ISYE 6501

Homework 5

Artur Cabral, Marta Bras, Pedro Pinto, Katie Price

2019-09-22

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. One recent situation that have happened in my country is the population being affected by the virus H1N1. The first time it happened the was no drug to combat the simptoms of the new virus, and a number of the population affected unfortunately died. The virus was studied, and a drug was developed to fight and cure the simptoms. The next year, there was a new problem, which was the stock of the drug not attending the demand from the population affected. A linear regression model could have been used to estimate the demand of the drug in a specific year, therefore pharmacies would stock accordingly, and properly attend the population. Possible predictors to be used in this model include, but not limited to: number of infecction cases, percentage of death by this infection, amount of drug produced, number of cases cured by the drug, average age of population infected, etc.

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:

$M = 14.0$ $S0 = 0$ $Ed = 10.0$ $Po1 = 12.0$ $Po2 = 15.5$ $LF = 0.640$ $MF = 94.0$ $Pop = 150$ $NW = 1.1$ $U1 = 0.120$ $U2 = 3.6$ $Wealth = 3200$ $Ineq = 20.1$ $Prob = 0.04$ $Time = 39.0$

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.

Before initating a linear regression model, it's important to look at the distribution of the variables and the relationship between them.

1. Importing and initial analysis of the data

```
#reading the data
crime_data <- read.table("uscrime.txt", header = TRUE)

# Overall statistics
summary_table <- crime_data %>%
  summarize(number_states = nrow(crime_data),
            total_crime_rate = sum(Crime),
            average_crime_rate = mean(Crime),
            min_crime_rate = min(Crime),
```

Table 1: Crime Rate - overall statistics

number_states	total_crime_rate	average_crime_rate	min_crime_rate	max_crime_rate
47	42539	905.0851	342	1993

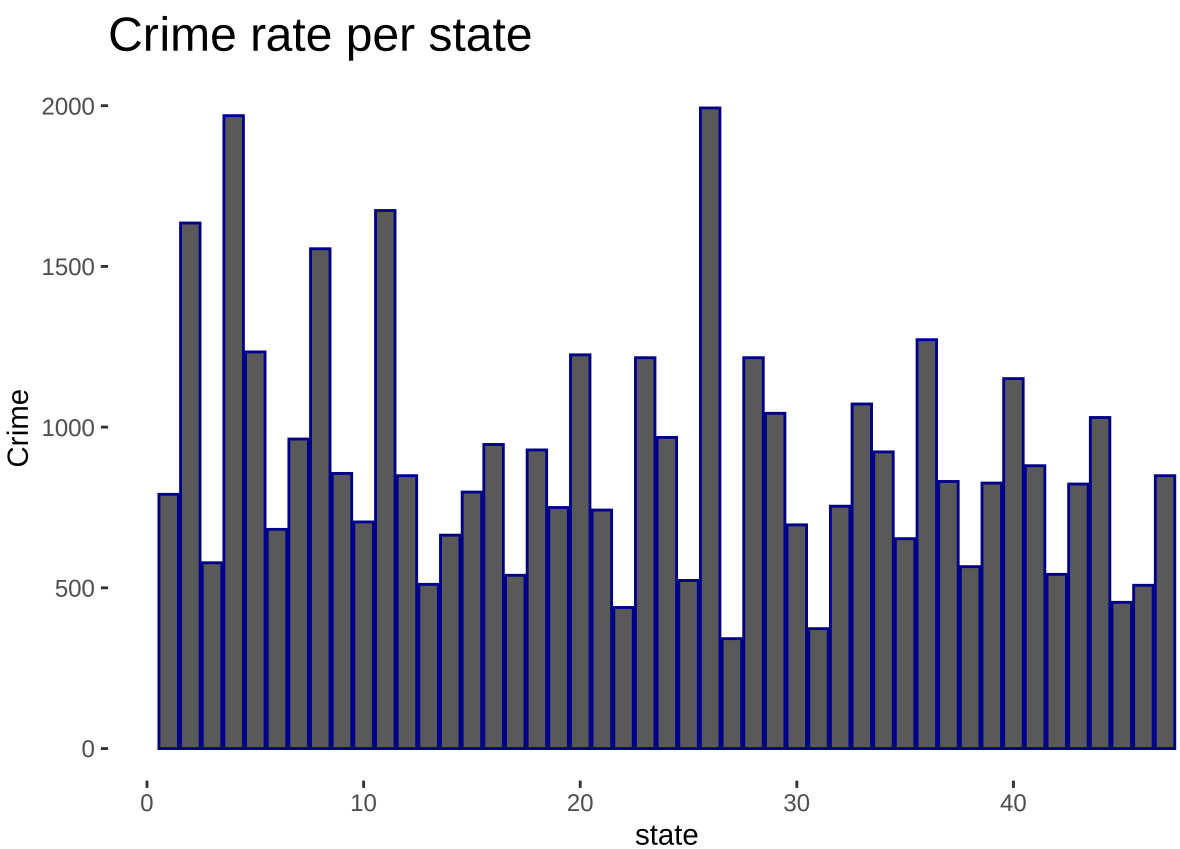
Table 2: Crime Rate per state

So	number_states	total_crime_rate	average_crime_rate	min_crime_rate	max_crime_rate
0	47	28830	930.0000	342	1993
1	47	13709	856.8125	439	1555

```
            max_crime_rate = max(Crime))

kable(summary_table, caption = "Crime Rate - overall statistics") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "bordered"))
```

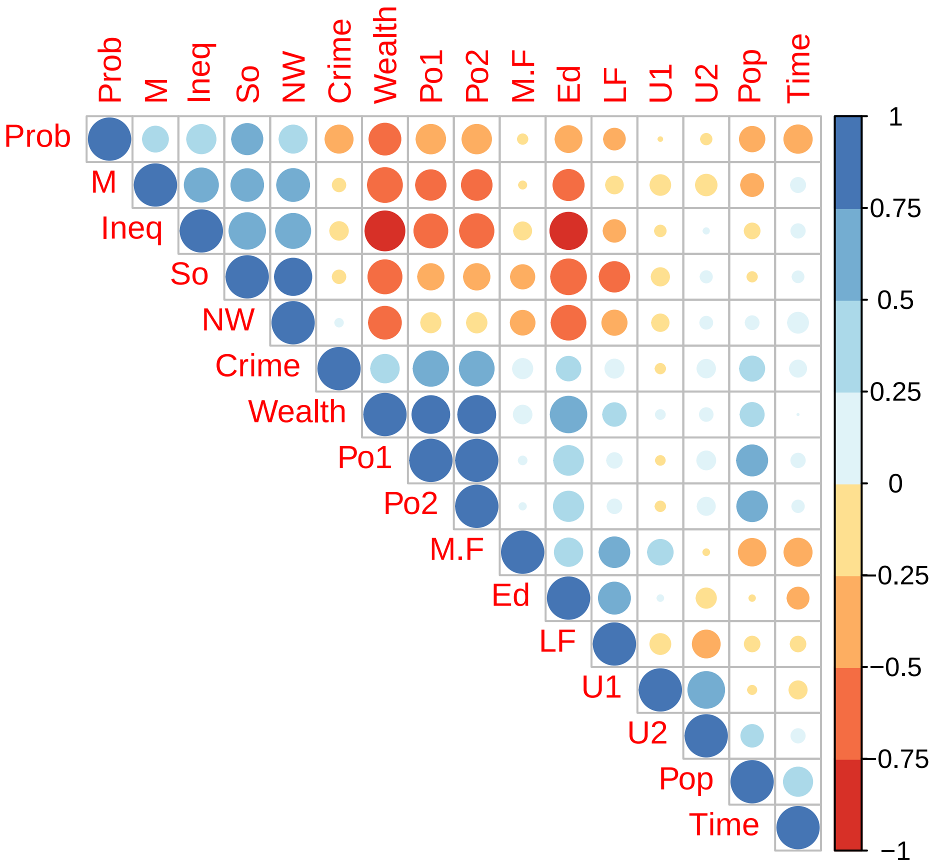
```
# Statistics per state
summary_table_state <- crime_data %>% group_by(So) %>%
  summarize(number_states = nrow(crime_data),
            total_crime_rate = sum(Crime),
            average_crime_rate = mean(Crime),
            min_crime_rate = min(Crime),
            max_crime_rate = max(Crime))

kable(summary_table_state, caption = "Crime Rate per state") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "bordered"))
```

```
# Crime rate per state
## adding an index variable
state <- seq(1, length(crime_data$So))
crime_data_2 <- cbind(state, crime_data)
state <- factor(crime_data_2$state)

ggplot(crime_data_2, aes(x=state, y = Crime)) +
  geom_col(color='darkblue') +
  ggtitle("Crime rate per state") +
  theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()) +
  theme(plot.title = element_text(size=18))
```



From the table above, we can see that the minimum crime rate between the 47 states is 342 crimes per 100,000 people. The maximum crime rate was 1993 per 100,000 people. The minimum and maximum crime rate are different from northern to southern states. We can also see some states have much higher crime rates than others.

```
#Correlations between variables

M <-cor(crime_data)
corrplot(M, type="upper", order="hclust",
         col=brewer.pal(n=8, name="RdYlBu"))
```

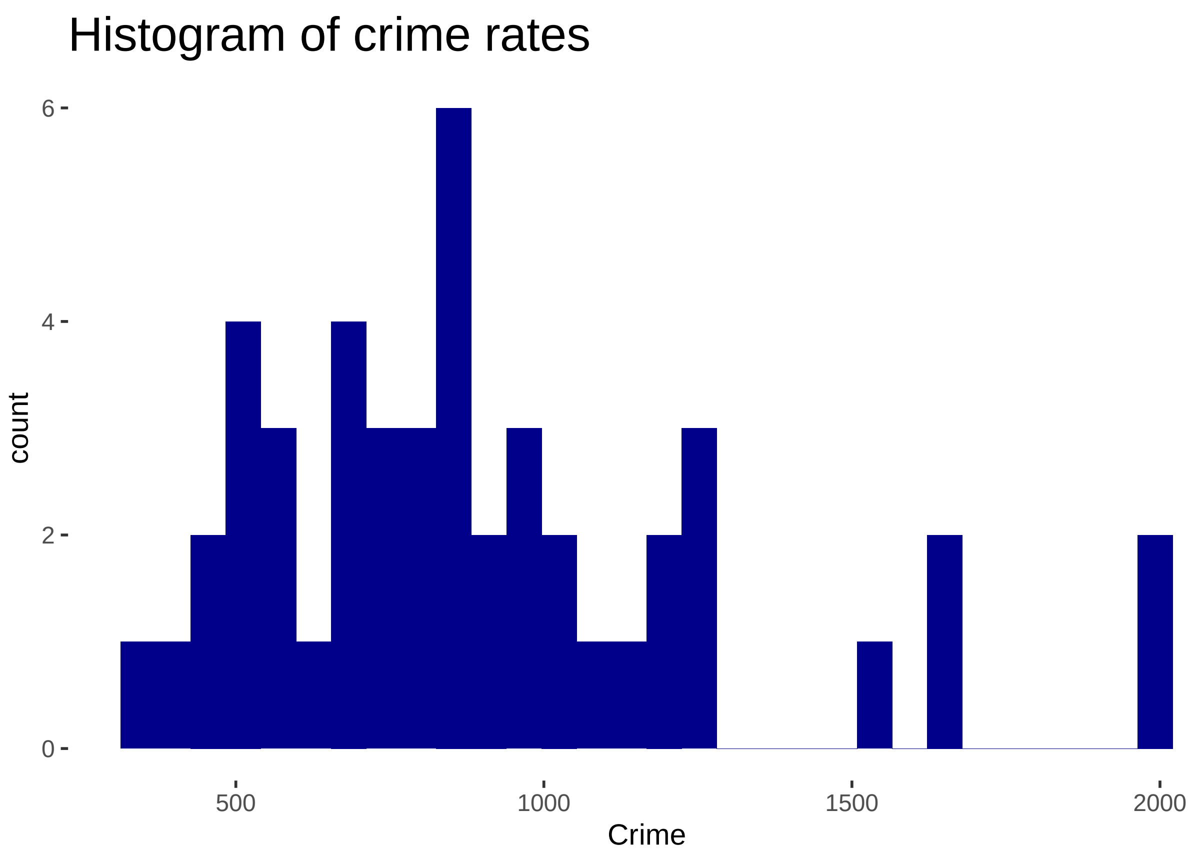


As we can see from the correlation matrix above, a lot of variables have correlations higher than 50% and some variables have correlations higher than 75%. Considering the small number of data points, the risk of over-fitting is especially high. For instances, variables Inequality and Wealth and variables Inequality and Education have correlations above 75%.

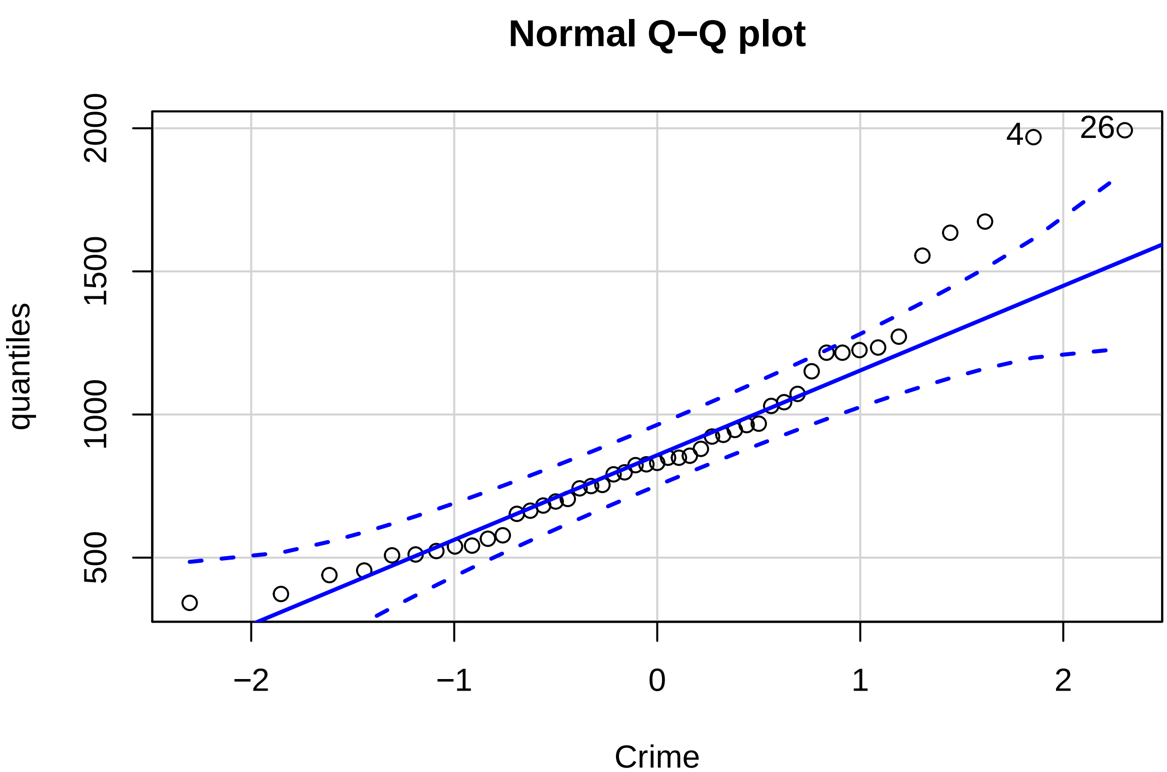
After doing a preliminary analysis we know things we should be aware when building the regression model: 1. Y variable is not normally distributed and there are outliers in the data, 2. There is strong correlation between possible explanatory variables.

```
#distribution of the y variable
# Histogram of crime
ggplot(crime_data, aes(x=Crime)) +
  geom_histogram(fill = "darkblue") +
  ggtitle("Histogram of crime rates") +
  theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()) +
  theme(plot.title = element_text(size=18))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

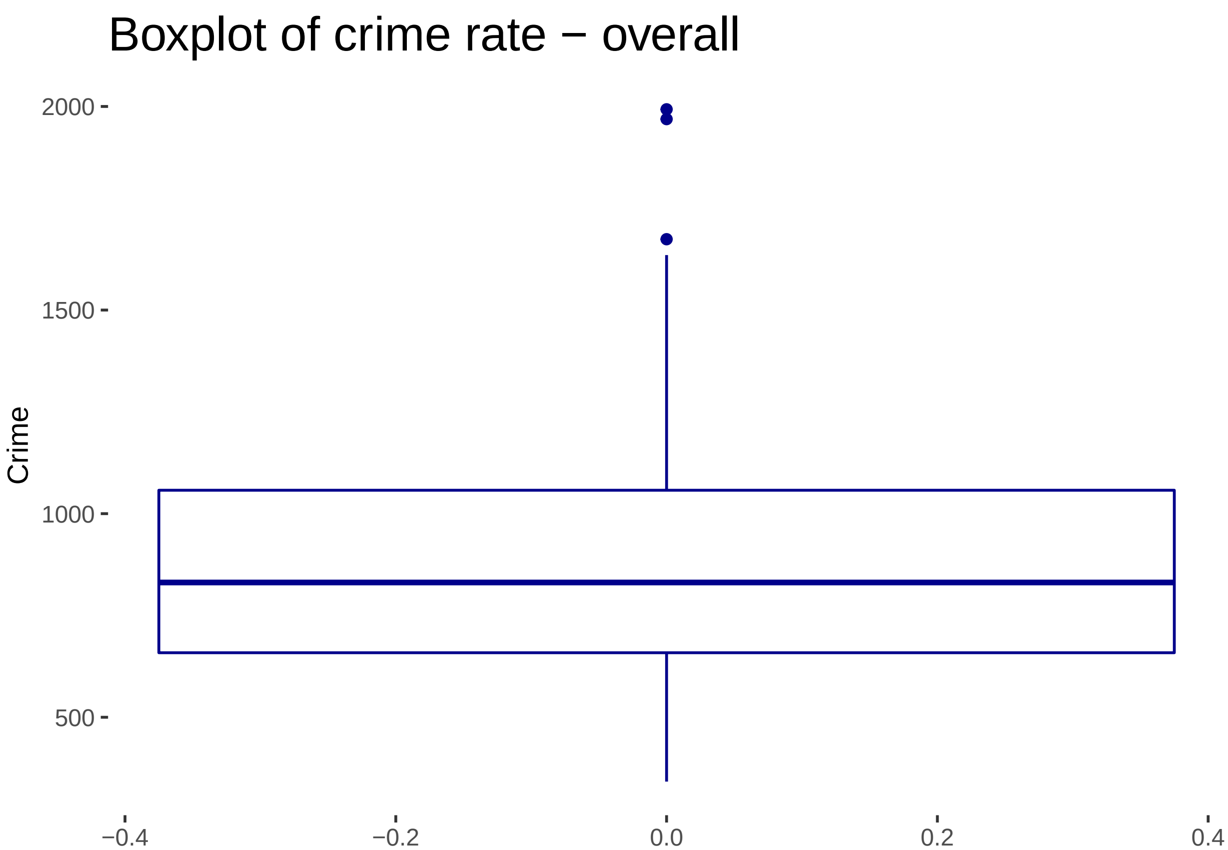



```
##QQplot graph of crime rate overall
qqPlot(crime_data$Crime, main = "Normal Q-Q plot", xlab = "Crime", ylab = "quantiles")
```



```
## [1] 26  4
```

```
# Boxplots of crime rate
## changing to factor variable to use in the histogram
crime_data$So <- factor(crime_data$So)

ggplot(crime_data, aes(y=Crime)) +
  geom_boxplot(color = "darkblue") +
  ggtitle("Boxplot of crime rate - overall") +
  theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()) +
  theme(plot.title = element_text(size=18))
```



The crime variable has a distribution with heavy tales on the right side, which is visible in both the qqplot and the histogram.

2. Linear regression model - all variables

We will start by understnading how the model performs using all possible explanatory variables.

```
#linear model with all explanatory variables
model <- lm(Crime ~ . , data = crime_data)
#summary model
summary(model)
```

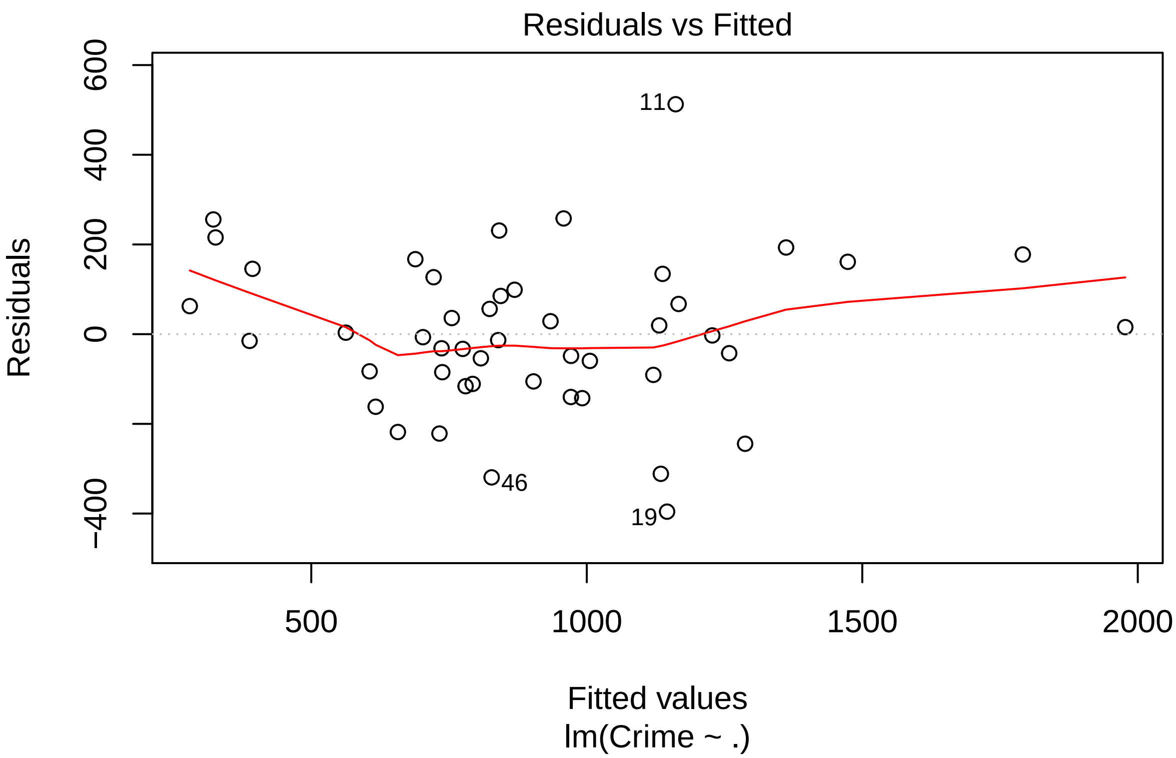
```
##
```

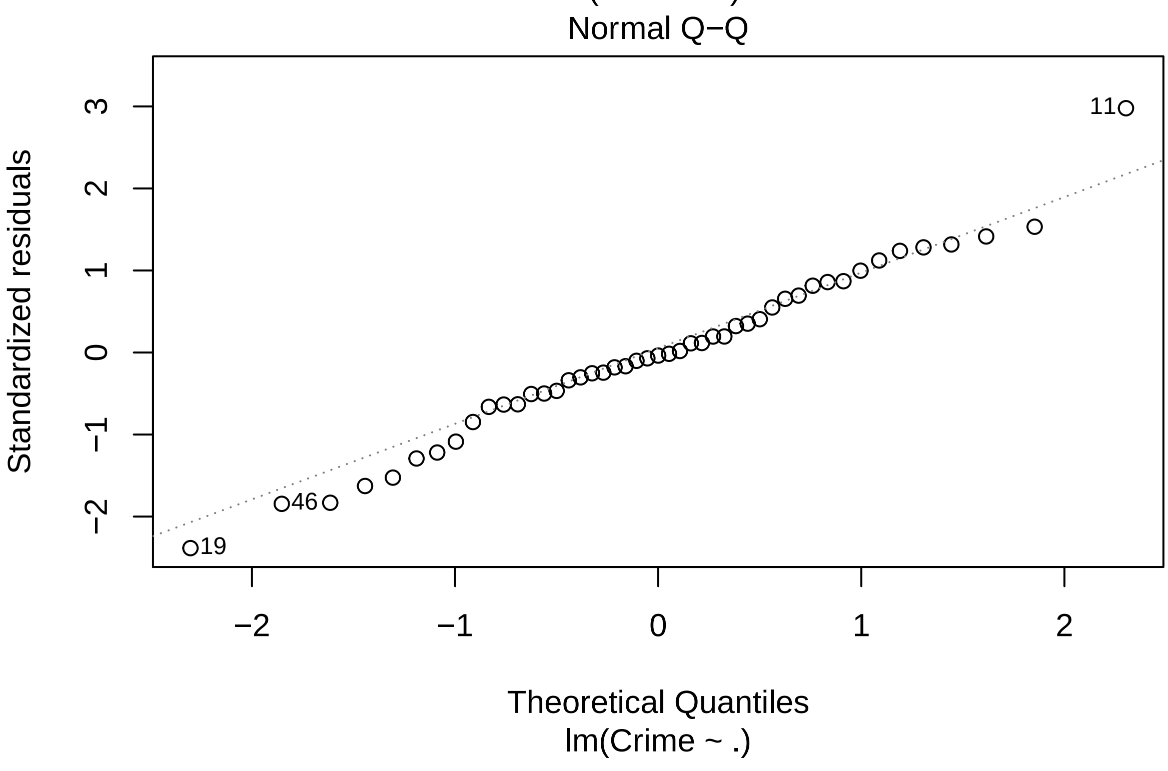
```
## Call:
## lm(formula = Crime ~ ., data = crime_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -395.74  -98.09   -6.69  112.99  512.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So1         -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
```

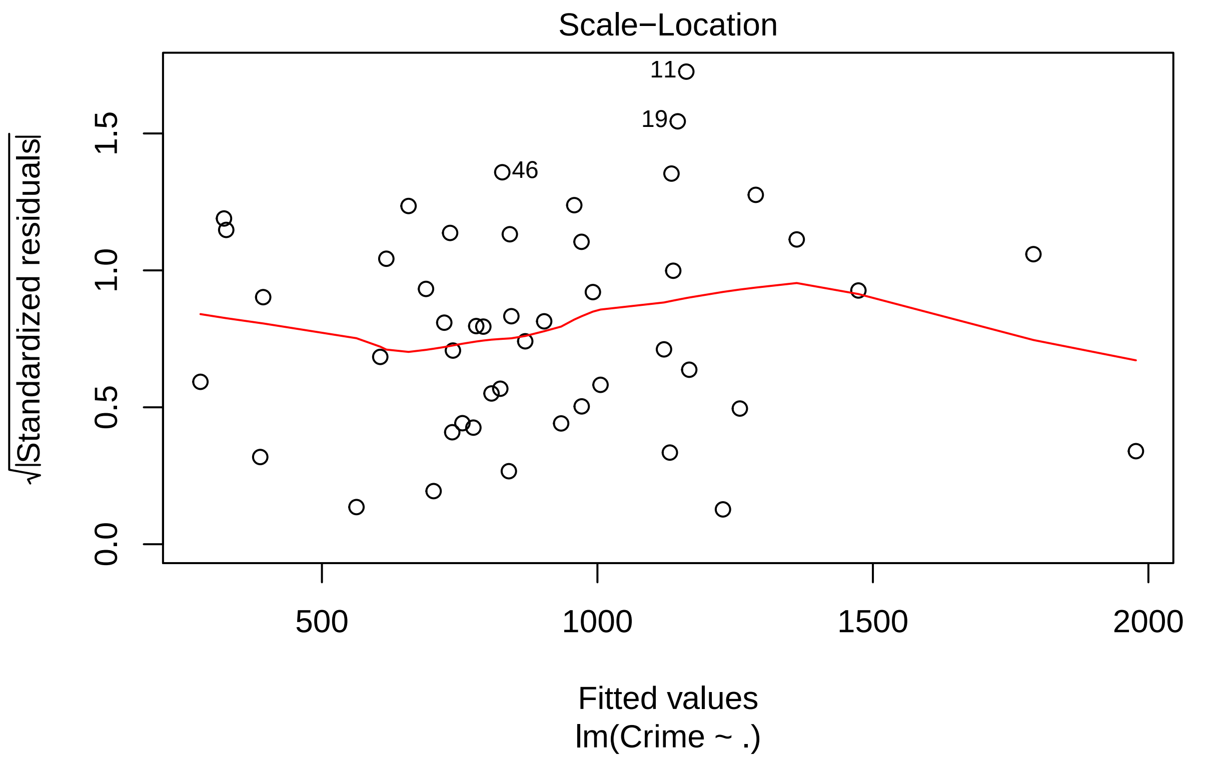
The model using all explanatory variables has a R2 of 80.3%. The adjusted R2 is significantly lower (70.8%) which might be a sign of overfitting. Additionally, only 5 out of the 15 variables are significant at a 5% significance level.

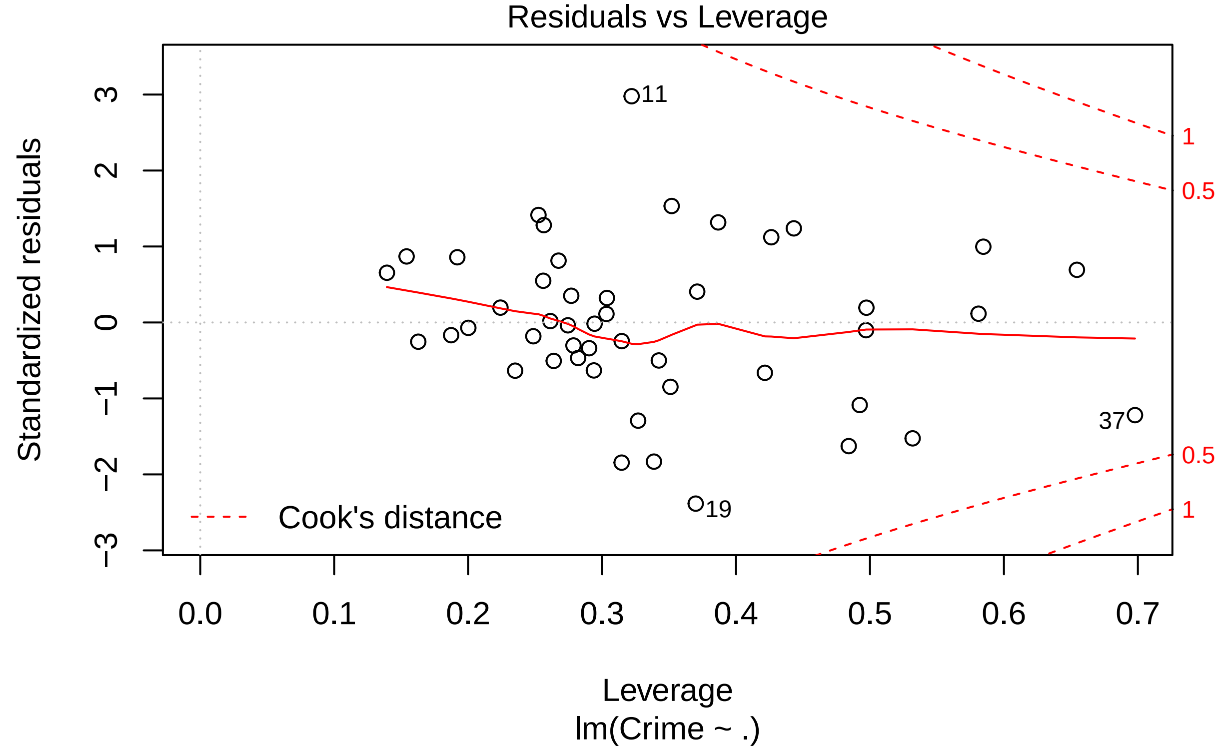
To annalyze the goodness-of-the fit of the model it is important to plot the results of the model and understand if the assumptions for linear regression are being met or not.

```r
#plots of the results
plot(model)
```


Residuals vs Fitted
Residuals
600
400
200
0
−400
500
1000
1500
2000
11
46
19
Fitted values
lm(Crime ~ .)


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
11
46
19
−2
−1
0
1
2
Theoretical Quantiles
lm(Crime ~ .)





Looking at the residual vs fitted plot, we can see assumption of linearity and constant variance does not seem to hold as most of the data points are concentrated in the middle of. The QQplot reveals a distribution close to normal, with some outliers on both sides of the distribution.

```
#predicting on the new data
newdata = data.frame(M=14.0, So = 0, Ed = 10.0, Po1 = 12.0, Po2 = 15.5, LF = 0.640, M.F = 94.0, Pop = 15
```

```r
newdata$So = as.factor(newdata$So)
predict(model, newdata)
```

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

When we predict the crime rate in the linear model using all explanatory variables, the predicted crime rate is 155, half than the minimum crime rate in the database. Not only the model includes variables that are not significant but it also results in questionable predictions.

The model predicts that given the inputs for the explanatory variables, the overall crime rate will be 155 per 100,000 inhabitants.

3. Linear regression model - forward selection

To understand which variables to use in the model, we can use the regsubsets package in the leaps package. It returns multiple models with different sizes up to the maximum number of variables defined. It can also be used in combination with the caret package in R, allowing to perform cross-validation with the train() function. We will use the forwared selection, which starts with one variable and adds additionally variables based on which variable is the best for a specific criteria. It stops adding variables if they no longer make the model better based on that criteria.

3.1. Leap function

```r
# Set seed for reproducibility
set.seed(123)
# Set up repeated k-fold cross-validation
train.control <- trainControl(method = "cv", number = 10)
# Train the model
step.model_1 <- train(Crime ~., data = crime_data,
                    method = "leapForward",
                    tuneGrid = data.frame(nvmax = 1:15),
                    trControl = train.control
                    )

#results from the mddel for different number of variables
step.model_1$results
```

```
##    nvmax     RMSE  Rsquared      MAE    RMSESD RsquaredSD     MAESD
## 1      1 288.6803 0.5228157 240.5501 143.17312  0.2977296 124.63342
## 2      2 272.5431 0.5669029 229.3939 157.32323  0.3295535 130.51675
## 3      3 271.4696 0.5793512 225.0962 129.96420  0.2651761 108.07110
## 4      4 270.7751 0.5858610 224.5047 120.61199  0.2582675 109.35970
## 5      5 274.4471 0.5517368 226.2092 103.22087  0.2264569  86.52449
## 6      6 237.8908 0.6352862 194.5121 113.91813  0.2704280 100.54200
## 7      7 261.6499 0.5840557 209.2954 101.97586  0.2365082  96.07206
## 8      8 262.4524 0.6122299 205.8438  99.76725  0.2869717  94.04247
## 9      9 267.8824 0.6006502 207.4250 100.92142  0.2610353  92.98494
## 10    10 274.4854 0.5577193 217.9605  98.20975  0.2801872  87.38319
## 11    11 271.7854 0.5263632 217.3852  88.74278  0.2931330  75.49982
```

```
## 12    12 279.8652 0.5192279 221.3027  83.60589  0.2832111  72.29232
## 13    13 284.6493 0.5193404 227.8602  84.39618  0.2729154  71.64011
## 14    14 282.7471 0.5187387 229.0520  86.00962  0.2708502  74.06545
## 15    15 281.2540 0.5154576 228.6937  88.83478  0.2731962  75.02496
```

```r
#summary
summary(step.model_1$finalModel)
```

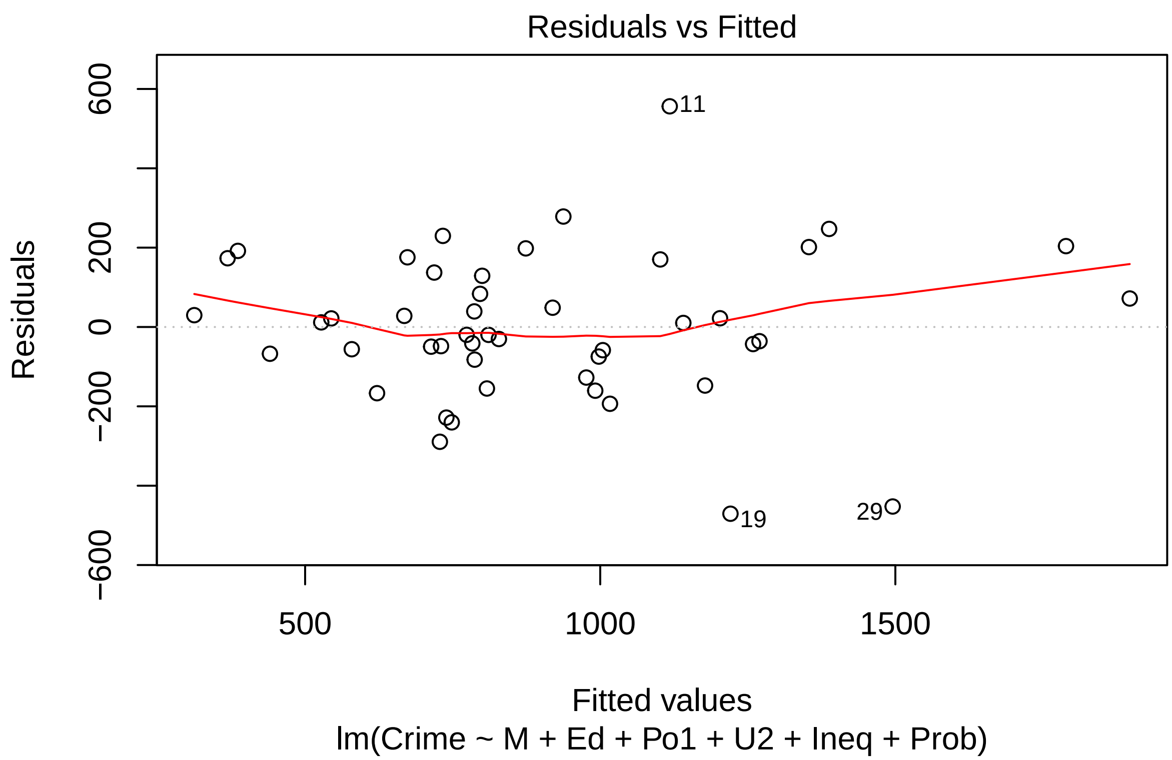
```
## Subset selection object
## 15 Variables  (and intercept)
##        Forced in Forced out
## M          FALSE      FALSE
## So1        FALSE      FALSE
## Ed         FALSE      FALSE
## Po1        FALSE      FALSE
## Po2        FALSE      FALSE
## LF         FALSE      FALSE
## M.F        FALSE      FALSE
## Pop        FALSE      FALSE
## NW         FALSE      FALSE
## U1         FALSE      FALSE
## U2         FALSE      FALSE
## Wealth     FALSE      FALSE
## Ineq       FALSE      FALSE
## Prob       FALSE      FALSE
## Time       FALSE      FALSE
## 1 subsets of each size up to 6
## Selection Algorithm: forward
##          M   So1 Ed  Po1 Po2 LF  M.F Pop NW  U1  U2  Wealth Ineq Prob Time
## 1  ( 1 ) " " " " " " "*" " " " " " " " " " " " " " " " "    " "  " "  " "
## 2  ( 1 ) " " " " " " "*" " " " " " " " " " " " " " " " "    "*"  " "  " "
## 3  ( 1 ) " " " " "*" "*" " " " " " " " " " " " " " " " "    "*"  " "  " "
## 4  ( 1 ) "*" " " "*" "*" " " " " " " " " " " " " " " " "    "*"  " "  " "
## 5  ( 1 ) "*" " " "*" "*" " " " " " " " " " " " " " " " "    "*"  "*"  " "
## 6  ( 1 ) "*" " " "*" "*" " " " " " " " " " " " " "*" " "    "*"  "*"  " "
```

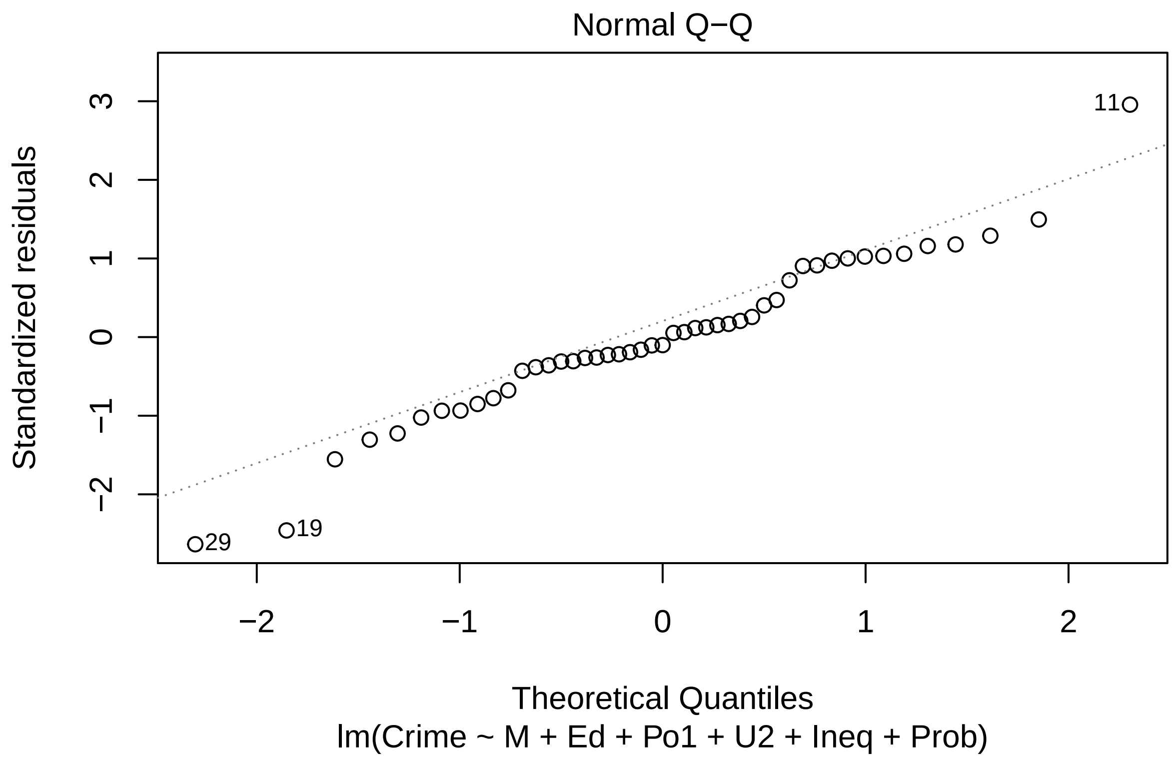
```r
#model with selected variables
model_2 <- lm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime_data)

#summary
summary(model_2)
```

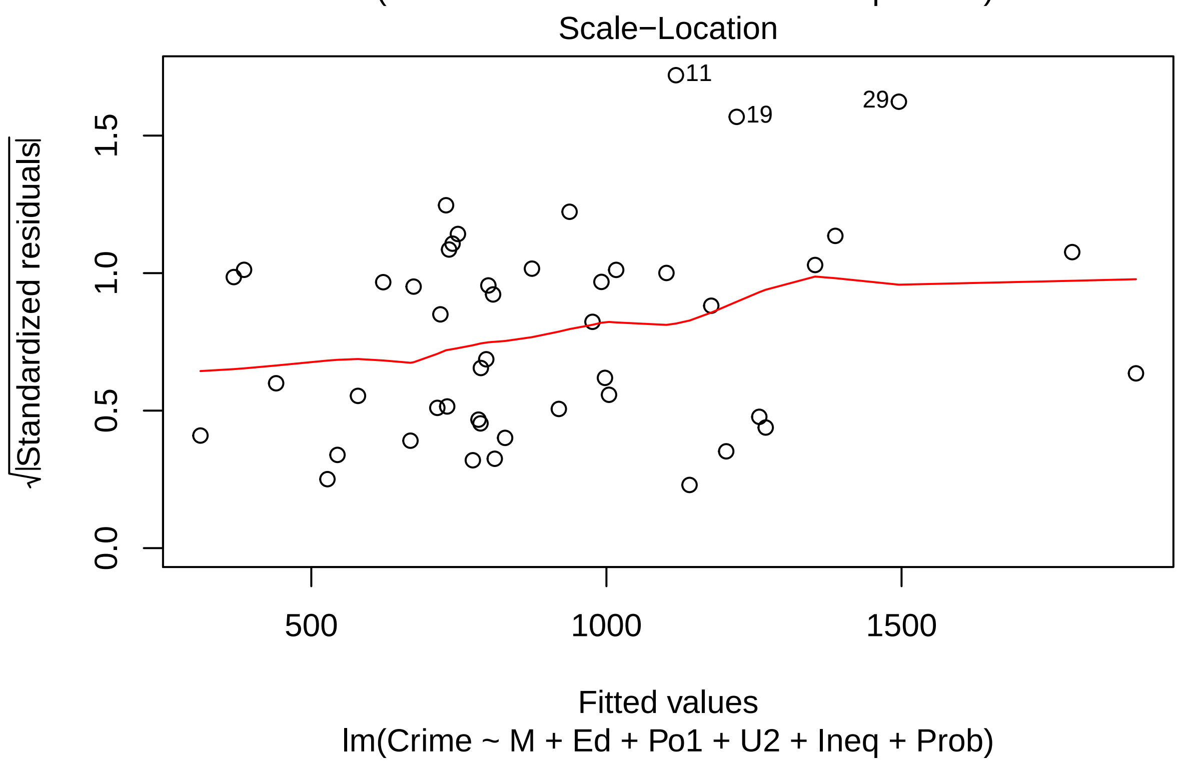
```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -470.68  -78.41  -19.68  133.12  556.23
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84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

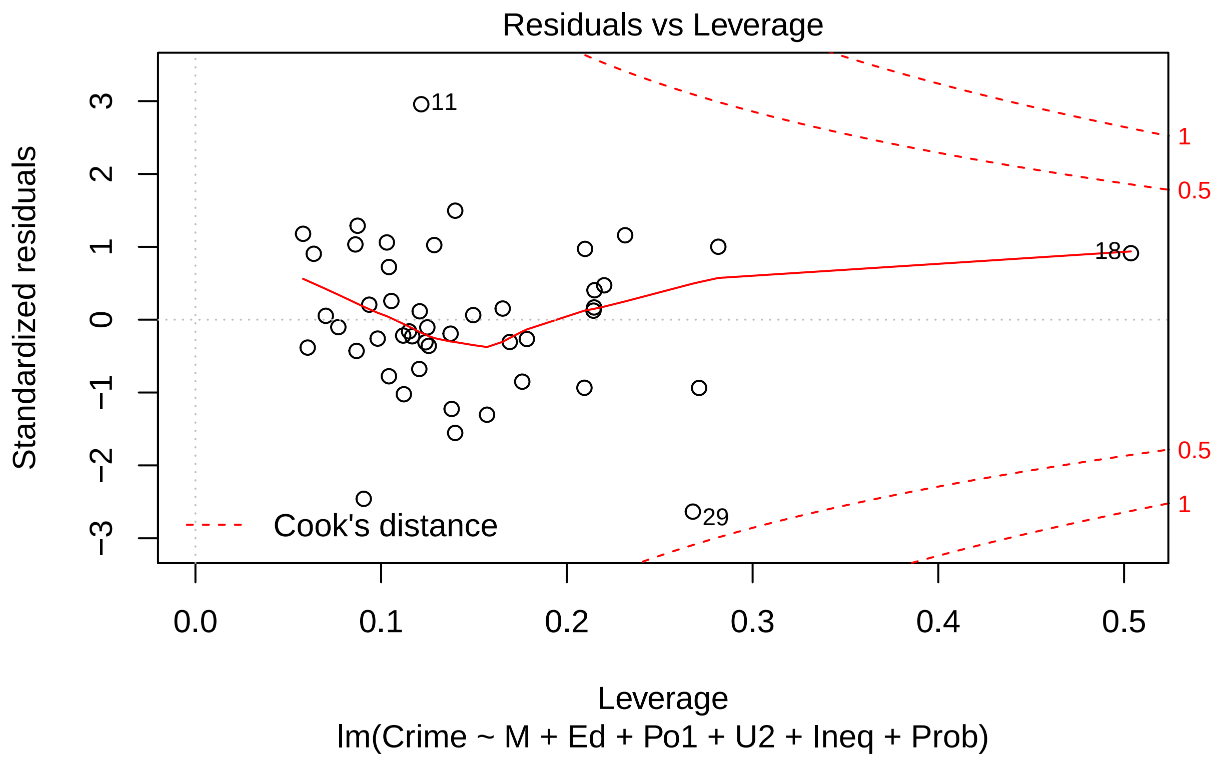
```
## M             105.02      33.30   3.154  0.00305 **
## Ed            196.47      44.75   4.390 8.07e-05 ***
## Po1           115.02      13.75   8.363 2.56e-10 ***
## U2             89.37      40.91   2.185  0.03483 *
## Ineq           67.65      13.94   4.855 1.88e-05 ***
## Prob        -3801.84    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 of the results
plot(model_2)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob)
−2
−1
0
1
2
3
11
19
29


Scale−Location
√|Standardized residuals|
Fitted values
lm(Crime ~ M + Ed + Po1 + U2 + Ineq + Prob)
0.0
0.5
1.0
1.5
500
1000
1500
11
19
29



Using the leap function with forward selection, we can see the best model has 6 variables as this is the model that minimizes the MAE on the testing datasets. The R2 from this model is slightly lower than the model with all variables (76% vs 80%) but the R2 adjusted is higher, which might indicates the model is less subject to overfitting.

Also, all the variables in this model are statistically significant and we removed the high correlation problem between inequality and wealth.

Looking at the residuals vs fitted plot, we see the residuals are better dispersed, not as concentrated in the middle as in the full model.

```
#predicting on the new data
predict(model_2, newdata)
```

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

The prediction for the crime in the new model is more reliable than in the full model, even though it is higher than the third quantile value for the variable. This might be a result of the outliers in the model, wich have a high impact, considering the small number of observations.

The model predicts that given the inputs for the explanatory variables, the overall crime rate will be 1302 per 100,000 inhabitants.

3.2. Step function

```
# Set seed for reproducibility
set.seed(123)
# Set up repeated k-fold cross-validation
```

```r
train.control <- trainControl(method = "cv", number = 10)
# Train the model
step.model <- train(Crime ~., data = crime_data,
                    method = 'lmStepAIC',
                    trControl = train.control
                    )
```

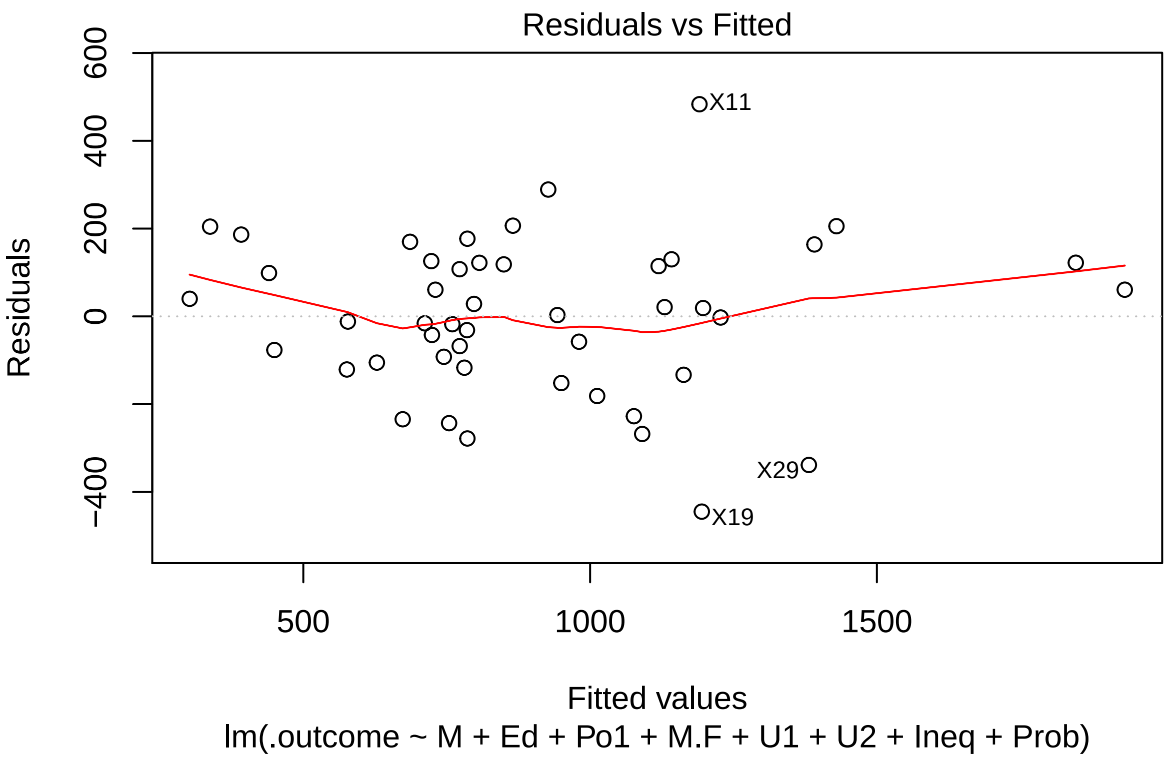
```r
#results from the mddel for different number of variables
step.model$results
```

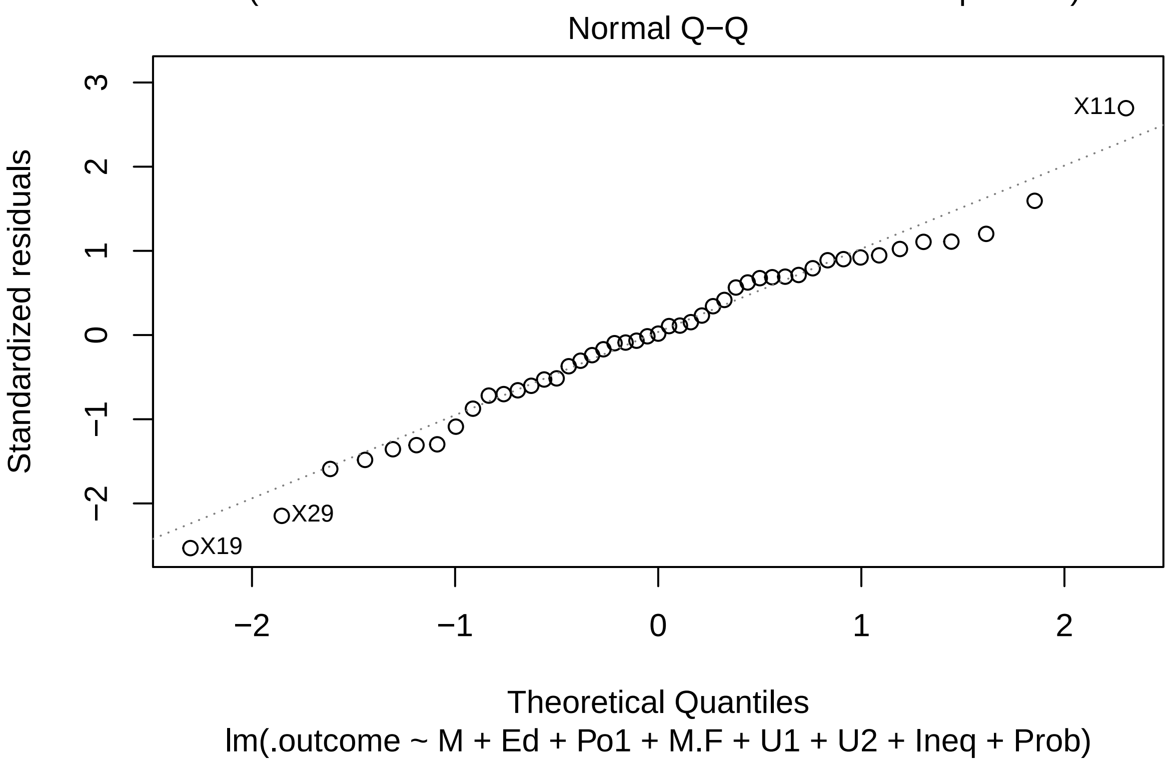
```
##   parameter     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      none 259.8035 0.6016036 207.6698 112.2749  0.2750211 101.9361
```

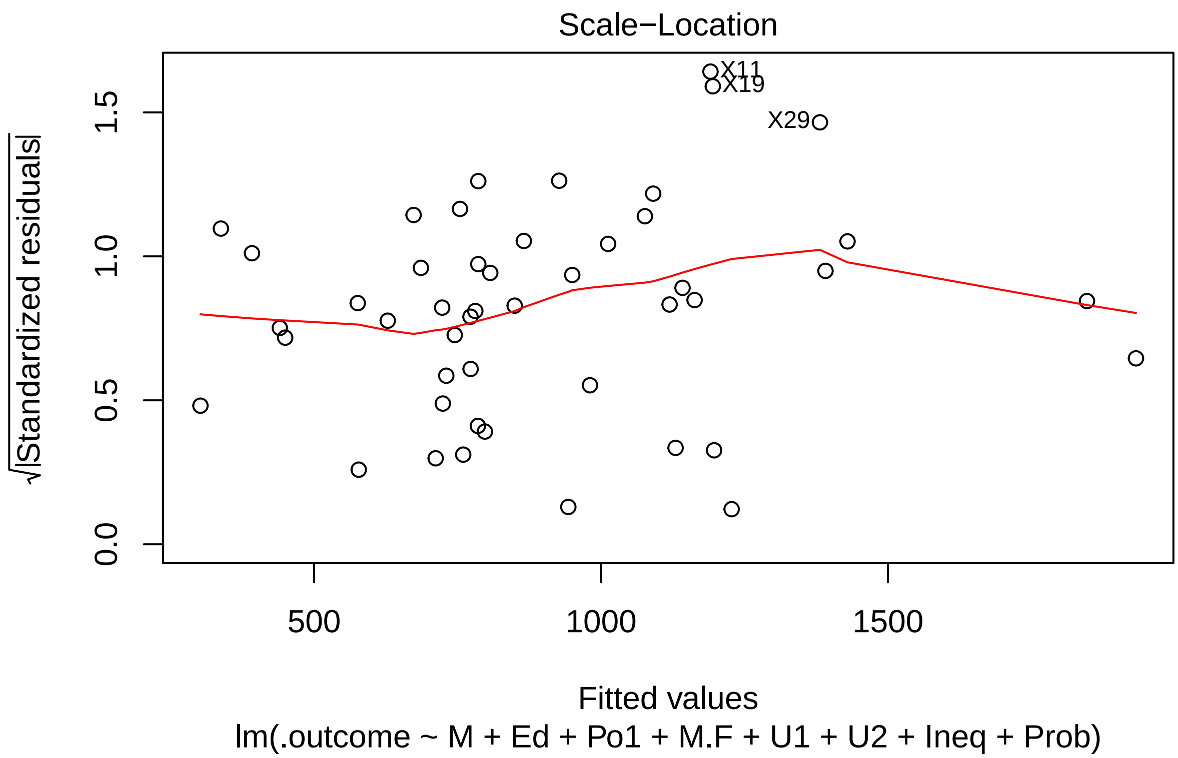
```r
#summary
summary(step.model$finalModel)
```

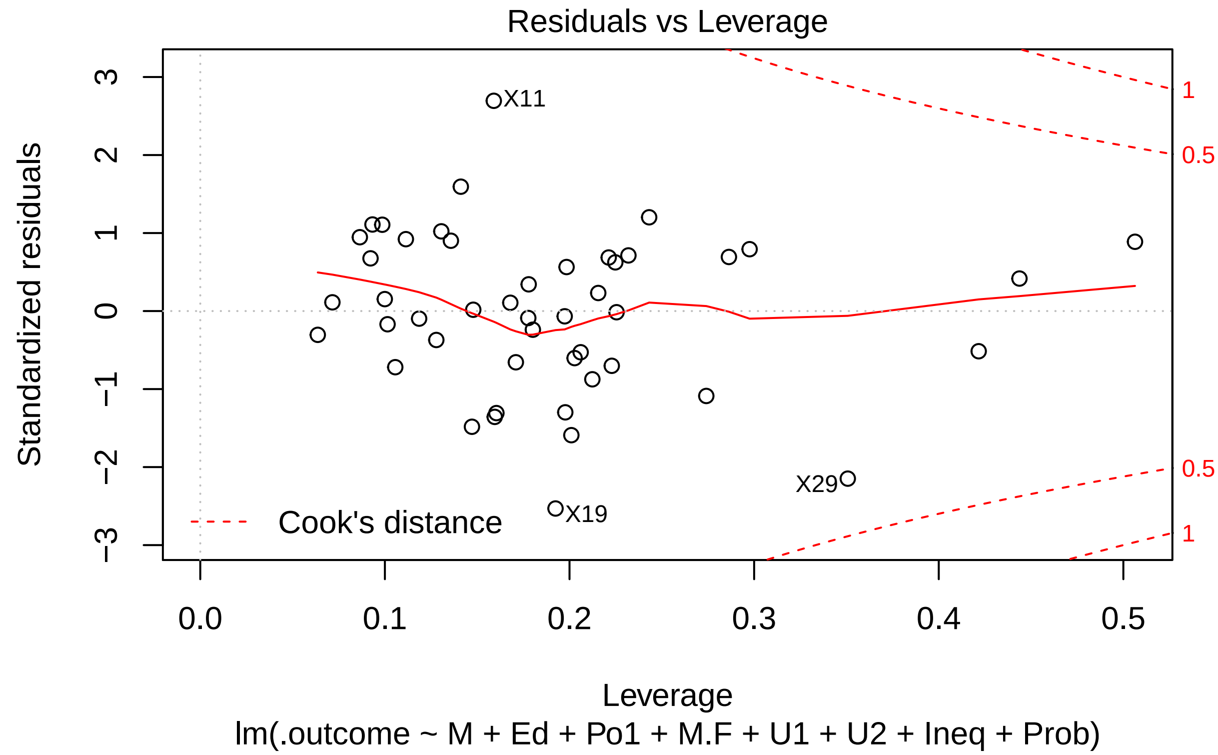
```
##
## Call:
## lm(formula = .outcome ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq +
##     Prob, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -444.70 -111.07    3.03  122.15  483.30
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6426.10    1194.61  -5.379 4.04e-06 ***
## M              93.32      33.50   2.786  0.00828 **
## Ed            180.12      52.75   3.414  0.00153 **
## Po1           102.65      15.52   6.613 8.26e-08 ***
## M.F            22.34      13.60   1.642  0.10874
## U1          -6086.63    3339.27  -1.823  0.07622 .
## U2            187.35      72.48   2.585  0.01371 *
## Ineq           61.33      13.96   4.394 8.63e-05 ***
## Prob        -3796.03    1490.65  -2.547  0.01505 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 195.5 on 38 degrees of freedom
## Multiple R-squared:  0.7888, Adjusted R-squared:  0.7444
## F-statistic: 17.74 on 8 and 38 DF,  p-value: 1.159e-10
```

```r
#plot of the results
plot(step.model$finalModel)
```


Residuals vs Fitted
X11
X29
X19
Residuals
600
400
200
0
−400
500
1000
1500
Fitted values
lm(.outcome ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob)


Normal Q−Q
X11
X29
X19
Standardized residuals
3
2
1
0
−1
−2
0
Theoretical Quantiles
lm(.outcome ~ M + Ed + Po1 + M.F + U1 + U2 + Ineq + Prob)





Using the step function to perform a stepwise function selection by AIC, we can see the best model has 8 variables as this model minimizes the AIC.The selection is done by testing different numbers of paramenters and the maximum likelyhood value that results in the lowest set of paramenters with best statistical significance for the model. The R2 from this model is slightly lower than the model with all variables (78% vs 80%) but the R2 adjusted is higher (74% vs 70%), which might indicates the model is less subject to overfitting.

Looking at the residuals vs fitted plot, we see the residuals are better dispersed, not as concentrated in the middle as in the full model.

```
#predicting on the new data
predict(step.model, newdata)
```

```
##        1
## 1038.413
```

3.3. Conclusion

Using the stepwise function selecting the lowest AIC, produces lower R2 than the full model but higher adjusted R2. Additionally, it only includes two variables that are not significant, comparing with 10 non significant variables in the first model.

The leap function produces slightly lower R but the RMSE and MAE values in the training dataset are lower and it only includes significant variables.

Using the regression function from the leap function, the final model would be $-5040.50 + 105.02M + 196.47Ed + 115.02Po1 + 89.375U2 + 67.65Ineq - 3801.84Prob$

The prediction woould then be 1304.245 crimes per 100,000 inhabitants.