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.

Working in financial advising, I would use a linear regression model to predict the annual portfolio growth rate for a client based on several factors. This model could help guide investment strategy decisions like rebalancing the portfolio or adjusting the risk level based on a client's preferences and financial goals. Below are 5 predictors I would consider using:

1. Initial Portfolio Value: The starting value of the investment, as larger portfolios might perform differently due to economies of scale or diversification.
2. Asset Allocation (% in Stocks, Bonds, etc.): The percentage of the portfolio allocated to various asset classes like equities, bonds, and cash, which can significantly impact returns.
3. Risk Tolerance (Score or Category): A measure of how much risk the client is willing to take, which might affect the aggressiveness of the investment strategy (my firm uses a 1 – 10 scale).
4. Market Volatility: Historical or expected volatility in the market, which can influence returns, especially for portfolios heavy in equities.
5. Time Horizon: The length of time the investment is expected to be held, as longer horizons typically allow for compounding and recovery from market downturns.

By using a model with these factors, I would be able to provide insight to clients on what their estimated portfolio growth rate should look like.

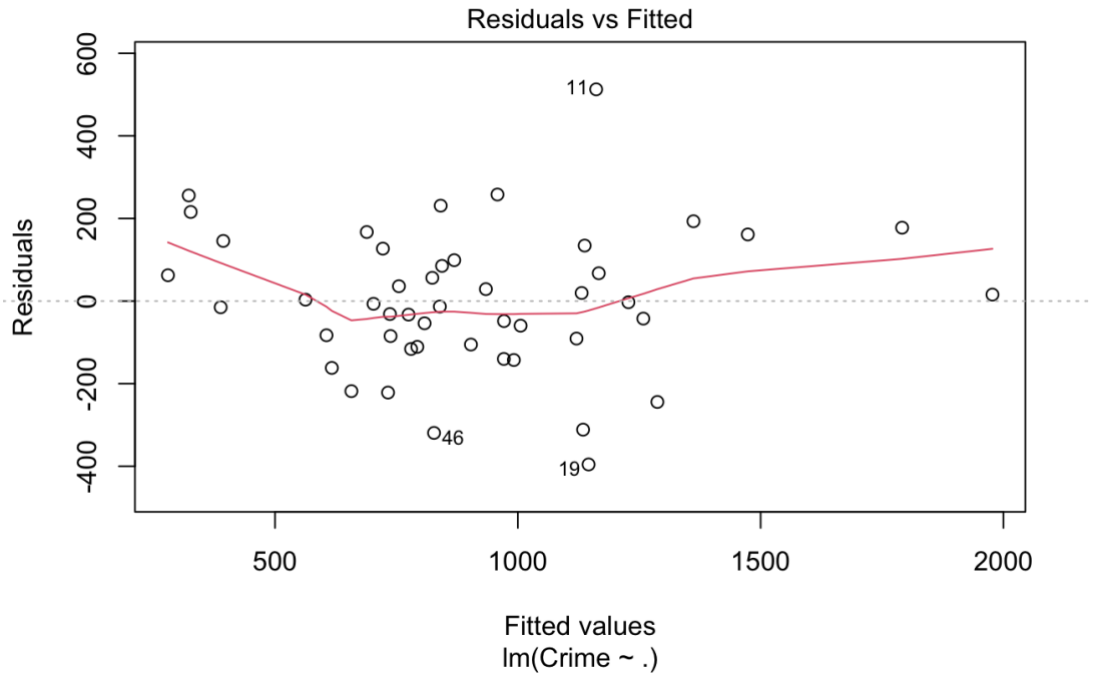
Question 8.2

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:

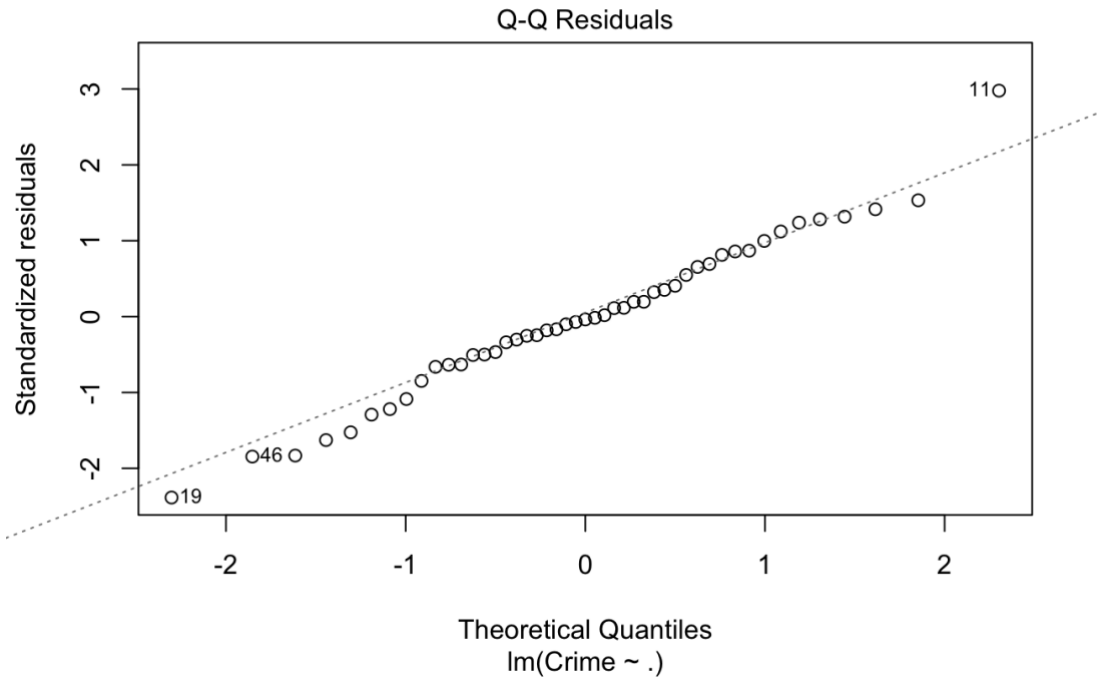
M = 14.0
So = 0
Ed = 10.0
Po1 = 12.0
Po2 = 15.5

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.

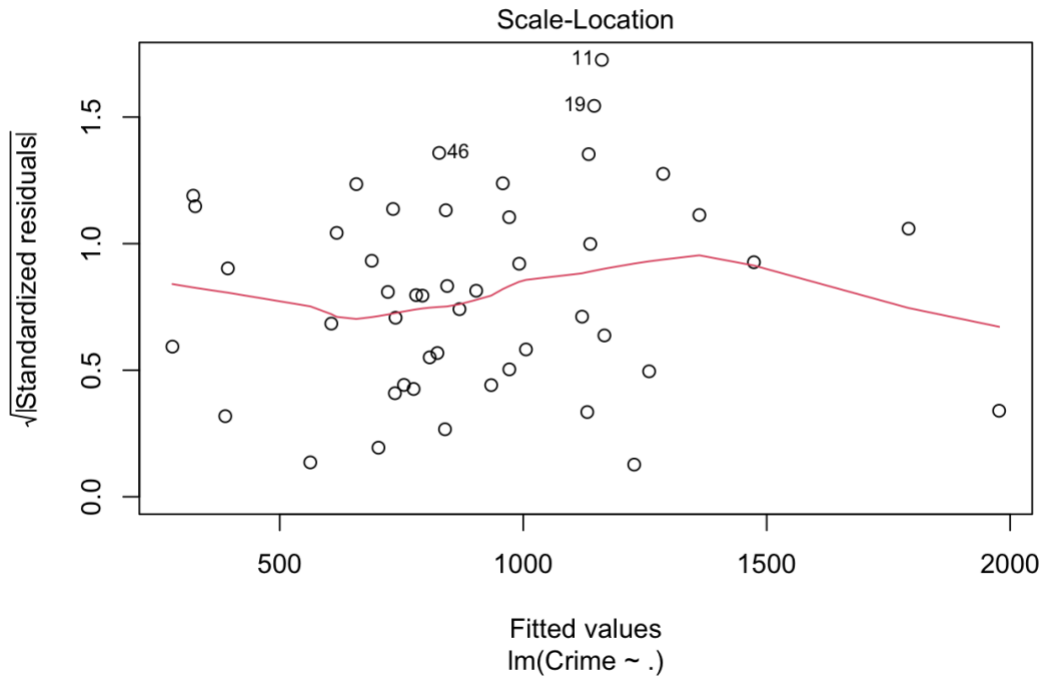
After setting the seed and reading the data file, I looked at the first 6 records of what is in the data file. For the linear regression aspect of the testing, I decided to use the "lm" function and not the "glm" function. There was really no reason for the decision, and I ended up going back after I ran everything and tested it with the "glm" function. In the "lm" function, the "Crime" column of the data file was the response variable that I wanted, so in my model I tested all columns except the "Crime" column. I then plotted this model, which resulted in 4 different graphs. The "Results vs Fitted" graph shows that the residuals are distributed evenly for the most part across 0 Residuals (the horizontal length). Residuals in this case is the difference between what is in the observations in the data set and what is predicted.



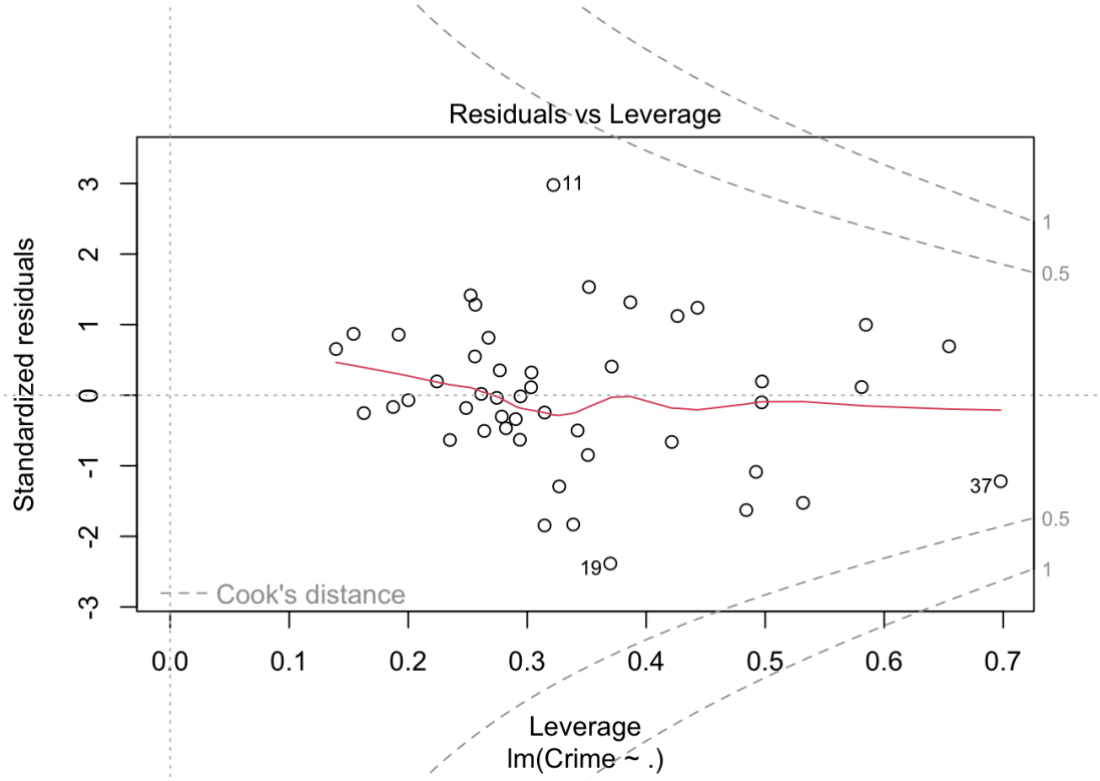
The “Q-Q Residuals” graph shows if the data is normally distributed or not. Between one standard deviation, it is shown that pretty much all the data is concentrated. It is not perfectly normally distributed, but for this test it is enough.



The “Scale-Location” graph is a scaled version of the “Residuals vs Fitted” graph. It standardizes the residuals to fit them. The standardized residuals are more sensitive to the data than the non-standardized ones, which is why the graph is different. Like the first graph, all the x values look to be independent of each other.



The “Residuals vs Leverage” graph shows that the points that are further away from the 0 line are high influential points. Cooks distance tells you if any point lies beyond the dashed 0.5 or 1 line, they could hurt the model if it is an outlier. This graph shows that there are no points that lie beyond that range, so there are likely no outliers that will hurt the model. If there are outliers that are beyond that range, the model will try to fit them into it, which hurts the pattern it is trying to form.



Looking at the 4 of these graphs, it shows that a linear regression model can be used. After seeing all of this, I took a summary of the model. From the summary, I could see which attribute was important and which was not. The ones that are important had lower p values. The following was the output:

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

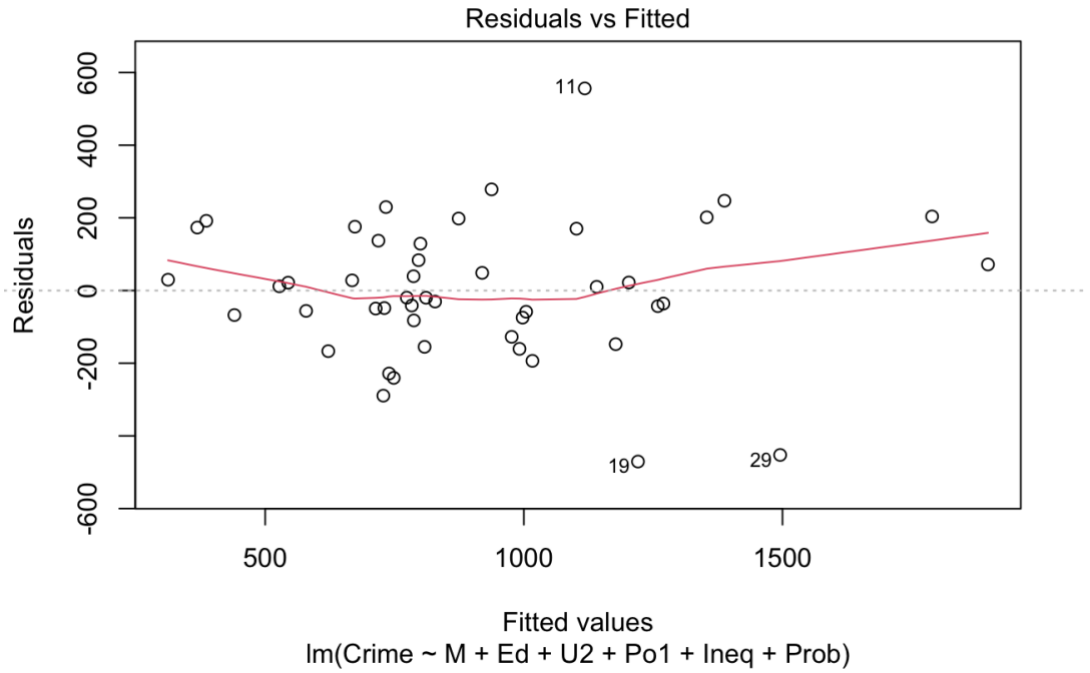
Residual standard error: 209.1 on 31 degrees of freedom

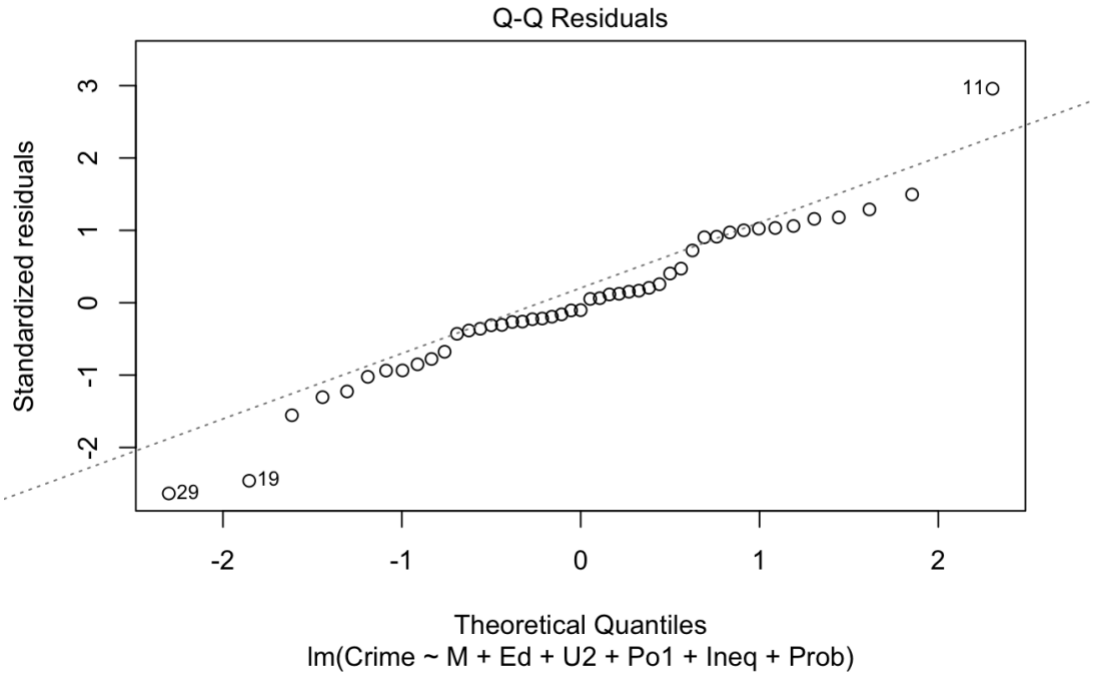
Multiple R-squared: 0.8031, Adjusted R-squared: 0.7078

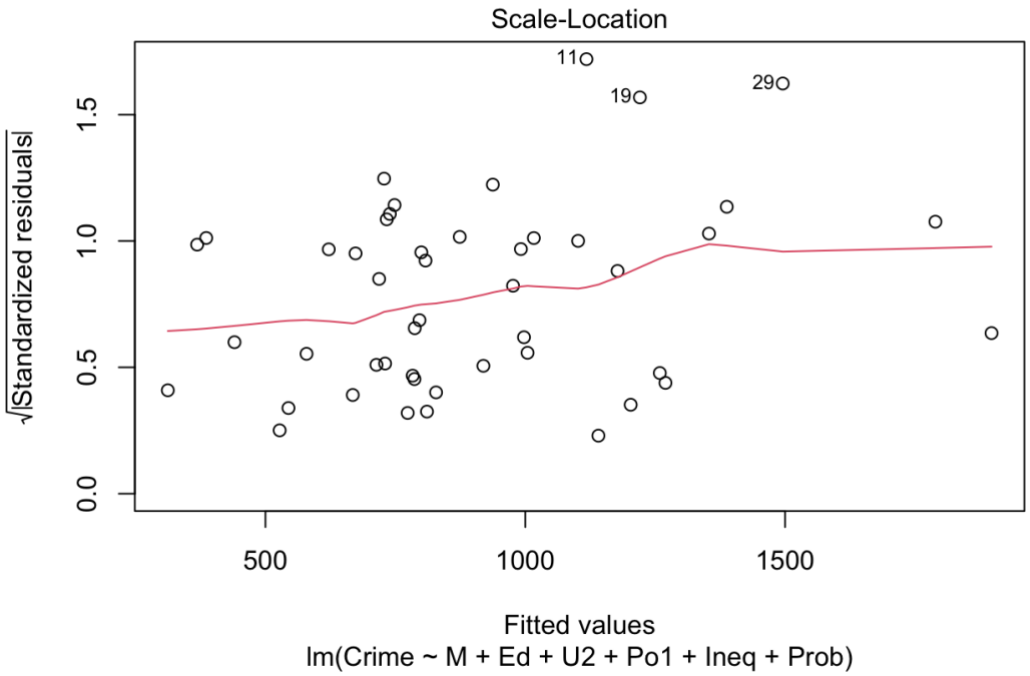
F-statistic: 8.429 on 15 and 31 DF, p-value: 3.539e-07

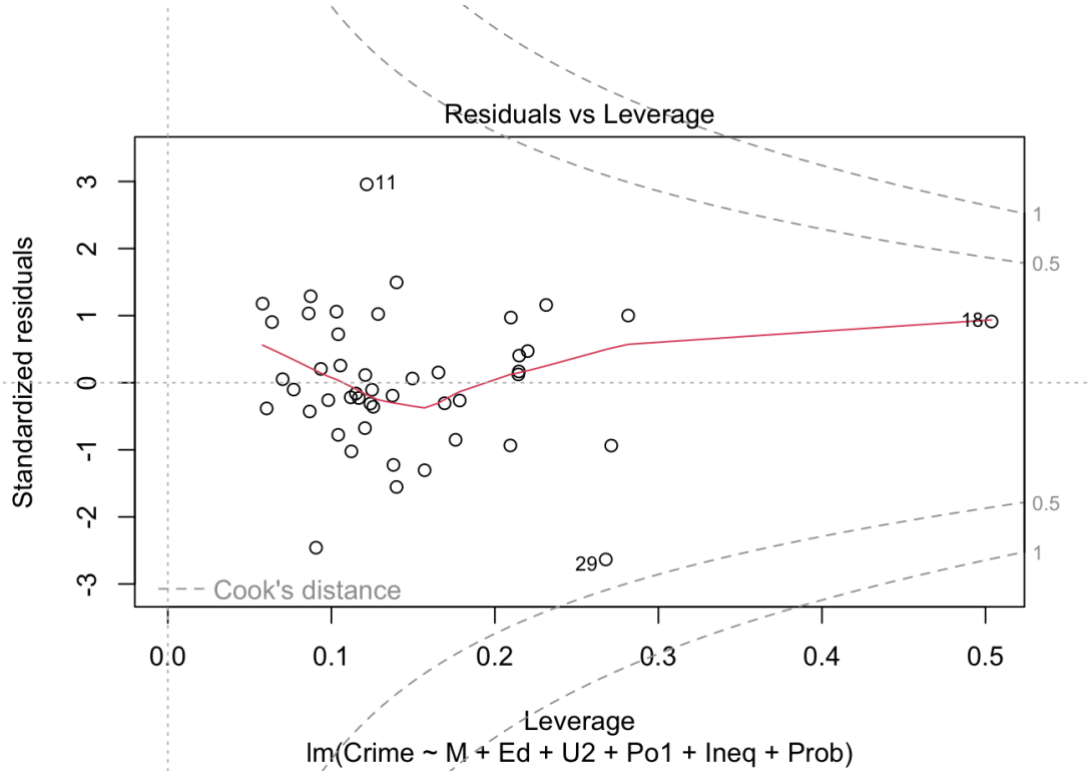
The p value I decided to look at was anything <= 0.05. Looking at the chart, it shows that the coefficients of M, Ed, Po1, U2, Ineq, and Prob all fit that case. Another thing to note here is the residual standard error and the adjusted R squared value. The lower the residual

standard error and the higher the adjusted R squared, the better the model. With this in mind, I wanted to see how the points that are not significant influenced the model. To do this, I removed all coefficients that did not have a p value <= 0.05 and ran a new model. To see how the non-significant coefficients influenced the first model, I will look at the new model's standard error and adjusted R squared value. The new model produced the following graphs, which mean the same things as the ones above, just now with different data.









As shown in the graphs, everything that held true for the first model holds true in this second model. I then ran a summary of this second model to see what values I would get. The numbers it produced were:

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 ***
U2	89.37	40.91	2.185	0.03483 *
Po1	115.02	13.75	8.363	2.56e-10 ***
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

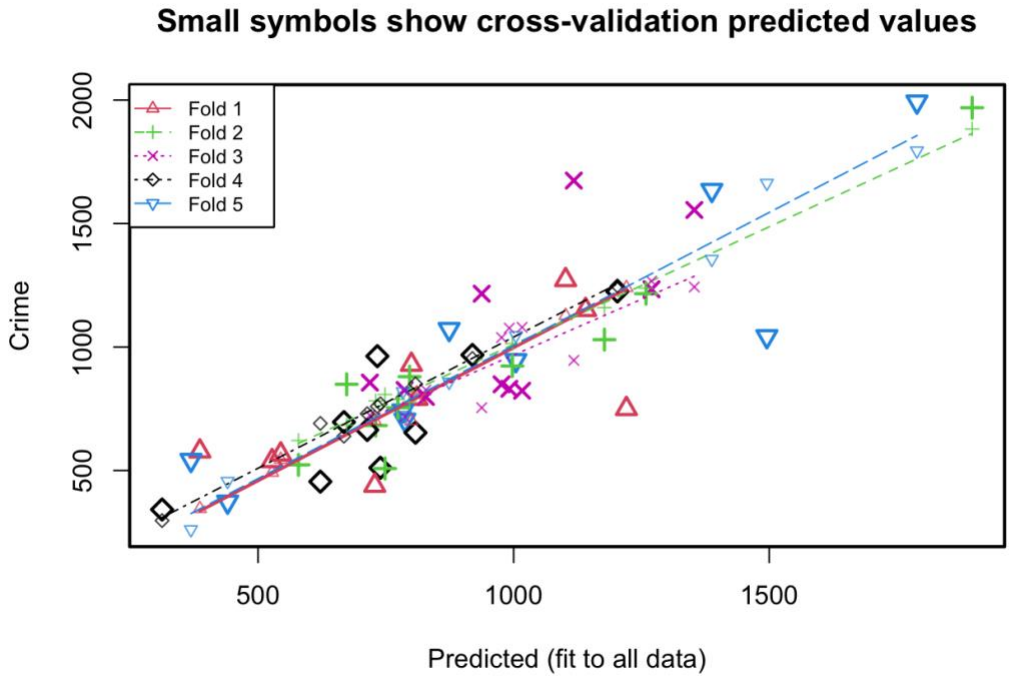
Residual standard error: 200.7 on 40 degrees of freedom

Multiple R-squared: 0.7659, Adjusted R-squared: 0.7307

F-statistic: 21.81 on 6 and 40 DF, p-value: 3.418e-11

Using only the significant coefficients in the first model for the second model, the residual standard error decreased from 209.1 to 200.7. This isn’t a huge difference, but it is an improvement none the less. The adjusted R squared value increased from 0.7078 to 0.7307, which once again, is not a huge jump, but it is an increase. With this, to predict the observed crime rate in a city with the given numbers in the question, I decided to use the second model due to outputting better standard error and R squared values. I created a data frame in R with all the given values of each attribute so that if I wanted to look at the output with the first model, I could also do that. After creating the data frame, I predicted the value of the crime rate of the given data with the second model that I created. The equation came to be: crime rate = -5040.50 + 105.02 * 14.0 + 196.47 * 10.0 + 89.37 * 3.6 +

115.02 * 12.0 + 67.65 * 20.1 – 3801.84 * 0.040 = 1304.245. So, a city with the given values in the question is expected to have a crime rate of 1304.245. To check the quality of this model, I first applied cross validation and then calculated the R squared value. I chose to use 5 folds in the cross-validation test, meaning I divided the data into 5 parts.



The cross-validation gave me the mean square error of each cross-validation test, which was 53586.08. To find the R squared value, I need the sum of squared errors and the sum of differences between attributes and the mean. I found the sum of squared errors by using the cross-validation function I created and found the sum of differences by getting the sum of the real data points – the mean of the real data points and squaring them. The R squared value = 1 – sum of squared errors/sum of differences. The cross-validation R squared value came out to be 0.6339817. Due to overfitting, the R squared value is fairly low, but like said in class, it still is not bad for 0.3 and above is pretty good. Using the “glm” function, I got a lot of the same results, but it was interesting to see that it returned an AIC value when ran. The AIC values would be used to compare models instead of the R squared values of the “lm” function.

R Code and Output:

```
> set.seed(123)
> crimedata <- read.table("uscrime.txt", header = TRUE)
> model <- lm(Crime ~., data = crimedata)
> plot(model)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
> summary(model)

Call:
lm(formula = Crime ~ ., data = crimedata)

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

> model2 <- lm(Crime ~ M + Ed + U2 + Po1 + Ineq + Prob, data = crimedata)
> plot(model2)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
> summary(model2)

Call:
```

```
lm(formula = Crime ~ M + Ed + U2 + Po1 + Ineq + Prob, data = crimedata)

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 ***
U2             89.37     40.91   2.185  0.03483 *
Po1           115.02     13.75   8.363 2.56e-10 ***
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

> 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,
+                    NW = 1.1, U1 = 0.120, U2 = 3.6, Wealth = 3200, Ineq = 20.1, Prob = 0.040, Time = 39.0)
> pred_model <- predict(model2, test)
```

```
> pred_model
     1
1304.245
> pacman::p_load(DAAG)
> cv <- cv.lm(crimedata, model2, m=5)
```

fold 1

Observations in test set: 9

	1	3	17	18	19	22	36	38	40
Predicted	810.825487	386.1368	527.3659	800.0046	1220.6767	728.3110	1101.7167	544.37325	1140.79061
cvpred	785.364736	345.3417	492.2016	700.5751	1240.2916	701.5126	1127.3318	544.69903	1168.21107
Crime	791.000000	578.0000	539.0000	929.0000	750.0000	439.0000	1272.0000	566.00000	1151.00000
CV residual	5.635264	232.6583	46.7984	228.4249	-490.2916	-262.5126	144.6682	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	34	41	44
Predicted	1897.18657	730.26589	673.3766	579.06379	1259.00338	773.68402	997.54981	796.4198	1177.5973
cvpred	1882.73805	781.75573	684.3525	621.37453	1238.31917	788.03429	1013.92532	778.0437	1159.3155

Crime 1969.00000 682.00000 849.0000 523.00000 1216.00000 754.00000 923.00000 880.0000 1030.0000

CV residual 86.26195 -99.75573 164.6475 -98.37453 -22.31917 -34.03429 -90.92532 101.9563 -129.3155

46

Predicted 748.4256

cvpred 807.6968

Crime 508.0000

CV residual -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 37 39 43

Predicted 1269.84196 1353.5532 718.7568 1117.7702 828.34178 937.5703 991.5623 786.6949 1016.5503

cvpred 1266.79544 1243.1763 723.5331 946.1309 826.28548 754.2511 1076.5799 717.0989 1079.7748

Crime 1234.00000 1555.0000 856.0000 1674.0000 798.00000 1216.0000 831.0000 826.0000 823.0000

CV residual -32.79544 311.8237 132.4669 727.8691 -28.28548 461.7489 -245.5799 108.9011 -256.7748

47

Predicted 976.4397

cvpred 1038.3321

Crime 849.0000

CV residual -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	30	35	45
Predicted	733.3799	739.3727	713.56395	1202.9607	919.39117	312.20470	668.01610	808.0296	621.8592
cvpred	759.9655	770.2015	730.05546	1247.8616	953.72478	297.19321	638.87118	850.6961	690.6802
Crime	963.0000	511.0000	664.00000	1225.0000	968.00000	342.00000	696.00000	653.0000	455.0000
CV residual	203.0345	-259.2015	-66.05546	-22.8616	14.27522	44.80679	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	31	33	42
Predicted	1387.8082	787.27124	1004.3984	783.27334	1789.1406	1495.4856	440.4394	873.8469	368.7031
cvpred	1355.7097	723.66781	1046.8197	819.71145	1794.6456	1663.6272	456.5736	857.7052	260.9211
Crime	1635.0000	705.00000	946.0000	742.00000	1993.0000	1043.0000	373.0000	1072.0000	542.0000
CV residual	279.2903	-18.66781	-100.8197	-77.71145	198.3544	-620.6272	-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
```

```
> SSE <- attr(cv, "ms")*nrow(crimedata)
> SST = sum((crimedata$Crime - mean(crimedata$Crime))^2)
> CVR = 1 - SSE/SST
> CVR
[1] 0.6339817
```