total number of claims waiting to be processed could affect the time each individual claim takes.

Training Received: The amount or quality of training an employee has received might correlate with processing speed.

Question 8.2 Using crime data from http://www.statsci.org/data/general/uscrime.txt (file uscrime.txt, description at http://www.statsci.org/data/general/uscrime.html), use regression (a useful R function is lm or glm) to predict the observed crime rate in a city with the following data: 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 Show your model (factors used and their coefficients), the software output, and the quality of fit. Note that because there are only 47 data points and 15 predictors, you'll probably notice some overfitting. We'll see ways of dealing with this sort of problem later in the course.

The approach I used is as follows: Firstly, view each of the predictors. Then, I can create a linear regression model using all predictors and evaluate the data. Next, filter out some predictors based on their p-value. I can then create a new linear regression model using only the significant predictors. I can compare these two models based on their adjusted R^2 values. Finally, I can perform cross validation on both models to check the quality of fitting.

As always, call necessary libraries and import the data set. I re-organized the data by moving the 'Crime' column to the first column. I do this because when creating a model formula for linear regression (lm()), the format is "response ~ predictors". This is simply to make things easier when calling the function and writing the code.

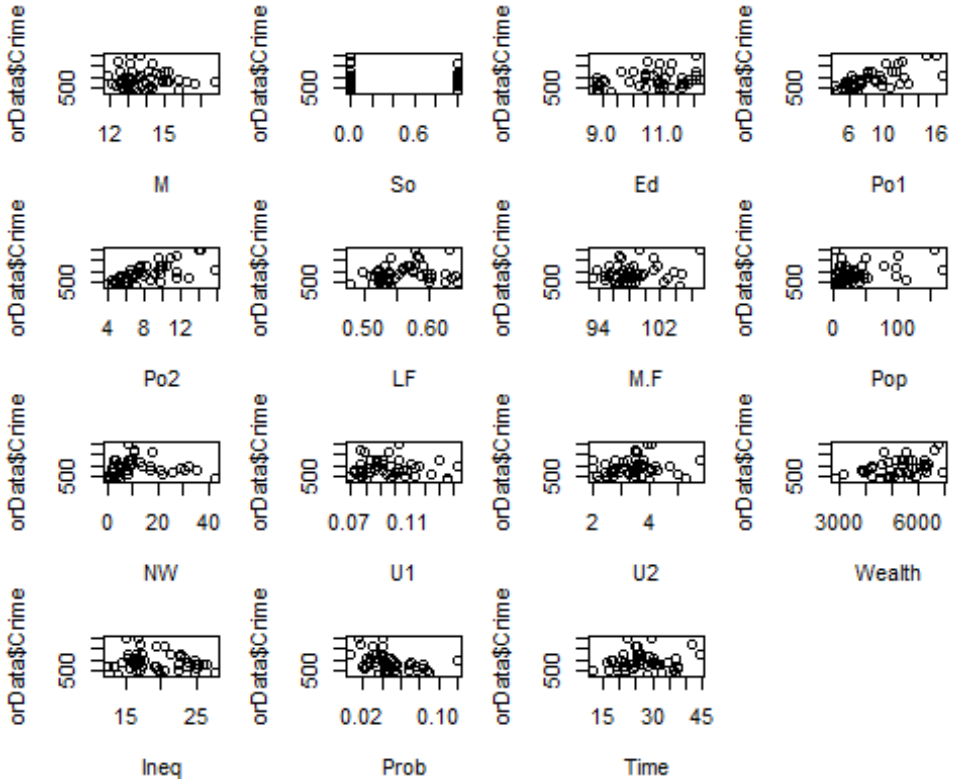
```r
#Question 8.2
#housekeeping
library(pacman)
pacman::p_load(rio, stats, DAAG)

#import data
data <- import("D:/… /uscrime.txt")
#swap column to fit formula (crime is first column)
orData <- data[c(16, 1:15)]
```

I create a subset ‘pred’ that contains all columns except the first (Crime), meaning ‘pred’ contains all of the predictor variables. ‘headers’ is a list that corresponds to each of the predictor variables. I prepared a plotting widow to display the 15 predictors in a 4x4 grid. Looping through each predictor column in ‘pred’, a plot is made and labeled with the appropriate variable name from the ‘headers’ list. Each predictor is plotted against the response variable ‘Crime’. The main purpose here is to visualize the relationship between each predictor and the response variable. Exploring the data this way allows me to not only understand the aforementioned relationship but also to identify patterns, spot outliers, and understand correlation (or lack of). In linear regression, it is assumed that the relationship between predictor and response variable is linear (as the name implies). By creating the scatterplots, we can visually see if each predictor has a linear relationship vs ‘Crime’. This also gives me a sense of the predictive power of each variable. Predictors with an obvious trend are more likely to be useful in the regression model. It is also useful for detecting multicollinearity (predictors that are highly correlated with each other), and high multicollinearity can cause issues in regression models.

```r
#subset pred. plot the predictors/response
pred = orData[-1] #predictor variables
headers = list(
  "M",
  "So",
  "Ed",
  "Po1",
  "Po2",
  "LF",
  "M.F",
  "Pop",
  "NW",
  "U1",
  "U2",
  "Wealth",
  "Ineq",
  "Prob",
  "Time"
)
par(mfrow = c(4, 4), mar = c(4, 4, 2, 1)) #15 plots for each variable (4x4)
for (i in 1:15) {
```

```
  plot(pred[, i], orData$Crime, xlab = headers[i])
}
```



From the plots above, we can assume that variables like 'So' would be excluded, or that Po1, Po2, show a strong linear relationship. It looks like a lot of these predictors do not correlate much when plotted against 'Crime'.

I created a new data frame with the values given in the homework question.

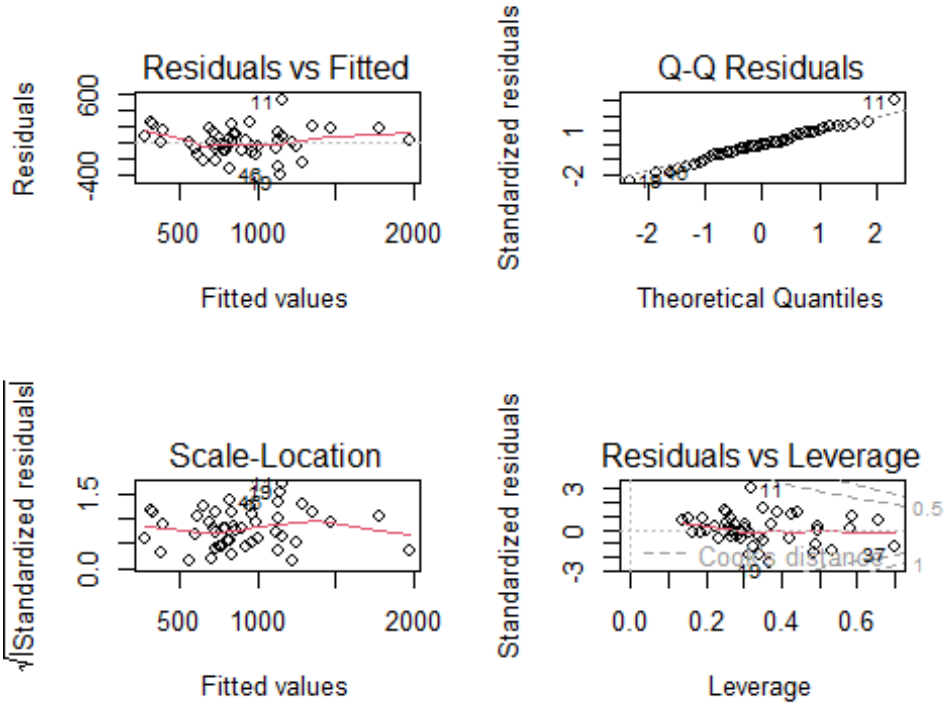
```
#dataframe with pre-defined values of the socio-economic factors
cityData = data.frame(
  M = 14.0,
  So = 0,
  Ed = 10.0,
  Po1 = 12.0,
  Po2 = 15.5,
  LF = 0.640,
  M.F = 94.0,
  Pop = 150,
  NW = 1.1,
  U1 = 0.120,
  U2 = 3.6,
  Wealth = 3200,
  Ineq = 20.1,
  Prob = 0.04,
  Time = 39.0
)
```

Now I'll fit the first linear regression model. This model will use all of the predictors and will automatically use 'Crime' as the response variable (since we moved it to the first column).

```
#fitting first linear regression model
formula1 <- formula(orData)
model1 <- lm(formula1, orData)
summary(model1) #focusing on r^2, pvalue, and coeff for predictors

##
## Call:
## lm(formula = formula1, data = orData)
##
## 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

coef1 <- model1$coefficients #coeff (beta values)
par(mfrow = c(2, 2))
plot(model1) #diagnostic plots to assess fit
```




```
crimePred <- predict.lm(model1, cityData) #use model to predict crime rate on predefined data (cityData)
crimePred

##        1
## 155.4349
```

'model1' is our first linear regression model. The summary tells me important values such as the coefficients, R-Squared values, and p-values for the predictors. Residuals are the differences between the observed values of the response variable 'Crime' and the values predicted by the model.

Residuals close to zero mean the predictions are accurate, while large residuals indicate that the model's predictions are far off for some observations.

The coefficients section provides estimates for each predictor in the model, along with standard errors, t-values, and p-values. Estimate represents the estimated effect of each predictor on the response variable. Std. Error shows the uncertainty around the estimate. A smaller standard error means the estimate is more precise. t value tests whether the coefficient is significantly different from zero. Finally, p-value tests the statistical significance of the predictor. A p-value less than 0.05 indicates that the predictor is significant.

In my output, the predictors M, Ed, Ineq and Prob are deemed significant (low p-value and multiple stars for significance codes). However, I also need to look closer at the significance codes. The stars indicate the level of significance. Even though, Po1 and U2

are slightly above 0.05, they are deemed “borderline significant” (denoted by the ‘.’), so I will include them as necessary predictors.

Finally, the R-Squared(0.8031) and Adjusted R-Squared(0.7078) values have a sizable difference between them, indicating that the model was overfitted. The output of ‘crimePred’ is 155, which falls on the lower end of what was expected.

This means for out next linear regression model, I will filter out any predictor with a p-value < 0.08, so that only the significant predictors mentioned are used. This new formula is: Crime ~ M + Ed + Ineq + Prob + U2 + Po1

Before doing that, I’ll go over the plots for ‘model1’. Residual vs Fitted checks for patterns in residuals. If the model fits well, the residuals should appear randomly scattered around zero. The Q-Q compares the distribution of the residuals to a normal distribution, since following a normal distribution is one of the assumptions in linear regression. In a good model, the points should follow a straight line, which is what is shown in that plot. Scale-Location shows the square root of the absolute residuals against the fitted values. This checks for homoscedasticity—the assumption that the residuals have constant variance across all levels of fitted values. Ideally, the plot should display a horizontal line with random scatter. A funnel shape (i.e., the spread of points increases or decreases), this indicates heteroscedasticity (non-constant variance), which can affect the model’s reliability. Lastly, Residual vs Leverage identifies points with high leverage (i.e., data points that have extreme predictor values) and outliers (points with large residuals).

Now that I know which predictors I want to keep, I will filter out the rest.

```
#filter pred by p-value
pvalue <- summary(model1)$coefficients[, 4] #extract p value from summary
coef <- model1$coefficients
orDataFitted <- orData[1]
n <- 2
for (i in 2:16) {
  if (pvalue[i] < 0.08) { #remove if pvalue if > 0.08, keeping U2 and Po1
    orDataFitted[n] = orData[i]
    n = n + 1
  }
}
```

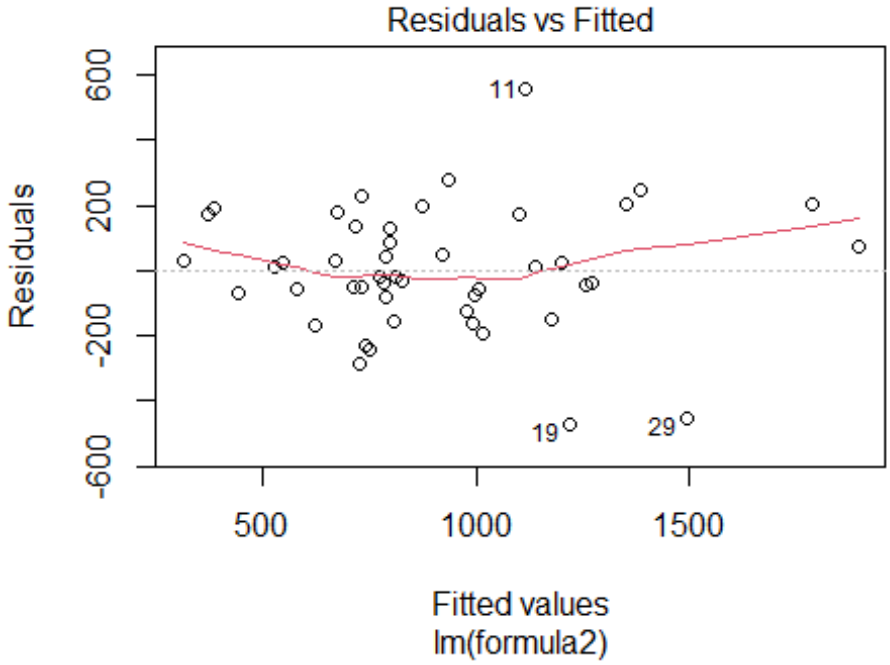
This removes any predictor with a p-value over 0.08, effectively giving me the formula

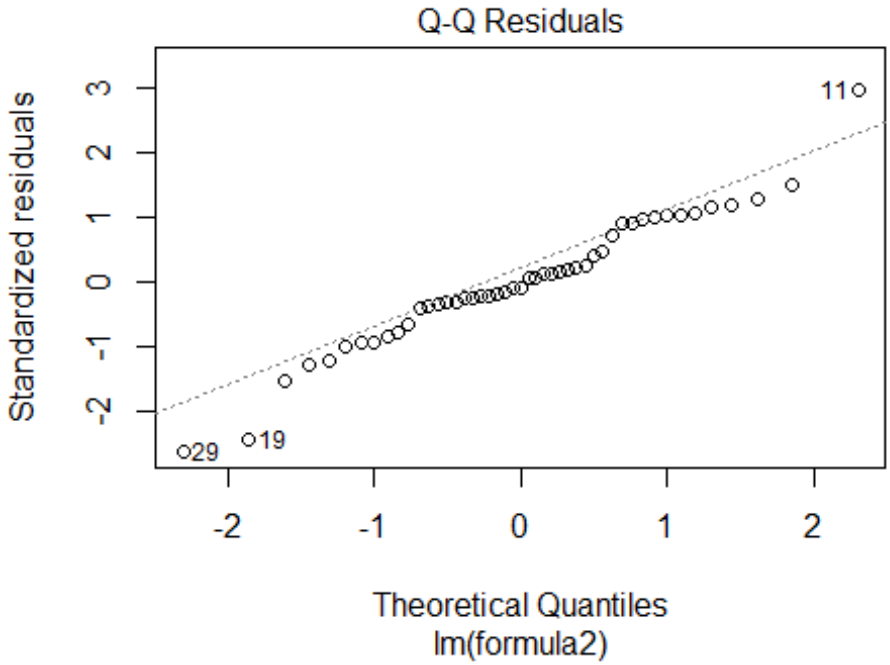
Crime ~ M + Ed + Ineq + Prob + U2 + Po1 I will now use this formula for a new linear regression model.

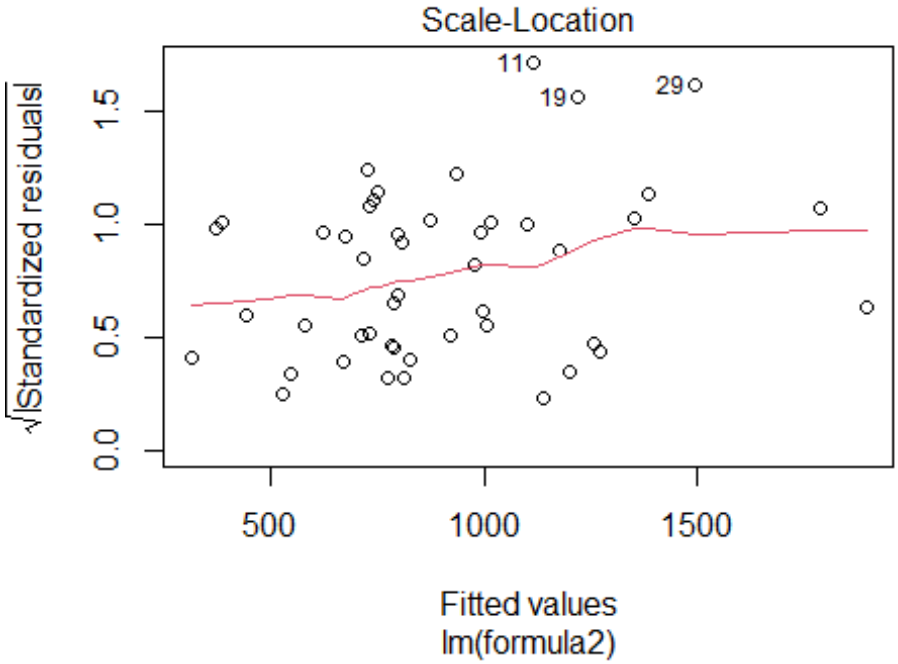
```
#fitting second linear regression model (filtered pred)
formula2 <- formula(orDataFitted) #second formula using filtered pred
model2 <- lm(formula2, orDataFitted)
summary(model2)
```

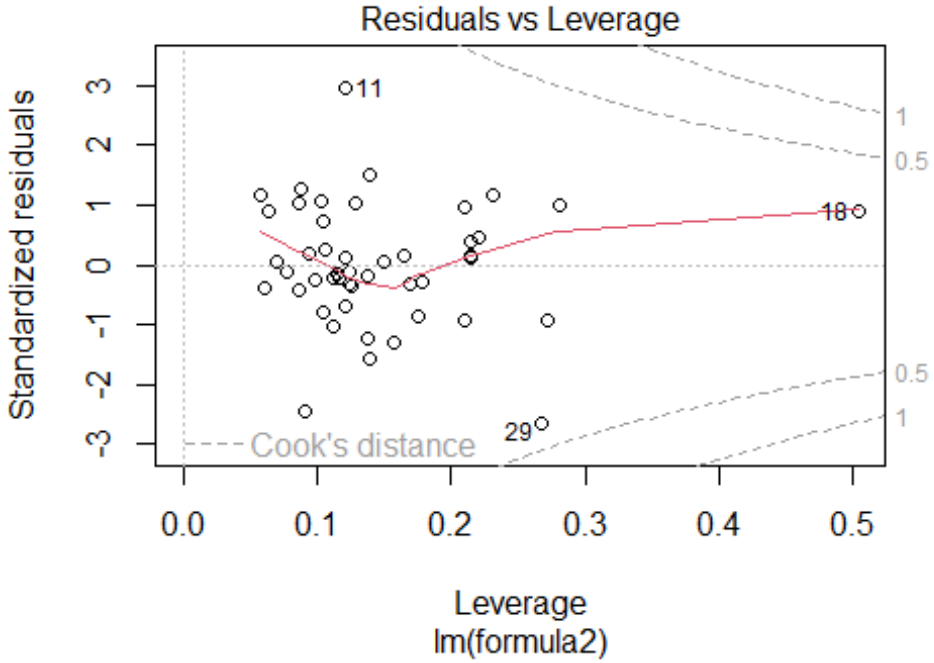
```
##
## Call:
## lm(formula = formula2, data = orDataFitted)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
## M             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

plot(model2)
```


Residuals vs Fitted
Residuals
600
200
-200
-600
11
19
29
500
1000
1500
Fitted values
lm(formula2)


Q-Q Residuals
Standardized residuals
3
2
1
0
-1
-2
11
29
19
-2
-1
0
1
2
Theoretical Quantiles
lm(formula2)






```
crimePredAdj <- predict.lm(model2, cityData)
crimePredAdj
```

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

This second linear regression model only uses the filtered predictors. Again, I print out the summary and diagnostic plots. The new predicted crime rate, 'crimePredAdj', is valued at 1304, which is more realistic. The p-values for the predictors are also all low. The difference between R Squared(0.7659) and Adjusted R-Squared(0.7307) has also reduced. The prediction of 1304 is closer to the actual/expected crime rate observed in the dataset. The predictors now have a stronger statistical significance, with lower p-values, indicating that they are more reliable in explaining the variation in crime rates.

Coefficients:

M (percent of males aged 14–24) has a significant positive coefficient of 105.02, meaning that an increase in this percentage is associated with an increase in crime rate.

Ed (average years of schooling) also has a strong positive effect, with a coefficient of 196.47.

Po1 (police per 100,000 residents in 1960) is highly significant, with a coefficient of 115.02, suggesting that the presence of police has a positive effect on crime

Prob (probability of arrest) has a negative coefficient of -3801.84, implying that a higher probability of arrest is associated with a reduction in crime rates.

U2 (unemployment rate of urban males aged 35-39) has a significant positive coefficient of 89.37, indicating that an increase in this unemployment rate is associated with an increase in crime rates.

Ineq (income inequality) has a significant positive coefficient of 67.65, suggesting that higher income inequality is also associated with increased crime rates

Finally, this model has an Adjusted R-squared value of 0.7307, meaning that about 73% of the variability in crime rates is explained by the model. This is still a good fit, and slightly higher than the first model's 0.7078

The residual standard error of 200.7 indicates that there is still some variability in the data not captured by the model, but overall the model is performing well.

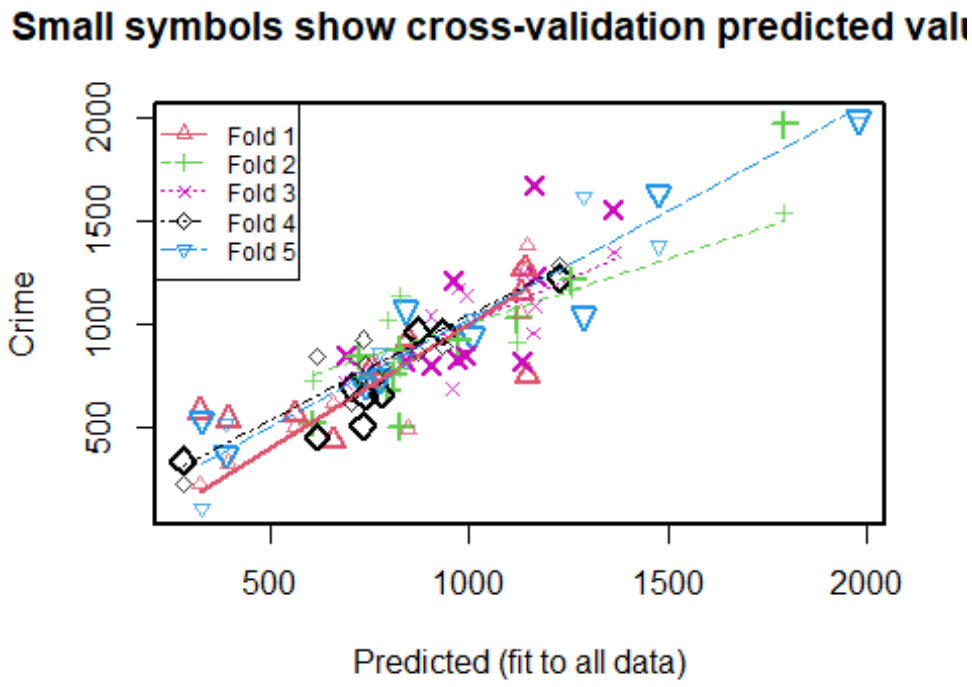
To check the quality of the fit for the two models, I can compare the R-Square/Adjusted R-Square values. Model1 has a higher R-Square value (0.8031 vs 0.7659) but Model2 has a higher Adjusted R-Square Value (0.7078 vs 0.7307). Since the Adjusted R-Squared value helps evaluate how well fit a model is and accounts for the number of predictors, this should be the value we go off of, meaning Model2 is better. I can further check the quality of this by cross validation.

Perform cross validation (5-folds) on both models. The data is split into 5 subsets, and the model is trained on 4 of them while being tested on the 5th, repeating this process for each subset. calculate total sum of squares, then the sum of squares for each model. After

dividing the individual by total, I can get the R-Squared values for the cross validation models

```
#cross validate on both models
par(mfrow = c(1, 1))
CrossVal1 <- cv.lm(orData, model1, m = 5)

## Warning in cv.lm(orData, model1, 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
```



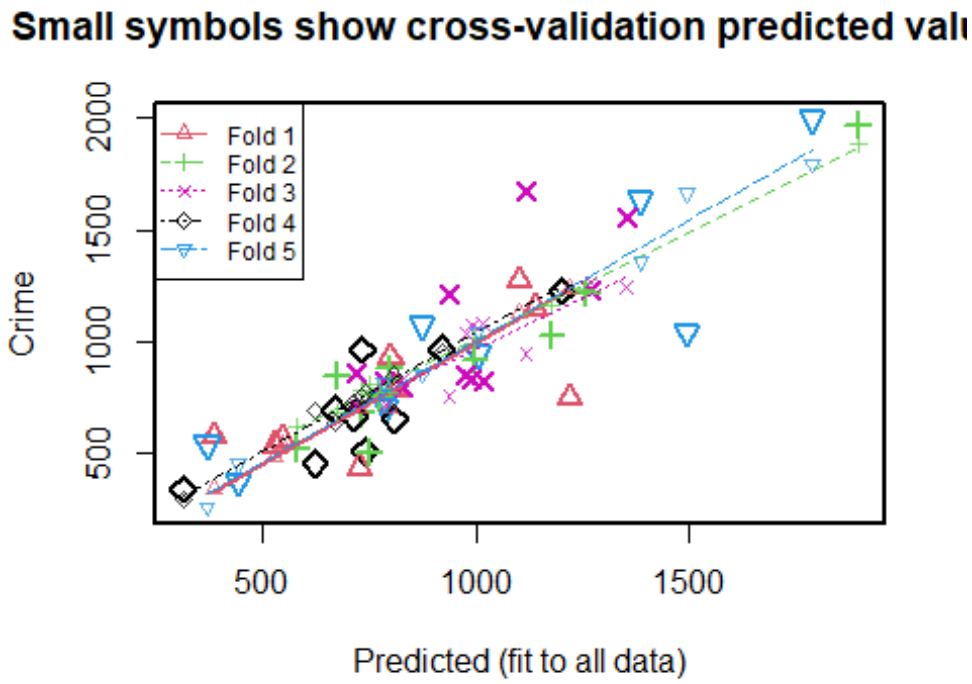

```
##
## fold 1
## Observations in test set: 9
##                      1        3       17       18        19        22
36
## Predicted   755.03222 322.2615 393.3633 843.8072 1145.7379  657.2092
1137.61711
## cvpred      719.48189 227.3811 334.2928 497.4904 1384.9349  620.1834
1261.61602
## Crime       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

CrossVal2 <- cv.lm(orData, model2, m = 5)

## Warning in cv.lm(orData, 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
```

```
#compare R^2 values of CV models
TotalSumSquare <- sum((orData$Crime - mean(orData$Crime)) ^ 2) #Total SS for
Crime variable
SumSquare1 <- attr(CrossVal1, "ms") * nrow(orData) # ss for cv model 1
SumSquare2 <- attr(CrossVal2, "ms") * nrow(orData) # ss for cv model 2
RSquareCV1 = 1 - SumSquare1 / TotalSumSquare # r^2 cv model 1
RSquareCV2 = 1 - SumSquare2 / TotalSumSquare # r^2 cv model 2
RSquareCV1
```

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

```
RSquareCV2
```

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

Cross validation model1 yields 0.419759 while cross validation model2 yields 0.6339817. Model2 is still the better model but the previous value of 0.7307 may have been too optimistic.