Crime Rate Prediction using Linear Regression

Alena Fedash

2022-09-24

Contents

Summary of Results

To start with, it is important to notice that we have a **high number of predictors - number of data points** ration: 15 parameters (without the Crime response variable) and only 47 observations. Based on the **one in ten rule**, it is good practice to have a 10:1 data points-predictors ratio. In the data set, we have a ratio of approximately 3:1 (47/15). This suggests that **the risk of overfitting in the regression model would be quite high**. Below is the summary of the steps I took to reduce the chances of overfitting and to build a model with good predictive power, and the results I got.

- To start with, I explored the data set and found that most of the variables are moderately skewed and are not normally distributed. However, since the response variable's distribution is close to normal, it is still possible to do regression analysis, since such a model does not require normally-distributed predictors.
- Then, before building the first regression model, I checked the variables for **pairwise correlation**, since there seemed to be many parameters describing similar attributes, such as police expenditures in two consequent years (Po1-Po2), wealth and inequality, male unemployment and for two age groups (U1-U2), etc. I built a **correlation plot**, which showed high correlation between several pairs of predictors (Po1-Po2, Wealth-Ineq, U1-U2, So-Ed, So-NW, Po(1,2)-Wealth(Ineq), etc.). The parameter Po1 seemed to be 'pulling down' the other variable, as it showed high correlation with several factors at once. This means that **I would probably need to test the model for multicollinearity and and remove several parameters that show high correlation**. Without this measure, the model is very likely to have **poor predictive power**.
- Next, to prove the point from the above, I built a regression model using all the predictors. For this assignment, I decided not to scale the data because we were given a test data point, which is not scaled. Attempting to scale it along with other data could prompt inaccurate prediction, since we have so few data points. As expected, the **model with all 15 predictors** made an **inaccurate prediction of 155** for the Crime rate, which is lower than the minimal value of the variable (300), and does not seem as a true value for a data point with such parameters. Moreover, **only 6 out of 15 predictors were found significant for the regression line**, confirming that some variables need to be removed in order to produce a model with normal predictive power. However, this step was helpful to take a look at the data in terms of the general appropriateness of linear regression for the data. **The 4 plots produced by the regression model** confirmed that there are no problematic cases in the data: residuals do not show non-linear patterns, the predictors-response relationship is linear, residuals have normal distribution and show homoscedasticity, and **outliers have no influence on the regression line**, meaning that removing them would not make any difference for the model. **Model 1: all predictors, 6 are significant, Residual standard error: 209.1 on 31 degrees of freedom, Multiple R-squared: 0.8031, Adjusted R-squared: 0.7078. Prediction: 155**

- Before leaving only significant factors in the model I decided to dive deeper and explore **multi-collinearity** of the variables. I used Farrar-Glauber test package to investigate multicollinearity in three steps. First, Overall Multicollinearity Diagnostics Measures on the model with all predictors confirmed with 5 out of 6 metrics (such as Farrar Chi-square, which had a high value of 497) that the set of vectors used for the model are linearly dependent, hence, multicollinearity is present. I applied Individual Multicollinearity Diagnostic Measures of the package to locate multicollinearity and check which variables cause it. The highest Variance Inflation Factor was, as expected, for Po2, followed by Po1 and Wealth (all had VIF over 10, proving multicollinearity). This meant that **Po1 should be the first one up for removal from the model**, as it may also be the cause for high VIF values of a few other variables. Finally, I checked the significance of partial correlation in the model, and the most significant correlation was once again for pairs Po1-Po2, U1-U2 and Wealth-Ineq.

- To deal with multicollinearity, I decided to try exclusion of high VIF and correlation values parameters from the model, step-by-step. Since Po2 appeared to be the most problematic parameter, I excluded it first and re-built the model. There was no significant change in R-squared, and only 6 parameters were significant, as in model 1. However, **removing Po2 significantly reduced VIF values of other variables (e.g. VIF of Po1 reduced from 104 to 5), showing that this remeial measure helped reduce multicollinearity within the model**. The prediction for the test data, however, was still on the lower side - 724, which seemed inaccurate compared to other similar variables. Also, there were still 6 parameters with VIF over 5 (reason to suspect collinearity), and Wealth with VIF larger than 10. So, Wealth was the next candidate for removal. **Model 2: all predictors except Po1, Residual standard error: 208.6 on 32 degrees of freedom, Multiple R-squared: 0.7976, Adjusted R-squared: 0.709. Prediction: 724**

- After excluding Wealth from the model, still only 6 predictors remained significant, suggesting that this version of the model is still not optimal. However, VIF values reduced once again, being >5 only for U1, Ineq and So. No significant change in R-squared was noticed, but the prediction seemed more real (although still under 1000) - 944 Crimes for the test data point. **Model 3: all predictors except Po1 and Wealth, Residual standard error: 207.9 on 33 degrees of freedom, Multiple R-squared: 0.7927, Adjusted R-squared: 0.711. Prediction: 944**

- Exclusion of U1 once again did not produce change in the number of significant predictors - only 6 remained such, just as in the previous models. No significant change in R-squared was seen. **VIF values this time were all under 5, except for Ineq with 5.44**. The prediction seemed to be the most accurate so far - 1225 Crime rate for the test data. **At this point it was obvious that multicollinearity has reduced, however to produce the best possible model for the given data, I needed to exclude the other insignificant predictors.** This would give the model more predictive power by reducing the inclusion of 'randomness' with the parameters that do no impact the regression line. **Model 4: all predictors except Po1, Wealth and U1. Residual standard error: 210.7 on 34 degrees of freedom, Multiple R-squared: 0.7806, Adjusted R-squared: 0.7032**

- With the final set of 6 variables (**% of Males [M], mean years of schooling in adults <25 y. [Ed], Police protection expenditures in 1959 [Po1], Unemployment of urban males 35-39y. [U2], income inequality [Ineq], pobability of imprisonment [Prob]**), the metric of the model have significantly improved. Prediction increased to **1304 Crime rate** for the test data, **residual standard error** reduced to 200 (compared to 208 with the first model). **Multiple R-Squared** slightly lowered to 76.6, however **Adjusted R-squared** (metric using only significant predictors) improved to the never seen before 73.1, suggesting that the set of parameters was chosen correctly this time. R-squared might have reduced due to the work with multicollinearity - fewer variables help lower overfitting, so the general 'explanation of the data' (R-squared) reduced due to some unexplained randomness. Moreover, VIF was lower than 4 for each of the parameters, the four plots for residuals indicated no problem with the data points as before, and the multicollinearity diagnostic measure indicated no collinearity withing data vectors of the model this time. **Model with final predictors set: M + Ed + Po1 + U2 + Ineq + Prob, Residual standard error: 200.7 on 40 degrees of freedom, Multiple R-squared: 0.7659, Adjusted R-squared: 0.7307.**

After determining the set of parameters for the model, it is important to estimate its quality. To do this, I chose 5-fold cross validation (instead of AIC or BIC): CV is the general practice for linear regression quality estimation, but also because AIC and BIC may sometimes lead us toward choosing an over-complicated or over-simplified model respectively. Moreover, both AIC and BIC are related to cross-validation, but cross-validation does not produce their common problems. For cross-validation, I chose 5 as the number of folds instead of the commonly used 10 because of a small number of data points in the data set: too many folds may each have 'incomplete' representation of data and the results might be inaccurate.

- After cross validation, the true **R-Squared value** for the model with the final set of 6 predictors was found to be **63.4**, compared to 76.6 without CV. **Adjusted R-squared** reduced to from 73 to **57.9** showing that even with the use of significant factors only we had some overfitting in the model (and we still might have some left, since 6 predictors for 47 data points does not comply with one in ten rule discussed above). As for the first **model with all 15 predictors**, the **R-squared** after CV was 41.9 instead of 80, and **Adjusted R-squared** - 33.3 instead of 70. This once again demonstrates overfitting with the first model and gives an example of how important it is to reduce multicollinearity and use only significant parameters to increase the power of prediction of a linear regression model.

Further comments and reasoning for each step can be found in the solution below.

Solution in R

Step 0: Load the libraries

```
library(dplyr)
library(tidyverse)
library(dslabs)
library(data.table)
library(ggplot2)
library(plotly)
library(outliers)
library(qcc)
library(mctest)
library(ppcor)
library(car)
library(psych)
library(ggthemes)
library(corrplot)
library(DAAG)
```

Step 1: Load the dataset

```
data <- read.table("uscrime.txt",
                   header = TRUE,
                   stringsAsFactors = FALSE,
                   sep = "",
                   dec = ".")
head(data)
```

```
##      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
```

Step 2: Basic Explorations

Before performing regression, let's explore the variables and make some initial assumptions.

No NA values in the data set:

```r
is.null(data)
```

```
## [1] FALSE
```

```r
describe.by(data)
```

```
## Warning: describe.by is deprecated.  Please use the describeBy function
```

```
## Warning in describeBy(x = x, group = group, mat = mat, type = type, ...): no
## grouping variable requested
```

```
##        vars  n    mean     sd  median trimmed     mad     min     max   range
## M         1 47   13.86   1.26   13.60   13.75    1.19   11.90   17.70    5.80
## So        2 47    0.34   0.48    0.00    0.31    0.00    0.00    1.00    1.00
## Ed        3 47   10.56   1.12   10.80   10.59    1.19    8.70   12.20    3.50
## Po1       4 47    8.50   2.97    7.80    8.21    2.82    4.50   16.60   12.10
## Po2       5 47    8.02   2.80    7.30    7.76    2.82    4.10   15.70   11.60
## LF        6 47    0.56   0.04    0.56    0.56    0.05    0.48    0.64    0.16
## M.F       7 47   98.30   2.95   97.70   98.02    1.93   93.40  107.10   13.70
## Pop       8 47   36.62  38.07   25.00   29.95   22.24    3.00  168.00  165.00
## NW        9 47   10.11  10.28    7.60    8.56    7.71    0.20   42.30   42.10
## U1       10 47    0.10   0.02    0.09    0.09    0.02    0.07    0.14    0.07
## U2       11 47    3.40   0.84    3.40    3.35    0.89    2.00    5.80    3.80
## Wealth   12 47 5253.83 964.91 5370.00 5286.67 1111.95 2880.00 6890.00 4010.00
## Ineq     13 47   19.40   3.99   17.60   19.28    3.56   12.60   27.60   15.00
## Prob     14 47    0.05   0.02    0.04    0.05    0.02    0.01    0.12    0.11
## Time     15 47   26.60   7.09   25.80   26.35    6.37   12.20   44.00   31.80
## Crime    16 47  905.09 386.76  831.00  863.05  314.31  342.00 1993.00 1651.00
##         skew kurtosis     se
## M       0.82     0.38   0.18
## So      0.65    -1.61   0.07
## Ed     -0.32    -1.15   0.16
## Po1     0.89     0.16   0.43
## Po2     0.84     0.01   0.41
## LF      0.27    -0.89   0.01
## M.F     0.99     0.65   0.43
## Pop     1.85     3.08   5.55
## NW      1.38     1.08   1.50
## U1      0.77    -0.13   0.00
## U2      0.54     0.17   0.12
## Wealth -0.38    -0.61 140.75
## Ineq    0.37    -1.14   0.58
## Prob    0.88     0.75   0.00
```

```
## Time    0.37    -0.41   1.03
## Crime   1.05     0.78  56.42
```

We can see that our predictors have different scales. For example, U1 and U2 (unemployment of urban male aged 14-24 and 35-39) describe the same parameter (but for different age groups), but for U1 10% is displayed as 0.10, while for U2 10% is 10.0. **Skewness**: few of the predictors have fairly symmetrical data points (**skewness between -0.5 and 0.5**), most are moderately skewed (**between -1 and -0.5 or 0.5 and 1**), Pop (state population, 1960), NW (percentage of nonwhites), and the response variable Crime are highly skewed (**less than -1 or greater than 1**). As for **kurtosis**, most parameters have platykurtic distribution(<3), while population is mesokurtic.

Next, let's check boxplots for each data to visualize distribution of each predictor:

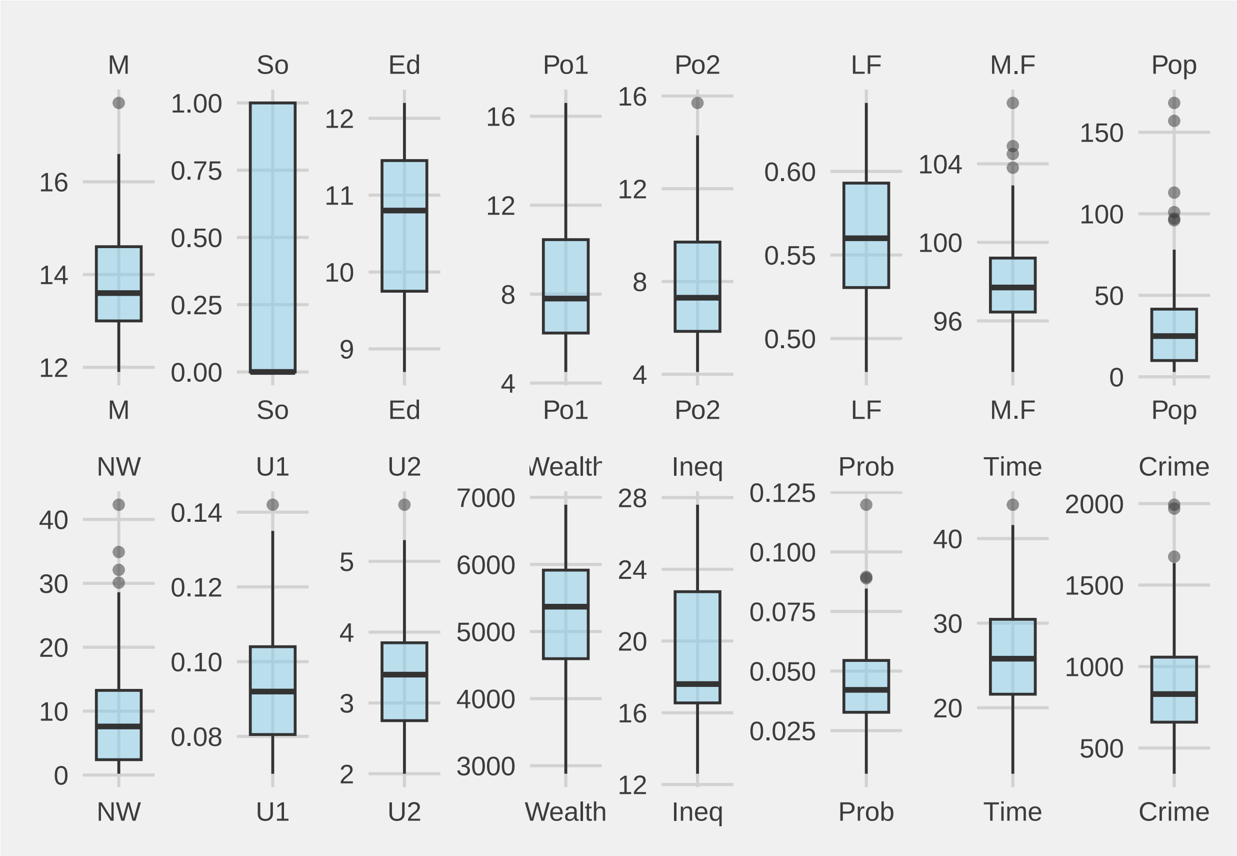
```r
#melt data for easier visualization
melted<-melt(data)
```

```
## Warning in melt(data): The melt generic in data.table has been passed a
## data.frame and will attempt to redirect to the relevant reshape2 method; please
## note that reshape2 is deprecated, and this redirection is now deprecated as
## well. To continue using melt methods from reshape2 while both libraries are
## attached, e.g. melt.list, you can prepend the namespace like
## reshape2::melt(data). In the next version, this warning will become an error.
```

```
## No id variables; using all as measure variables
```

```r
#boxplots
box_plots <- ggplot(melted,
                    aes(x=factor(variable), y=value))+
              geom_boxplot(alpha=.5, fill="skyblue")+
              facet_wrap(~variable, ncol=8, scale="free")+
              theme_fivethirtyeight()


box_plots
```



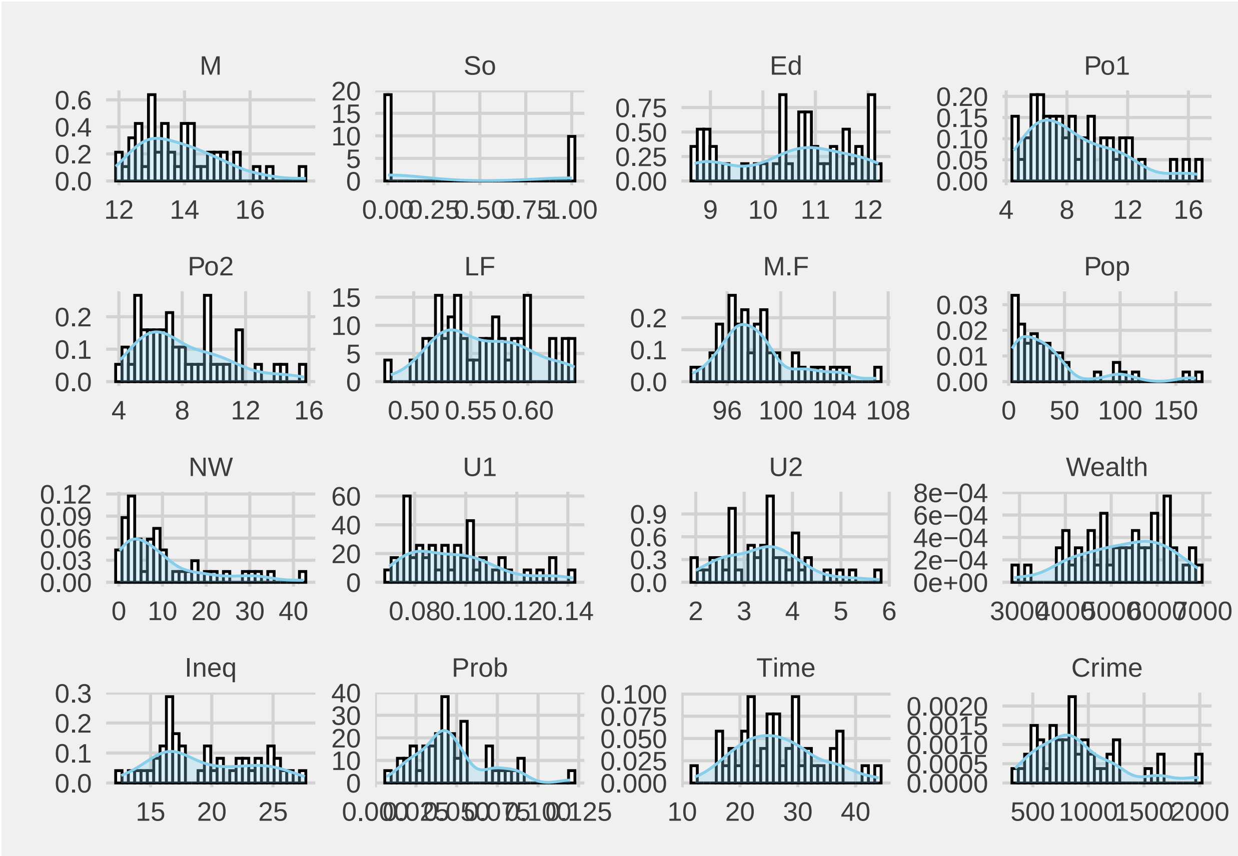
We have one binary variable, So (indicator for southern states). Some variables might have outliers (Pop, NW), many are skewed to the right.

Let's also plot a histogram with a density curve for each variable to see if any of them are normally distributed:

```
hist_plots <- ggplot(melted,
              aes(x=value))+
              geom_histogram(aes(y=..density..), colour="black", fill="white")+
              geom_density(alpha=.3, color="skyblue", fill="skyblue")+
              facet_wrap(~variable, scale="free")+
              theme_fivethirtyeight()
hist_plots
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.

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



The graphs support our assumption that most variables are right-skewed. The only normally distributed variable appears to be Time.

Even though the predictors and the response variable are not normally distributed, we can still do regression, since it does not require normal data.

Step 3: Pairwise Correlation

Before building our regression model, I would like to look at the variables and check if any of them have pairwise correlations. This step is important for feature selection - removing one the highly correlative features from a pair helps us save more predictive power for the model.

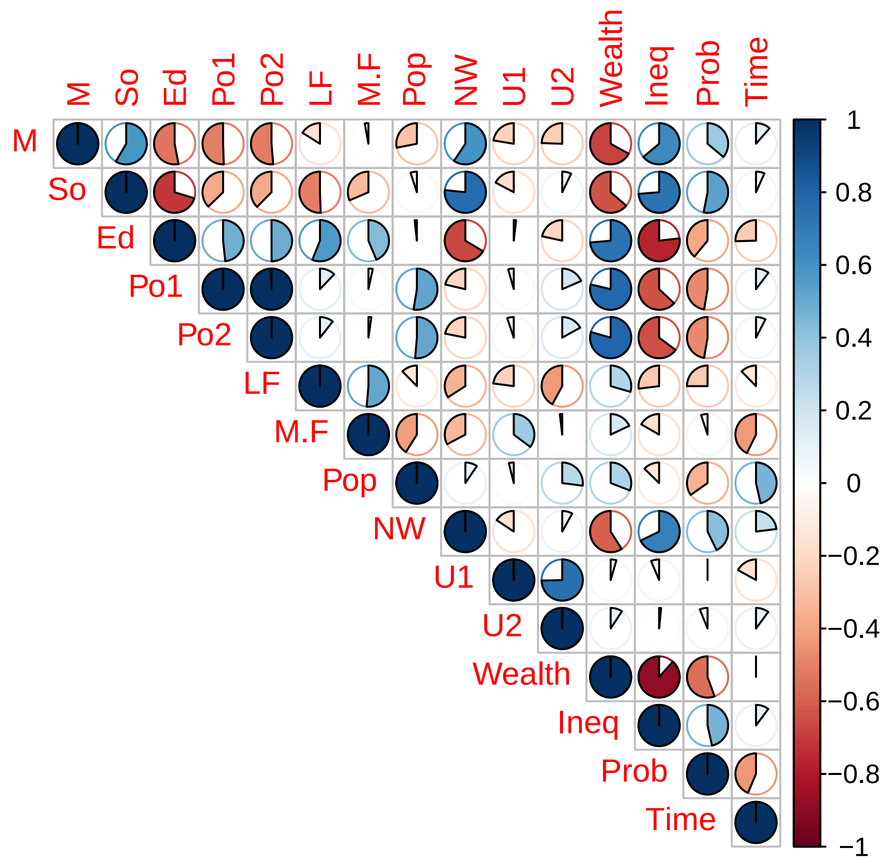
Knowing the description of the variables, I would suspect that the following predictors might have high correlation, since they are describing similar factors:

- Po1 and Po2, as those are expenditures on police protection in two consequent years (1959, 1960)
- U1 and U2, since they show urban male unemployment rate for two age groups (14-24, 35-39)
- Wealth and Ineq, since wealth parameter of a family and it's inequality level seem to be describing the same parameter (inequality is probably (to a large extent) based on wealth)
- NW and Ineq, more inequality in states with higher percentage of non-white population
- NW and Ed, states with higher non-white population number might have less mean years of schooling
- Ed-Wealth(Ineq) - states with wealthier population might have more mean years of schooling among adults
- So-NW - southern states migh have had more non-white population in 1960s

- Po1 / Po2 and wealth / ineq - people who have a job should be wealthier (hence higher equality for such individuals)

Keeping that in mind, let's build a correlation plot for predictors to visually inspect which variables might have high correlation:

```
corrplot(cor(data[,-16]), method='pie', type="upper")
```



Based on the plot, we can see that for the following pairs of variables correlation is very high:

- Po1-Po2 (correlation is almost 1!)
- Wealth-Ineq
- U1-U2
- So-Ed, So-NW
- Po1(Po2)-Wealth, Po1(Po2)-Ineq
- NW-Ineq, NW-Wealth
- Ed-NW, Ed-Wealth, Ed-Ineq

Get exact correlation values to refer to further on:

```
cm <- as.table(round(cor(data[,-16]),1))
cm
```

```
##           M   So   Ed  Po1  Po2   LF  M.F  Pop   NW   U1   U2 Wealth Ineq Prob
## M       1.0  0.6 -0.5 -0.5 -0.5 -0.2  0.0 -0.3  0.6 -0.2 -0.2   -0.7  0.6  0.4
## So      0.6  1.0 -0.7 -0.4 -0.4 -0.5 -0.3  0.0  0.8 -0.2  0.1   -0.6  0.7  0.5
```

```
## Ed     -0.5 -0.7  1.0  0.5  0.5  0.6  0.4  0.0 -0.7  0.0 -0.2    0.7 -0.8 -0.4
## Po1    -0.5 -0.4  0.5  1.0  1.0  0.1  0.0  0.5 -0.2  0.0  0.2    0.8 -0.6 -0.5
## Po2    -0.5 -0.4  0.5  1.0  1.0  0.1  0.0  0.5 -0.2 -0.1  0.2    0.8 -0.6 -0.5
## LF     -0.2 -0.5  0.6  0.1  0.1  1.0  0.5 -0.1 -0.3 -0.2 -0.4    0.3 -0.3 -0.3
## M.F     0.0 -0.3  0.4  0.0  0.0  0.5  1.0 -0.4 -0.3  0.4  0.0    0.2 -0.2 -0.1
## Pop    -0.3  0.0  0.0  0.5  0.5 -0.1 -0.4  1.0  0.1  0.0  0.3    0.3 -0.1 -0.3
## NW      0.6  0.8 -0.7 -0.2 -0.2 -0.3 -0.3  0.1  1.0 -0.2  0.1   -0.6  0.7  0.4
## U1     -0.2 -0.2  0.0  0.0 -0.1 -0.2  0.4  0.0 -0.2  1.0  0.7    0.0 -0.1  0.0
## U2     -0.2  0.1 -0.2  0.2  0.2 -0.4  0.0  0.3  0.1  0.7  1.0    0.1  0.0 -0.1
## Wealth -0.7 -0.6  0.7  0.8  0.8  0.3  0.2  0.3 -0.6  0.0  0.1    1.0 -0.9 -0.6
## Ineq    0.6  0.7 -0.8 -0.6 -0.6 -0.3 -0.2 -0.1  0.7 -0.1  0.0   -0.9  1.0  0.5
## Prob    0.4  0.5 -0.4 -0.5 -0.5 -0.3 -0.1 -0.3  0.4  0.0 -0.1   -0.6  0.5  1.0
## Time    0.1  0.1 -0.3  0.1  0.1 -0.1 -0.4  0.5  0.2 -0.2  0.1    0.0  0.1 -0.4
##        Time
## M       0.1
## So      0.1
## Ed     -0.3
## Po1     0.1
## Po2     0.1
## LF     -0.1
## M.F    -0.4
## Pop     0.5
## NW      0.2
## U1     -0.2
## U2      0.1
## Wealth  0.0
## Ineq    0.1
## Prob   -0.4
## Time    1.0
```

What does this mean for our model? I will start with a regression using all parameters (to refer to it when trying other combinations of parameters), however we would probably **need to remove several parameters with high correlation from above** (1 from each pair). As we can see, the number of such parameters is high. This could be due to one or two parameters that correlates with many predictors at once and 'drags' other pairs into high correlation value, or it could be several predictors which we would need to remove from the model in order to give the prediction more power. Further on, I will also **test the model for multicollinearity** to support or refute assumptions from above.

Step 4: Linear regression (all predictors)

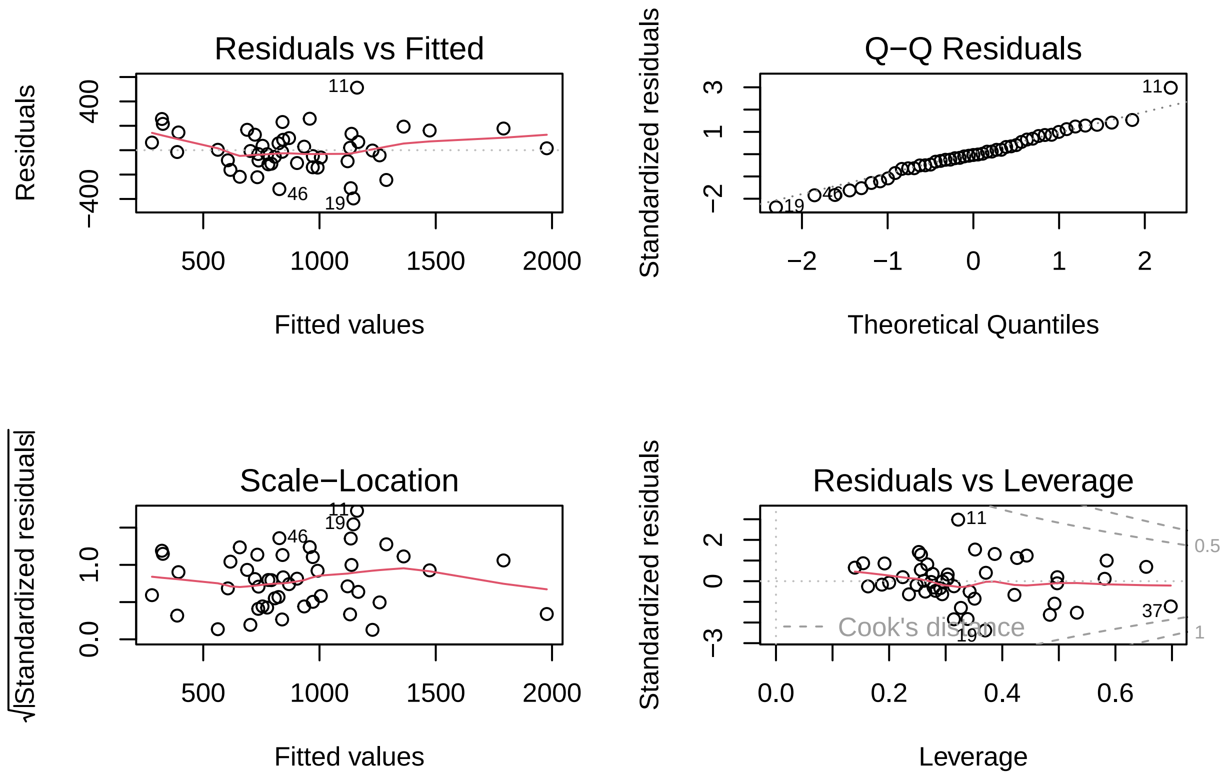
Let's start with a regression model using all parameters and see how many of them would be significant, and how much data it would explain (R-squared):

```r
par(mfrow=c(2,2))
#regression:
model_1 <- lm(Crime ~ ., data=data)
#get model summary
summary(model_1, vif=TRUE)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = 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 *
## 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
```

```r
#plot results
plot(model_1)
```



For a regression model with all parameters, we get R square od 79.76% (the model explains 79% of the data). We can see that only 6 predictors were found to be significant - only of of the police expenditures variables, one of the unemployment rates, Inequality and not wealth - so perhaps initial assumptions were correct and variables with high correlation to others should be removed.

From the plots we can see that: **1. Residuals vs Fitted**: Residuals do not seem to show non-linear patterns - the relationship between predictor variables and response variable is linear, there is no distinct pattern, the residuals are spread around a horizontal line. **2. Normal Q-Q**: Residuals are normally distributed - they follow a straight line well **3. Scale-Location**: Horizontal line with randomly spread points confirms homoscedasticity - residuals are spread equally across the predictor range. **4. Residuals vs Leverage**: Checking for influential outlier cases, we do not have outlying values in the upper-right or lower-right corners (where outliers could be influential). This means that **outliers do not influence the regression line, so it would not make a difference if we removed / left them**. As for Cook's distance, all our values are beyond the 0.1 distance, so all cases are influential and all data points affect the fitted values. Based on the plot, we do not have problematic cases with the model.

Let's make a prediction for the given data point using this model:

- Create a data frame using give parameter values:

```
test_data <- 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)
test_data
```

```
##    M So Ed Po1  Po2   LF M.F Pop  NW   U1  U2 Wealth Ineq Prob Time
## 1 14  0 10  12 15.5 0.64  94 150 1.1 0.12 3.6   3200 20.1 0.04   39
```

- Make a prediction:

```
pred_model_1 <- predict(model_1, test_data)
pred_model_1
```

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

We have a very low predicted crime rate, 155 - it is a lot smaller than the mean of 905. Since all of the parameter values of our test data point are **withing the range of other data points**, and the values of the test point are not abnormal, the problem lies with the model itself and the use of so many insignificant parameters. This is a demonstration why we **need to have only significant predictors in our regression model** - otherwise, predictions would be inaccurate just like this one.

We can go ahead and build a regression model using only the 6 significant predictors. However, before that, I would like to dive deeper and examine our model for multicollinearity.

Step 5: Detecting Multicollinearity

Since the diagnostic plots for our model with all parameters do not show any problematic cases, the reason for getting so many insignificant coefficients might be **multicollinearity**.

To inspect this issue, let's start with refering to the correlation matrix

```
cm
```

```
##           M   So   Ed  Po1  Po2   LF  M.F  Pop   NW   U1   U2 Wealth Ineq Prob
## M       1.0  0.6 -0.5 -0.5 -0.5 -0.2  0.0 -0.3  0.6 -0.2 -0.2   -0.7  0.6  0.4
## So      0.6  1.0 -0.7 -0.4 -0.4 -0.5 -0.3  0.0  0.8 -0.2  0.1   -0.6  0.7  0.5
## Ed     -0.5 -0.7  1.0  0.5  0.5  0.6  0.4  0.0 -0.7  0.0 -0.2    0.7 -0.8 -0.4
## Po1    -0.5 -0.4  0.5  1.0  1.0  0.1  0.0  0.5 -0.2  0.0  0.2    0.8 -0.6 -0.5
## Po2    -0.5 -0.4  0.5  1.0  1.0  0.1  0.0  0.5 -0.2 -0.1  0.2    0.8 -0.6 -0.5
## LF     -0.2 -0.5  0.6  0.1  0.1  1.0  0.5 -0.1 -0.3 -0.2 -0.4    0.3 -0.3 -0.3
## M.F     0.0 -0.3  0.4  0.0  0.0  0.5  1.0 -0.4 -0.3  0.4  0.0    0.2 -0.2 -0.1
## Pop    -0.3  0.0  0.0  0.5  0.5 -0.1 -0.4  1.0  0.1  0.0  0.3    0.3 -0.1 -0.3
## NW      0.6  0.8 -0.7 -0.2 -0.2 -0.3 -0.3  0.1  1.0 -0.2  0.1   -0.6  0.7  0.4
## U1     -0.2 -0.2  0.0  0.0 -0.1 -0.2  0.4  0.0 -0.2  1.0  0.7    0.0 -0.1  0.0
## U2     -0.2  0.1 -0.2  0.2  0.2 -0.4  0.0  0.3  0.1  0.7  1.0    0.1  0.0 -0.1
## Wealth -0.7 -0.6  0.7  0.8  0.8  0.3  0.2  0.3 -0.6  0.0  0.1    1.0 -0.9 -0.6
## Ineq    0.6  0.7 -0.8 -0.6 -0.6 -0.3 -0.2 -0.1  0.7 -0.1  0.0   -0.9  1.0  0.5
## Prob    0.4  0.5 -0.4 -0.5 -0.5 -0.3 -0.1 -0.3  0.4  0.0 -0.1   -0.6  0.5  1.0
## Time    0.1  0.1 -0.3  0.1  0.1 -0.1 -0.4  0.5  0.2 -0.2  0.1    0.0  0.1 -0.4
##        Time
## M       0.1
## So      0.1
## Ed     -0.3
```

```
## Po1     0.1
## Po2     0.1
## LF     -0.1
## M.F    -0.4
## Pop     0.5
## NW      0.2
## U1     -0.2
## U2      0.1
## Wealth  0.0
## Ineq    0.1
## Prob   -0.4
## Time    1.0
```

Once again we can see that there are pairs with high correlation. To narrow down the search, let's look into values with >0.8 correlation: - Po1-Po2 (1.0) - Ineq-Wealth (0.9) - NW-So (0.8) - Wealth-Po1(Po2) (0.8) - Ed-Ineq (0.8)

Removing these problematic variables step-by-step, starting with the ones with higher values, might help us avoid multicollinearity.

Let's perform further diagnostics.

Farrar-Glauber Test

Overall diagnostic for multicollinearity in the model:

```r
#Overall Multicollinearity Diagnostics Measures
omcdiag(model_1)
```

```
##
## Call:
## omcdiag(mod = model_1)
##
##
## Overall Multicollinearity Diagnostics
##
##                        MC Results detection
## Determinant |X'X|:         0.0000         1
## Farrar Chi-Square:       683.4092         1
## Red Indicator:             0.4142         0
## Sum of Lambda Inverse:   282.0910         1
## Theil's Method:            0.5705         1
## Condition Number:        292.1574         1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

5 out of 6 tests confirm multicollinearity in our model with all parameters. We have a zero value of standardized determinant, which signalizes that the set of vectors we used for our model are **linearly dependent**. Adn linear dependence defines multicollinearity. We also have a positive and high value for Farrar Chi-Square (497). All this implies presence of multicollinearity.

Explore further to locate which variables cause multicollinearity:

```r
#Individual Multicollinearity Diagnostic Measures
#locate where mc is
imcdiag(model_1)
```

```
##
```

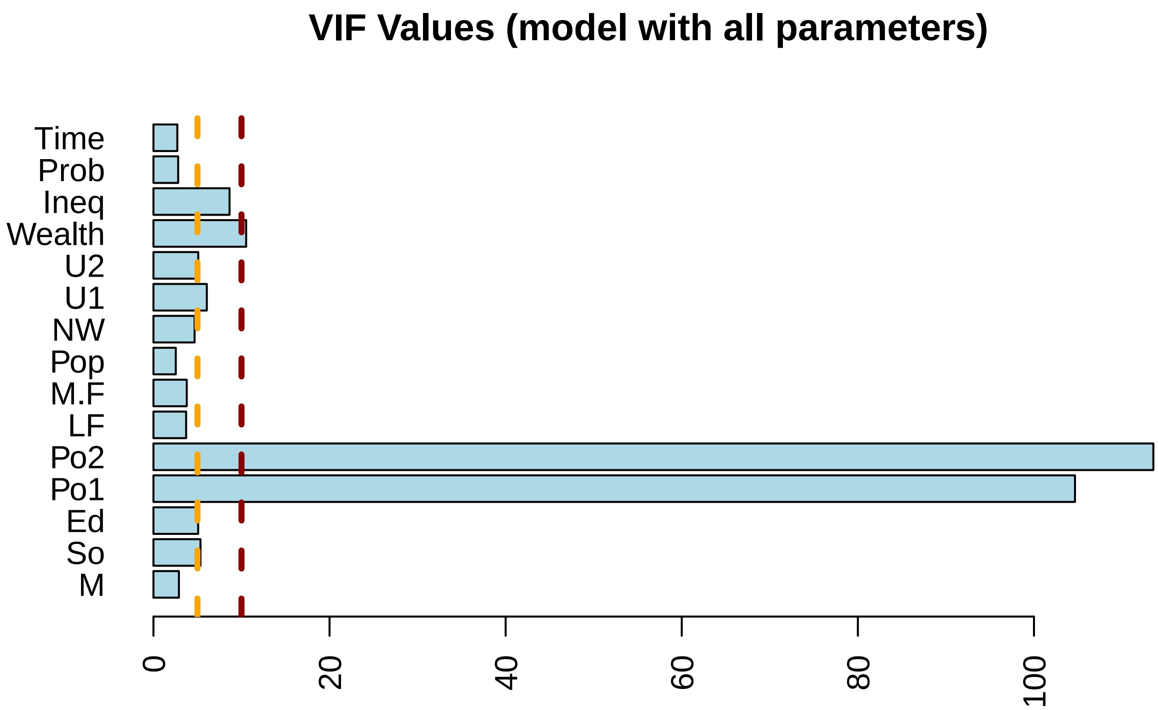
```
## Call:
## imcdiag(mod = model_1)
##
##
## All Individual Multicollinearity Diagnostics Result
##
##              VIF    TOL       Wi       Fi Leamer     CVIF Klein   IND1   IND2
## M         2.8924 0.3457   4.3256   4.8039 0.5880  -0.8136     0 0.1513 0.8307
## So        5.3428 0.1872   9.9264  11.0240 0.4326  -1.5028     1 0.0819 1.0321
## Ed        5.0774 0.1969   9.3199  10.3504 0.4438  -1.4281     0 0.0862 1.0196
## Po1     104.6587 0.0096 236.9341 263.1335 0.0977 -29.4374     1 0.0042 1.2576
## Po2     113.5593 0.0088 257.2783 285.7274 0.0938 -31.9409     1 0.0039 1.2585
## LF        3.7127 0.2693   6.2004   6.8861 0.5190  -1.0443     0 0.1178 0.9277
## M.F       3.7859 0.2641   6.3678   7.0720 0.5139  -1.0649     0 0.1156 0.9343
## Pop       2.5367 0.3942   3.5125   3.9009 0.6279  -0.7135     0 0.1725 0.7692
## NW        4.6741 0.2139   8.3979   9.3265 0.4625  -1.3147     0 0.0936 0.9981
## U1        6.0639 0.1649  11.5747  12.8546 0.4061  -1.7056     1 0.0721 1.0603
## U2        5.0889 0.1965   9.3460  10.3795 0.4433  -1.4314     1 0.0860 1.0202
## Wealth   10.5304 0.0950  21.7837  24.1925 0.3082  -2.9619     1 0.0415 1.1491
## Ineq      8.6445 0.1157  17.4732  19.4053 0.3401  -2.4315     1 0.0506 1.1228
## Prob      2.8095 0.3559   4.1359   4.5932 0.5966  -0.7902     0 0.1557 0.8178
## Time      2.7138 0.3685   3.9172   4.3504 0.6070  -0.7633     0 0.1612 0.8018
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## So , Po1 , Po2 , LF , M.F , Pop , NW , U1 , U2 , Wealth , Time , coefficient(s) are non-significant m
##
## R-square of y on all x: 0.8031
##
## * use method argument to check which regressors may be the reason of collinearity
## ===================================
```

As we can see, the fact that So , Po1 , Po2 , LF , M.F , Pop , NW , U1 , U2 , Wealth , Time coefficients are non-significant may be due to multicollinearity.

Let's inspect VIF values (**variance inflation factor**): VIF measure the effect of multicollinearity in the model on the variance of a coefficient in regression. **High VIF value** suggests that the independent variable has high collinearity with other variables of the model. VIF > 10 signalizes that there is a serious collinearity problem, and VIF > 5 gives reason to suspect collinearity. Ideally, we would have VIF lower than 5.

In our model there is definitely a problem with Po1, Po2 and Wealth (all of them might cause multicollinearity, or incluson of one of them could increase values for others)

```
vif_1 <- vif(model_1)
barplot(vif_1,
        main="VIF Values (model with all parameters)",
        horiz=TRUE,
        col="lightblue",
        las=2)
abline(v=5, lwd=3, lty=2, col="orange")
abline(v=10, lwd=3, lty=2, col="darkred")
```



We can also suspect that we would need to remove Ineq, U1 or So in the future.

As a third part of multicollinearity inspection, let's check the significance of partial correlation in the model:

```
pcor(data[,-16], method = "pearson")
```

```
## $estimate
##                  M           So          Ed         Po1         Po2          LF
## M       1.00000000  0.095011545 -0.07049732  0.03169183 -0.05226542 -0.14868894
## So      0.09501154  1.000000000  0.05946963  0.02727670 -0.06836453 -0.47285693
## Ed     -0.07049732  0.059469631  1.00000000 -0.16218269  0.19055536  0.32326037
## Po1     0.03169183  0.027276701 -0.16218269  1.00000000  0.97414941  0.21450483
## Po2    -0.05226542 -0.068364529  0.19055536  0.97414941  1.00000000 -0.28027657
## LF     -0.14868894 -0.472856933  0.32326037  0.21450483 -0.28027657  1.00000000
## M.F     0.31621752  0.154695672  0.09222478  0.11328490 -0.04278725  0.50334722
## Pop    -0.08239457  0.003567401 -0.01437748  0.05388079  0.04212623  0.14441506
## NW      0.28833759  0.421173001 -0.15936368 -0.18684908  0.29135095  0.33821514
## U1     -0.07992157 -0.379233033  0.21938511  0.01320785 -0.10476108 -0.43514880
## U2     -0.17487946  0.221699050 -0.30599626  0.08350954 -0.02036983  0.05006294
## Wealth -0.15613390  0.230659509  0.14731235  0.03706924  0.05636398  0.13371662
## Ineq   -0.06286788  0.374063514 -0.20797232  0.07715669 -0.09621237  0.18740663
## Prob   -0.08038454  0.143948019  0.03467900  0.22720908 -0.26933707 -0.09274065
## Time    0.15800936 -0.131514248 -0.03460833  0.30176053 -0.35233142 -0.07661089
##                M.F          Pop          NW          U1          U2      Wealth
## M       0.31621752 -0.082394570  0.28833759 -0.07992157 -0.17487946 -0.15613390
## So      0.15469567  0.003567401  0.42117300 -0.37923303  0.22169905  0.23065951
## Ed      0.09222478 -0.014377485 -0.15936368  0.21938511 -0.30599626  0.14731235
```

```
## Po1     0.11328490  0.053880787 -0.18684908  0.01320785  0.08350954  0.03706924
## Po2    -0.04278725  0.042126230  0.29135095 -0.10476108 -0.02036983  0.05636398
## LF      0.50334722  0.144415059  0.33821514 -0.43514880  0.05006294  0.13371662
## M.F     1.00000000 -0.386754153 -0.19043146  0.52616718 -0.17709103  0.10915081
## Pop    -0.38675415  1.000000000 -0.03532650  0.17781992 -0.03724395  0.08715964
## NW     -0.19043146 -0.035326503  1.00000000  0.19154070 -0.02381937 -0.20176157
## U1      0.52616718  0.177819916  0.19154070  1.00000000  0.77145555 -0.08395003
## U2     -0.17709103 -0.037243949 -0.02381937  0.77145555  1.00000000  0.15849903
## Wealth  0.10915081  0.087159644 -0.20176157 -0.08395003  0.15849903  1.00000000
## Ineq    0.20277254  0.249790560  0.07861383 -0.06796370  0.09456132 -0.59504042
## Prob   -0.05080410  0.002127945  0.31114141 -0.03593834 -0.01033791 -0.10104856
## Time   -0.17420993  0.239975925  0.29541907 -0.13533412  0.10434401  0.11425654
##               Ineq         Prob        Time
## M      -0.06286788 -0.080384543  0.15800936
## So      0.37406351  0.143948019 -0.13151425
## Ed     -0.20797232  0.034678996 -0.03460833
## Po1     0.07715669  0.227209080  0.30176053
## Po2    -0.09621237 -0.269337066 -0.35233142
## LF      0.18740663 -0.092740646 -0.07661089
## M.F     0.20277254 -0.050804095 -0.17420993
## Pop     0.24979056  0.002127945  0.23997592
## NW      0.07861383  0.311141410  0.29541907
## U1     -0.06796370 -0.035938338 -0.13533412
## U2      0.09456132 -0.010337909  0.10434401
## Wealth -0.59504042 -0.101048558  0.11425654
## Ineq    1.00000000 -0.147266786 -0.02193940
## Prob   -0.14726679  1.000000000 -0.56479730
## Time   -0.02193940 -0.564797297  1.00000000
##
## $p.value
##                 M          So          Ed          Po1          Po2           LF
## M      0.00000000 0.592997599 0.69196899 8.587811e-01 7.690952e-01 0.401329944
## So     0.59299760 0.000000000 0.73831167 8.782971e-01 7.008526e-01 0.004741031
## Ed     0.69196899 0.738311673 0.00000000 3.594601e-01 2.803702e-01 0.062204891
## Po1    0.85878112 0.878297096 0.35946011 0.000000e+00 3.035772e-22 0.223138556
## Po2    0.76909525 0.700852604 0.28037022 3.035772e-22 0.000000e+00 0.108380021
## LF     0.40132994 0.004741031 0.06220489 2.231386e-01 1.083800e-01 0.000000000
## M.F    0.06846226 0.382358437 0.60393398 5.235324e-01 8.101223e-01 0.002409512
## Pop    0.64317989 0.984024774 0.93567839 7.621607e-01 8.130036e-01 0.415149816
## NW     0.09818231 0.013116442 0.36798361 2.900054e-01 9.456367e-02 0.050413451
## U1     0.65320973 0.026970697 0.21253330 9.409015e-01 5.554358e-01 0.010110714
## U2     0.32255434 0.207628088 0.07840719 6.386779e-01 9.089647e-01 0.778578464
## Wealth 0.37789499 0.189372693 0.40575225 8.351232e-01 7.515363e-01 0.450893251
## Ineq   0.72392526 0.029304029 0.23789020 6.644944e-01 5.883119e-01 0.288542587
## Prob   0.65132742 0.416676065 0.84562284 1.962640e-01 1.234698e-01 0.601903014
## Time   0.37212076 0.458451552 0.84593365 8.284229e-02 4.098926e-02 0.666730673
##                M.F         Pop          NW           U1           U2       Wealth
## M      0.068462256 0.64317989 0.09818231 6.532097e-01 3.225543e-01 0.3778949896
## So     0.382358437 0.98402477 0.01311644 2.697070e-02 2.076281e-01 0.1893726932
## Ed     0.603933976 0.93567839 0.36798361 2.125333e-01 7.840719e-02 0.4057522499
## Po1    0.523532424 0.76216070 0.29000544 9.409015e-01 6.386779e-01 0.8351232235
## Po2    0.810122283 0.81300355 0.09456367 5.554358e-01 9.089647e-01 0.7515362531
## LF     0.002409512 0.41514982 0.05041345 1.011071e-02 7.785785e-01 0.4508932511
## M.F    0.000000000 0.02385044 0.28068895 1.392196e-03 3.163759e-01 0.5388932455
```

```
## Pop    0.023850442 0.00000000 0.84277588 3.143560e-01 8.343569e-01 0.6240288134
## NW     0.280688949 0.84277588 0.00000000 2.778439e-01 8.936293e-01 0.2525104482
## U1     0.001392196 0.31435601 0.27784390 0.000000e+00 9.271748e-08 0.6369027778
## U2     0.316375945 0.83435687 0.89362925 9.271748e-08 0.000000e+00 0.3706217731
## Wealth 0.538893246 0.62402881 0.25251045 6.369028e-01 3.706218e-01 0.0000000000
## Ineq   0.250090879 0.15423331 0.65853793 7.025265e-01 5.947586e-01 0.0002058034
## Prob   0.775383691 0.99047043 0.07326989 8.400876e-01 9.537275e-01 0.5696078975
## Time   0.324439470 0.17161858 0.08984095 4.453852e-01 5.570197e-01 0.5199534778
##                Ineq         Prob         Time
## M      0.7239252562 0.6513274218 0.3721207582
## So     0.0293040291 0.4166760646 0.4584515521
## Ed     0.2378902045 0.8456228424 0.8459336480
## Po1    0.6644943568 0.1962639795 0.0828422890
## Po2    0.5883118957 0.1234697741 0.0409892634
## LF     0.2885425875 0.6019030140 0.6667306726
## M.F    0.2500908788 0.7753836914 0.3244394697
## Pop    0.1542333067 0.9904704327 0.1716185790
## NW     0.6585379308 0.0732698927 0.0898409499
## U1     0.7025265442 0.8400875696 0.4453852316
## U2     0.5947586102 0.9537274580 0.5570197012
## Wealth 0.0002058034 0.5696078975 0.5199534778
## Ineq   0.0000000000 0.4058991013 0.9019826253
## Prob   0.4058991013 0.0000000000 0.0005016418
## Time   0.9019826253 0.0005016418 0.0000000000
##
## $statistic
##                 M         So          Ed         Po1        Po2         LF
## M       0.0000000  0.5399089 -0.39978776  0.17936616 -0.2960625 -0.8505666
## So      0.5399089  0.0000000  0.33700750  0.15435775 -0.3876351 -3.0357096
## Ed     -0.3997878  0.3370075  0.00000000 -0.92975305  1.0980643  1.9323865
## Po1     0.1793662  0.1543578 -0.92975305  0.00000000 24.3935698  1.2423406
## Po2    -0.2960625 -0.3876351  1.09806433 24.39356983  0.0000000 -1.6516844
## LF     -0.8505666 -3.0357096  1.93238652  1.24234059 -1.6516844  0.0000000
## M.F     1.8855502  0.8857534  0.52393504  0.64498827 -0.2422631  3.2952364
## Pop    -0.4676843  0.0201804 -0.08133974  0.30523916  0.2385137  0.8255894
## NW      1.7034304  2.6268644 -0.91316744 -1.07592654  1.7228746  2.0330434
## U1     -0.4535555 -2.3184515  1.27201804  0.07472142 -0.5958971 -2.7339923
## U2     -1.0047510  1.2861242 -1.81818986  0.47405719 -0.1152531  0.2835543
## Wealth -0.8941932  1.3409670  0.84251634  0.20983953  0.3193505  0.7632699
## Ineq   -0.3563393  2.2816646 -1.20276795  0.43776917 -0.5467960  1.0792537
## Prob   -0.4561999  0.8228629  0.19629209  1.31980687 -1.5820642 -0.5268911
## Time    0.9052075 -0.7504754 -0.19589164  1.79048085 -2.1296507 -0.4346541
##               M.F         Pop         NW          U1          U2     Wealth
## M       1.8855502 -0.46768430  1.7034304 -0.45355553 -1.00475100 -0.8941932
## So      0.8857534  0.02018040  2.6268644 -2.31845155  1.28612424  1.3409670
## Ed      0.5239350 -0.08133974 -0.9131674  1.27201804 -1.81818986  0.8425163
## Po1     0.6449883  0.30523916 -1.0759265  0.07472142  0.47405719  0.2098395
## Po2    -0.2422631  0.23851367  1.7228746 -0.59589712 -0.11525309  0.3193505
## LF      3.2952364  0.82558943  2.0330434 -2.73399226  0.28355432  0.7632699
## M.F     0.0000000 -2.37242739 -1.0973235  3.50013765 -1.01786608  0.6211616
## Pop    -2.3724274  0.00000000 -0.1999617  1.02219197 -0.21082986  0.4949329
## NW     -1.0973235 -0.19996169  0.0000000  1.10395797 -0.13478097 -1.1653006
## U1      3.5001377  1.02219197  1.1039580  0.00000000  6.85859917 -0.4765754
## U2     -1.0178661 -0.21082986 -0.1347810  6.85859917  0.00000000  0.9080849
```

```
## Wealth  0.6211616  0.49493294 -1.1653006 -0.47657542  0.90808488  0.0000000
## Ineq    1.1713893  1.45928839  0.4460876 -0.38535177  0.53732733 -4.1882228
## Prob   -0.2877630  0.01203750  1.8520086 -0.20342935 -0.05848317 -0.5745578
## Time   -1.0007836  1.39837085  1.7492139 -0.77267399  0.59349858  0.6505932
##              Ineq        Prob       Time
## M      -0.3563393 -0.45619994  0.9052075
## So      2.2816646  0.82286287 -0.7504754
## Ed     -1.2027680  0.19629209 -0.1958916
## Po1     0.4377692  1.31980687  1.7904808
## Po2    -0.5467960 -1.58206417 -2.1296507
## LF      1.0792537 -0.52689106 -0.4346541
## M.F     1.1713893 -0.28776297 -1.0007836
## Pop     1.4592884  0.01203750  1.3983708
## NW      0.4460876  1.85200860  1.7492139
## U1     -0.3853518 -0.20342935 -0.7726740
## U2      0.5373273 -0.05848317  0.5934986
## Wealth -4.1882228 -0.57455784  0.6505932
## Ineq    0.0000000 -0.84224996 -0.1241379
## Prob   -0.8422500  0.00000000 -3.8716203
## Time   -0.1241379 -3.87162034  0.0000000
##
## $n
## [1] 47
##
## $gp
## [1] 13
##
## $method
## [1] "pearson"
```

There is statistically significant partial correlation between 'po1' and 'po2'. The second high value is u1-u2 (6) and Wealth-Ineq(-4).

Step 6: Dealing with Multicollinearity

Although we can skip this step and move on to a model with only 6 significant parameters (based on our first regression), I would like to try removing predictors that cause multicollinearity step-by-step to see how it affects R-squared and other coefficients.

I will be excluding parameters based on VIF , recalculating the factor and removing another parameter (if necessary)

1. Excluding Po2

```r
#regression:
model_2 <- lm(Crime ~ .-Po2, data=data)
#get model summary
summary(model_2)
```

```
##
## Call:
## lm(formula = Crime ~ . - Po2, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -442.55 -116.46    8.86  118.26  473.49
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.379e+03  1.569e+03  -4.066 0.000291 ***
## M            8.986e+01  4.157e+01   2.162 0.038232 *
## So           5.669e+00  1.481e+02   0.038 0.969705
## Ed           1.773e+02  6.082e+01   2.915 0.006445 **
## Po1          9.653e+01  2.392e+01   4.035 0.000317 ***
## LF          -2.801e+02  1.408e+03  -0.199 0.843538
## M.F          1.822e+01  2.029e+01   0.898 0.376026
## Pop         -7.836e-01  1.286e+00  -0.609 0.546523
## NW           2.446e+00  6.187e+00   0.395 0.695239
## U1          -5.416e+03  4.178e+03  -1.296 0.204164
## U2           1.694e+02  8.215e+01   2.062 0.047441 *
## Wealth       9.072e-02  1.033e-01   0.878 0.386292
## Ineq         7.271e+01  2.256e+01   3.222 0.002921 **
## Prob        -4.285e+03  2.184e+03  -1.962 0.058484 .
## Time        -1.128e+00  6.692e+00  -0.168 0.867251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 208.6 on 32 degrees of freedom
## Multiple R-squared:  0.7976, Adjusted R-squared:  0.709
## F-statistic: 9.006 on 14 and 32 DF,  p-value: 1.673e-07
```

```r
#Check VIF
as.table(round(vif(model_2),2))
```

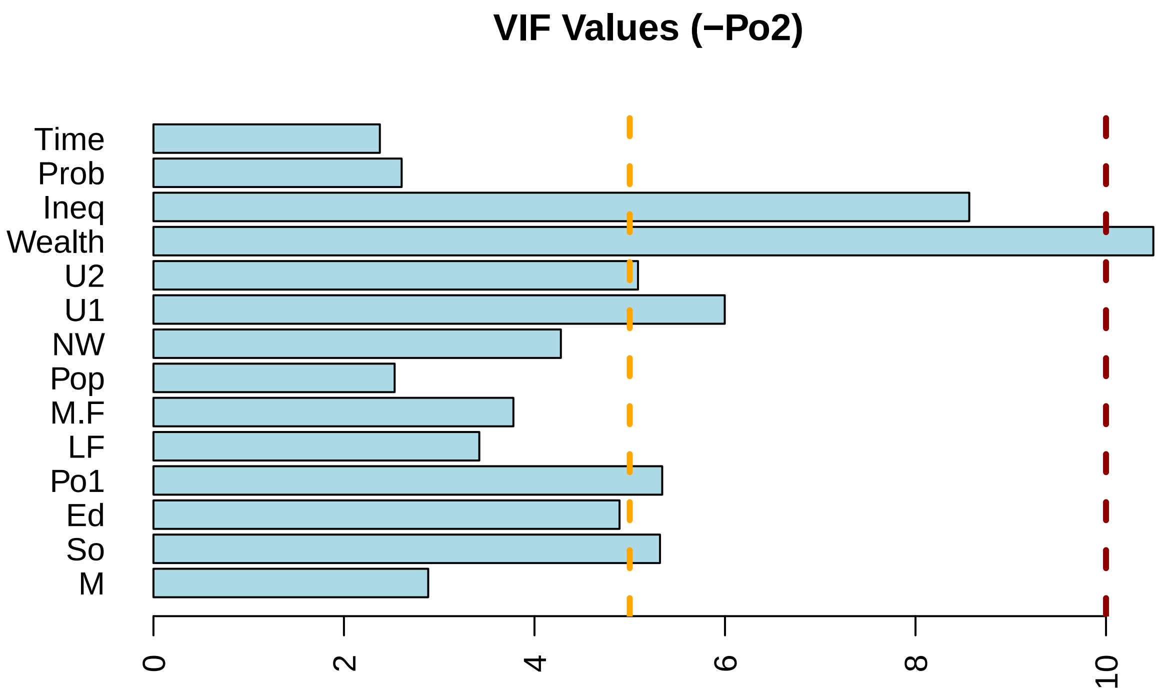
```
##      M     So     Ed    Po1     LF    M.F    Pop     NW     U1     U2 Wealth
##   2.88   5.32   4.89   5.34   3.42   3.78   2.53   4.28   6.00   5.09  10.50
##   Ineq   Prob   Time
##   8.56   2.61   2.38
```

```r
#check prediction
pred_model_2 <- predict(model_2, test_data)
pred_model_2
```

```
##        1
## 724.8202
```

No significant change in R-squared, and still only 6 significant parameters (same ones). However, **excluding Po2 made a huge difference in VIF** - Po1 does not have a value over 100. Also, Po1's significance code has changes from 0.1 to 0.001. Removing Po1 has helped us get a better prediction of the response variable - 724 lies withing the range of the Crime variable, although it still seems to be on the lower side of the crime variable (the test point has relatively low Wealth and high Ineq coefficients and other parameters similar to the rest of our data, so I would expect the value to be at least around the median).

```r
vif_2 <- vif(model_2)
barplot(vif_2,
        main="VIF Values (-Po2)",
        horiz=TRUE,
        col="lightblue",
        las=2)
abline(v=5, lwd=3, lty=2, col="orange")
abline(v=10, lwd=3, lty=2, col="darkred")
```



VIF plot has significantly changed -now most values are under 10, except for Wealth - our next candidate for exclusion.

2. Excluding Wealth

```
#regression:
model_3 <- lm(Crime ~ .-Po2 -Wealth, data=data)
#get model summary
summary(model_3)
```

```
##
## Call:
## lm(formula = Crime ~ . - Po2 - Wealth, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -469.4  -93.1   12.6  117.3  506.4
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6041.0176  1515.7345  -3.986 0.000351 ***
## M              84.0350    40.8957   2.055 0.047879 *
## So             35.2894   143.7092   0.246 0.807543
## Ed            185.9198    59.8202   3.108 0.003861 **
## Po1           105.0940    21.7659   4.828 3.06e-05 ***
## LF           -127.9865  1392.3561  -0.092 0.927317
## M.F            20.1254    20.1066   1.001 0.324141
```

```
## Pop            -0.6822     1.2761  -0.535 0.596494
## NW              1.3912     6.0482   0.230 0.819502
## U1          -5748.4126  4146.8729  -1.386 0.174980
## U2            180.7362    80.8400   2.236 0.032251 *
## Ineq           60.7323    17.9172   3.390 0.001829 **
## Prob        -4517.0792  2160.3360  -2.091 0.044315 *
## Time           -0.5337     6.6346  -0.080 0.936366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 207.9 on 33 degrees of freedom
## Multiple R-squared:  0.7927, Adjusted R-squared:  0.711
## F-statistic: 9.707 on 13 and 33 DF,  p-value: 7.32e-08
```

```r
#Check VIF
as.table(round(vif(model_3),2))
```

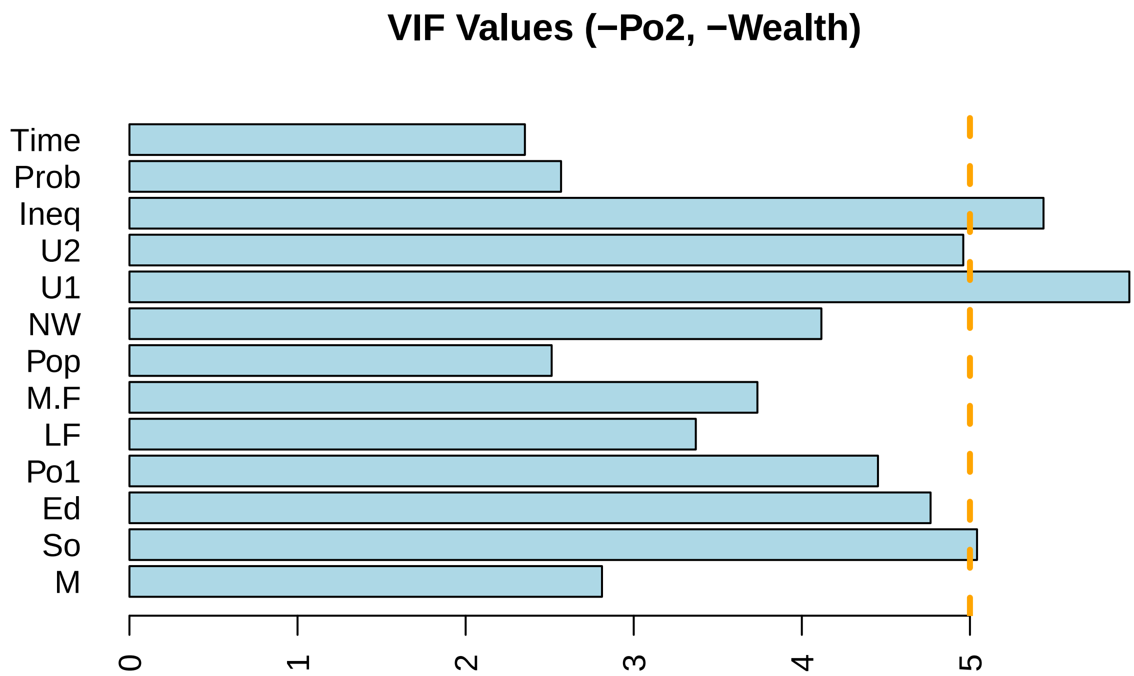
```
##    M   So   Ed  Po1   LF  M.F  Pop   NW   U1   U2 Ineq Prob Time
## 2.81 5.04 4.77 4.45 3.37 3.74 2.51 4.12 5.95 4.96 5.44 2.57 2.35
```

```r
#check prediction
pred_model_3 <- predict(model_3, test_data)
pred_model_3
```

```
##        1
## 944.9295
```

We still have the same 6 significant predictors. The **multiple R-Squared seems to be decreasing with each exclusion (80.31 – 79.76 – 79.27), but Adjusted R-Squared in increasing (70.78 – 70.9 – 71.1)** - so the explanation of the response variable using significant variables is increasing, as we exclude parameters with high VIF. Prediction is now closer to the mean value of Crime, meaning that removing insignificant variables has made our model more powerful in predictions.

```r
vif_3 <- vif(model_3)
barplot(vif_3,
        main="VIF Values (-Po2, -Wealth)",
        horiz=TRUE,
        col="lightblue",
        las=2)
abline(v=5, lwd=3, lty=2, col="orange")
abline(v=10, lwd=3, lty=2, col="darkred")
```



Now, all VIFs are under 10. However, we still have 2 similar parameters - U1 and U2, and U2 needs to be excluded, as its VIF is now the highest and it is not significant for our regression.

3. Excluding U1

```
#regression:
model_4 <- lm(Crime ~ .-Po2 -Wealth -U1 , data=data)
#get model summary
summary(model_4)
```

```
##
## Call:
## lm(formula = Crime ~ . - Po2 - Wealth - U1, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -440.00  -93.29   19.13  103.19  564.35
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5353.2369  1451.5076  -3.688 0.000784 ***
## M              87.5248    41.3676   2.116 0.041767 *
## So            116.5591   132.9728   0.877 0.386875
## Ed            169.9171    59.4859   2.856 0.007257 **
## Po1           119.4402    19.4058   6.155 5.43e-07 ***
## LF            723.8666  1266.2264   0.572 0.571304
## M.F             5.3647    17.2851   0.310 0.758178
```

```
## Pop            -0.9790     1.2750  -0.768 0.447866
## NW             -0.2120     6.0165  -0.035 0.972095
## U2             93.7073    51.6120   1.816 0.078259 .
## Ineq           60.8511    18.1582   3.351 0.001983 **
## Prob        -4525.7921  2189.4087  -2.067 0.046403 *
## Time            0.5327     6.6785   0.080 0.936893
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 210.7 on 34 degrees of freedom
## Multiple R-squared:  0.7806, Adjusted R-squared:  0.7032
## F-statistic: 10.08 on 12 and 34 DF,  p-value: 5.218e-08
```

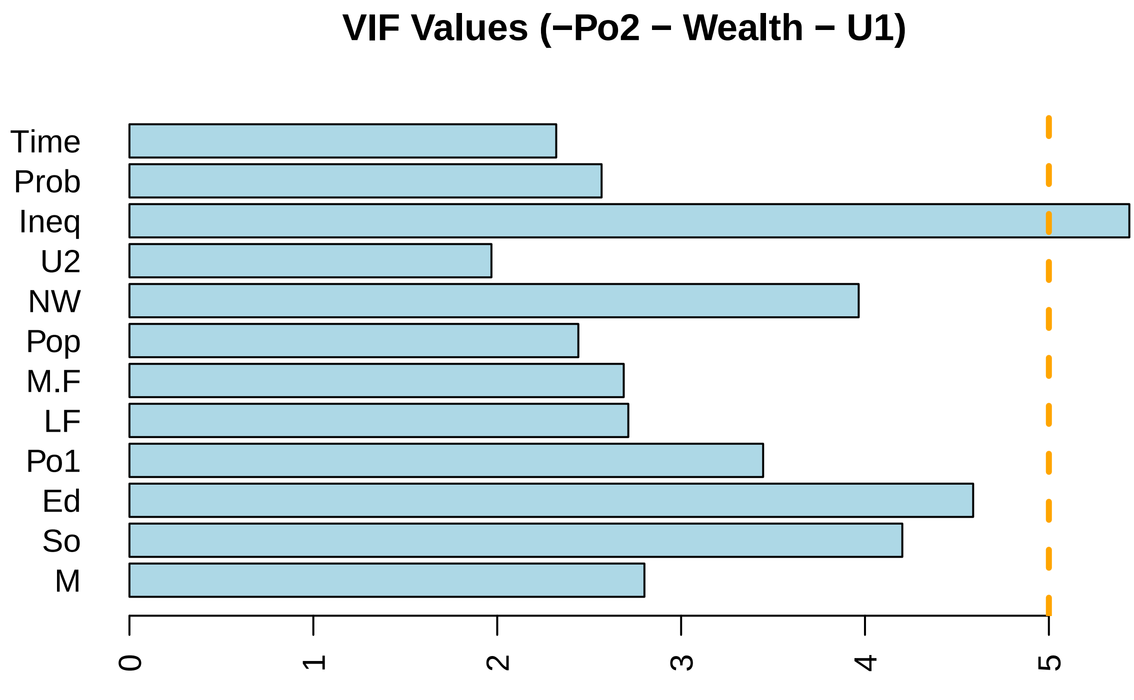
```r
#Check VIF
as.table(round(vif(model_4),2))
```

```
##    M   So   Ed  Po1   LF  M.F  Pop   NW   U2 Ineq Prob Time
## 2.80 4.20 4.59 3.45 2.71 2.69 2.44 3.97 1.97 5.44 2.57 2.32
```

```r
#check prediction
pred_model_4 <- predict(model_4, test_data)
pred_model_4
```

```
##        1
## 1225.232
```

```r
vif_4 <- vif(model_4)
barplot(vif_4,
        main="VIF Values (-Po2 - Wealth - U1)",
        horiz=TRUE,
        col="lightblue",
        las=2)
abline(v=5, lwd=3, lty=2, col="orange")
abline(v=10, lwd=3, lty=2, col="darkred")
```



Once again, we have a similar R-squared and adjusted R-squared (alhough adjusted R-squared also lowered a bit this time). We have seen that only 6 parameters are significant at each step, and although we managed to get rid of multicollinearity and reduce VIF (almost all predictors are <5 now, except for Ineq with 5.44), we did not get any new significant predictors. Although the prediction seems to be more accurate now (1225), we still need to leave only significant variables - with those that have low significance included, our model can fail to predict data points that are not so average and has predictors that are more on the extreme side.

Step 7: Final set of predictors

Since we have seen that each round only 6 predictors have stayed significant, lets use only them for our final model to increase its prediction power. **The 6 significant factors are:** -M percentage of males aged 14–24 in total state population -Ed mean years of schooling of the population aged 25 years or over -Po1 per capita expenditure on police protection in 1960 -U2 unemployment rate of urban males 35–39 -Ineq income inequality: percentage of families earning below half the median income -Prob probability of imprisonment: ratio of number of commitments to number of offenses

```r
par(mfrow=c(2,2))
#regression:
model_final <- lm(Crime ~ M+Ed+Po1+U2+Ineq+Prob , data=data)
#get model summary
summary(model_final)
```

```
##
## Call:
## lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = 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
```

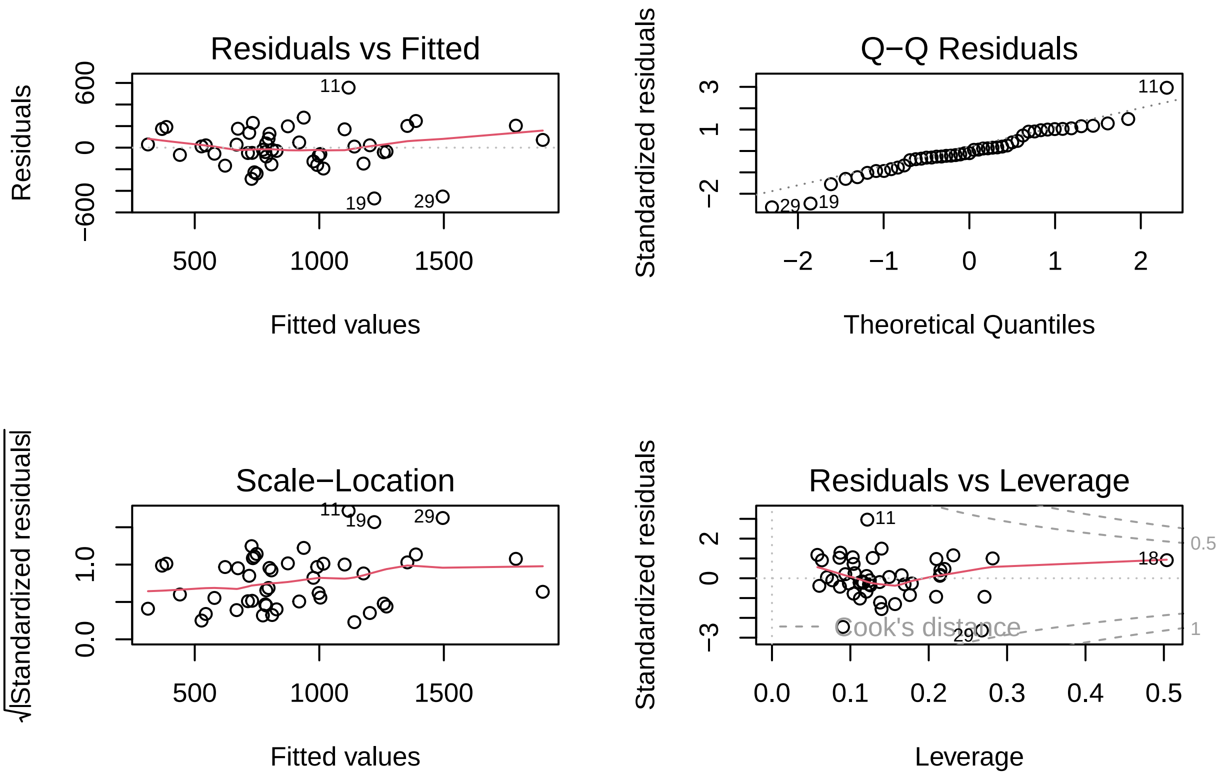
```r
#Check VIF
as.table(round(vif(model_final),2))
```

```
##    M   Ed  Po1   U2 Ineq Prob
## 2.00 2.86 1.91 1.36 3.53 1.38
```

```r
#check prediction
pred_model_final <- predict(model_final, test_data)
pred_model_final
```

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

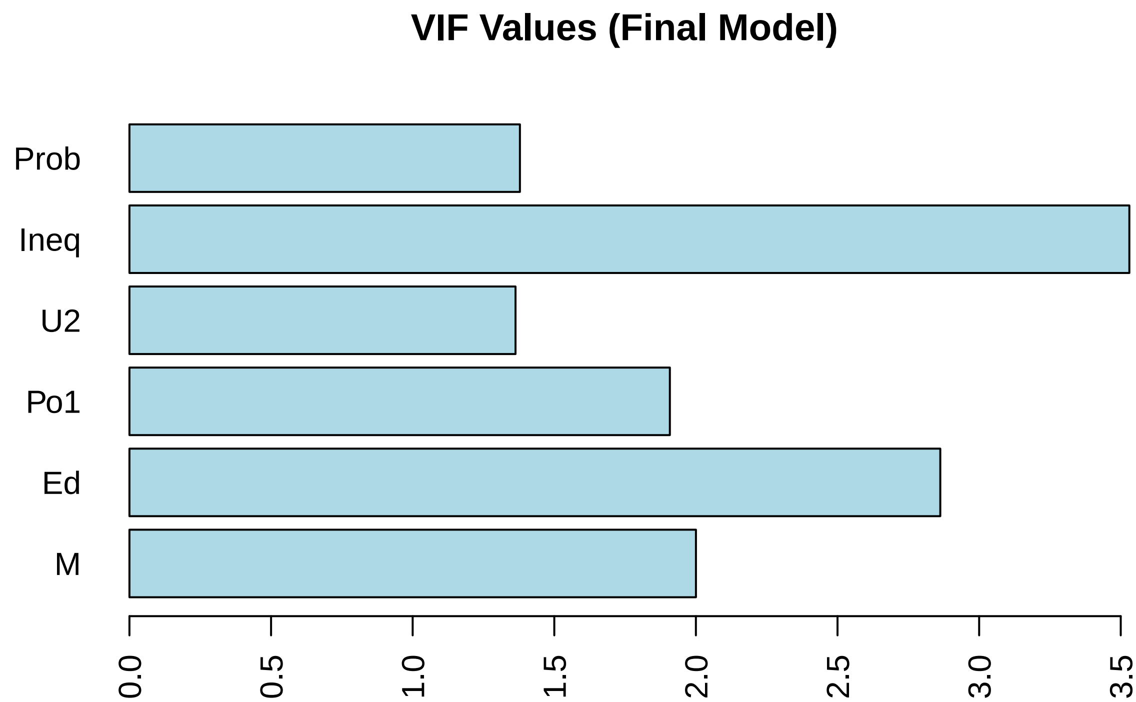
```r
#plot results
plot(model_final)
```



Although we have a lower Multiple R-Squared value, our **Adjusted R-squared is at 73%** - the highest it has been. **With the final set of predictors, our prediction for the test data point is a Crime rate of 1304.**

Plots for the final model do not indicate any problems: **1. Residuals vs Fitted**: Residuals show linear patterns - the relationship between the response and predictors is linear and can be explained with linear regression. **2. Normal Q-Q**: Residuals are normally distributed. **3. Scale-Location**: The line is horizontal and the points are spread randomly across it - homoscedasticity, residuals are equally spread across the range of response variable. **4. Residuals vs Leverage**: No outlying values in the upper-right or lower-right corners (outliers are not influential to the regression line and their removal wont play a role). Most values are beyond Cook's distance, so all cases influence the fitted values. It seems that value 18 is now found to be very influential, however we are not going to experiment with its removal - we have only few data points, and removing this insight into linear relationship of the predictors and response might make our future predictions for similar to n.18 points less accurate.

```
vif_final <- vif(model_final)
barplot(vif_final,
        main="VIF Values (Final Model)",
        horiz=TRUE,
        col="lightblue",
        las=2)
abline(v=5, lwd=3, lty=2, col="orange")
abline(v=10, lwd=3, lty=2, col="darkred")
```



In our final model, VIF values for each of the variables is lower than 5, meaning that the variables do not show collinearity with other predictors. To make sure that there is no multicollinearity in the final model, I will run the multicollinearity diagnostics tests again:

```r
#Overall Multicollinearity Diagnostics Measures
omcdiag(model_final)
```

```
##
## Call:
## omcdiag(mod = model_final)
##
##
## Overall Multicollinearity Diagnostics
##
##                        MC Results detection
## Determinant |X'X|:         0.0748         0
## Farrar Chi-Square:       111.9201         1
## Red Indicator:             0.4498         0
## Sum of Lambda Inverse:    13.0435         0
## Theil's Method:           -0.9448         0
## Condition Number:         91.0293         1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

Most of the tests did not detect multicollinearity, and that is supported by our VIF values. However, I would like to test the varaibles as well to make sure of that:

```r
#Individual Multicollinearity Diagnostic Measures
#locate where mc is
imcdiag(model_final)
```

```
##
## Call:
## imcdiag(mod = model_final)
##
##
## All Individual Multicollinearity Diagnostics Result
##
##         VIF    TOL      Wi      Fi Leamer   CVIF Klein   IND1   IND2
## M    2.0002 0.4999  8.2020 10.5026 0.7071 2.7746     0 0.0610 1.0402
## Ed   2.8629 0.3493 15.2757 19.5604 0.5910 3.9712     0 0.0426 1.3535
## Po1  1.9081 0.5241  7.4466  9.5353 0.7239 2.6468     0 0.0639 0.9900
## U2   1.3631 0.7336  2.9772  3.8123 0.8565 1.8908     0 0.0895 0.5541
## Ineq 3.5304 0.2833 20.7495 26.5695 0.5322 4.8971     0 0.0345 1.4909
## Prob 1.3787 0.7253  3.1055  3.9765 0.8517 1.9124     0 0.0885 0.5714
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
##
## * all coefficients have significant t-ratios
##
## R-square of y on all x: 0.7659
##
## * use method argument to check which regressors may be the reason of collinearity
## ===================================
```

Test results tell us that all predictors are significant and collinearity is not detected.

For our final linear regression model that explains 76% of the data (R-squared), there are 6 significant predictors:

-M, percentage of males aged 14–24 in total state population -Ed, mean years of schooling of the population aged 25 years or over -Po1, per capita expenditure on police protection in 1960 -U2, unemployment rate of urban males 35–39 -Ineq, income inequality: percentage of families earning below half the median income -Prob, probability of imprisonment: ratio of number of commitments to number of offenses

We have an R-squared of 76, and Adjusted R-Squared of 73. However, it is an estimation based on the training data, and considering that we have few data points (47), **overfitting** might be present in our model. This means that we cannot use the above Crime rate prediction and we need to estimate the true quality of our model.

Step 8: 5-fold Cross Validation

To estimate the quality of the model, I decided to perform 5-fold cross validation. I chose Cross-Validation not only because it is the general practice for linear regression quality estimation, but also because AIC and BIC may sometimes lead us toward choosing an over-complicated or over-simplified model respectively. Moreover, both AIC and BIC are related to cross-validation, but cross-validation does not produce their common problems.

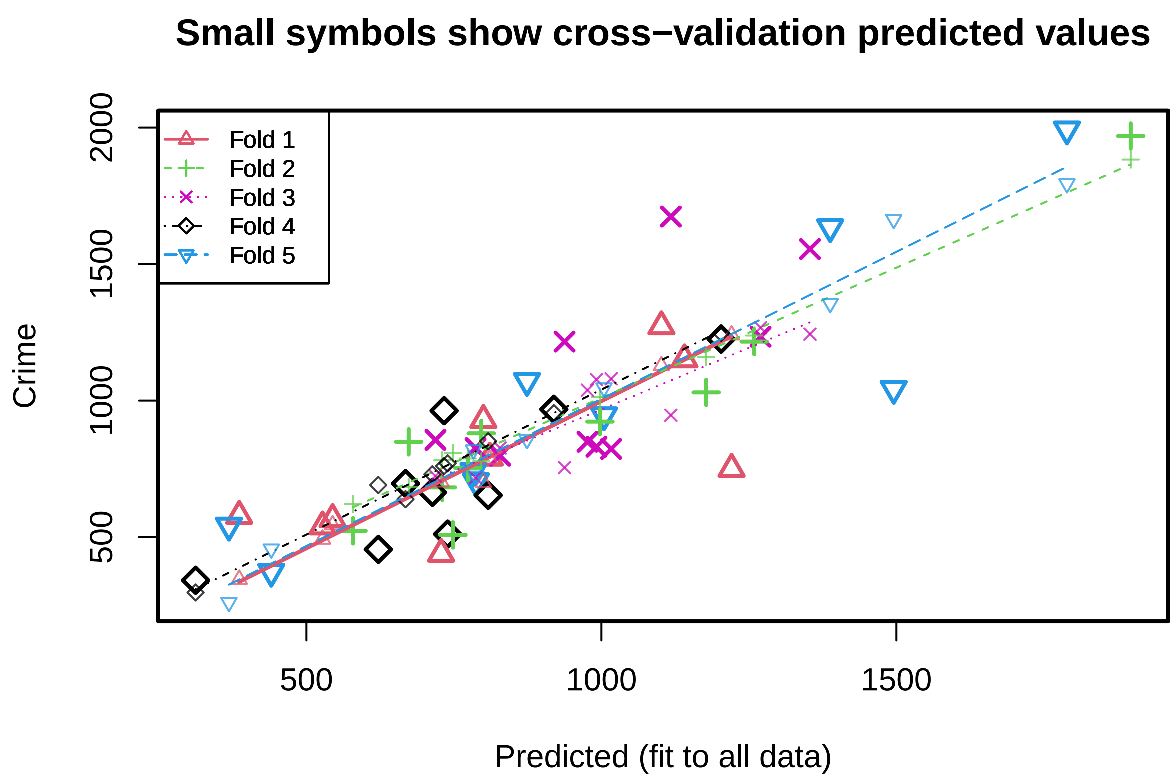
For cross-validation, I chose 5 as the number of folds instead of the commonly used 10 because of a small number of data points in our set (we only have 47 data points). Splitting data into too many folds would mean having 'incomplete' representation of data in each of the folds and having more variability caused by such a small range of each predictor in the fold.

Although the model with our final set of 6 variables was found to be the one with highest significance of parameters, I will run cross-fold validation on the first model with all 15 predictors as well to see how true accuracy compares to the R-squared values that we got on training data.

I will start with the final set of predictors and then go back to the first model.

```
cv_final <- cv.lm(data,
                  model_final,
                  m=5) #m is the number of folds
```

```
## Warning in cv.lm(data, model_final, 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
```

```
cv_final
```

```
##       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
```

We get an overall mean squared prediction error in cross-validated model with 6 significant predictor of 53586.

Let's calculate the new R-squared and adjusted R-squared: - first, we calculate the sum of squared errors (residuals) multiplying the mean squared error by the number of data points - then, we calculate the sum of squares total (SST), the squared differences between the response variable and its mean. - finally, we calculate R-squared using the formula (1-SSEresiduals/SST)

```r
#sum of squared errors (residuals)
sseres_cv_final <- nrow(data)*attr(cv_final,"ms")
sseres_cv_final
```

```
## [1] 2518546
```

```r
#sum of squares total
sst <- sum((data$Crime - mean(data$Crime))^2)
sst
```

```
## [1] 6880928
```

```r
#R-squared
R2_cv_final <- 1-sseres_cv_final/sst
R2_cv_final
```

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

```r
#adjusted R-squared
R2_adj_cv_final <- 1 - (((1-R2_cv_final)*(nrow(data)-1))/(nrow(data)-6-1)) #6 predictors
R2_adj_cv_final
```

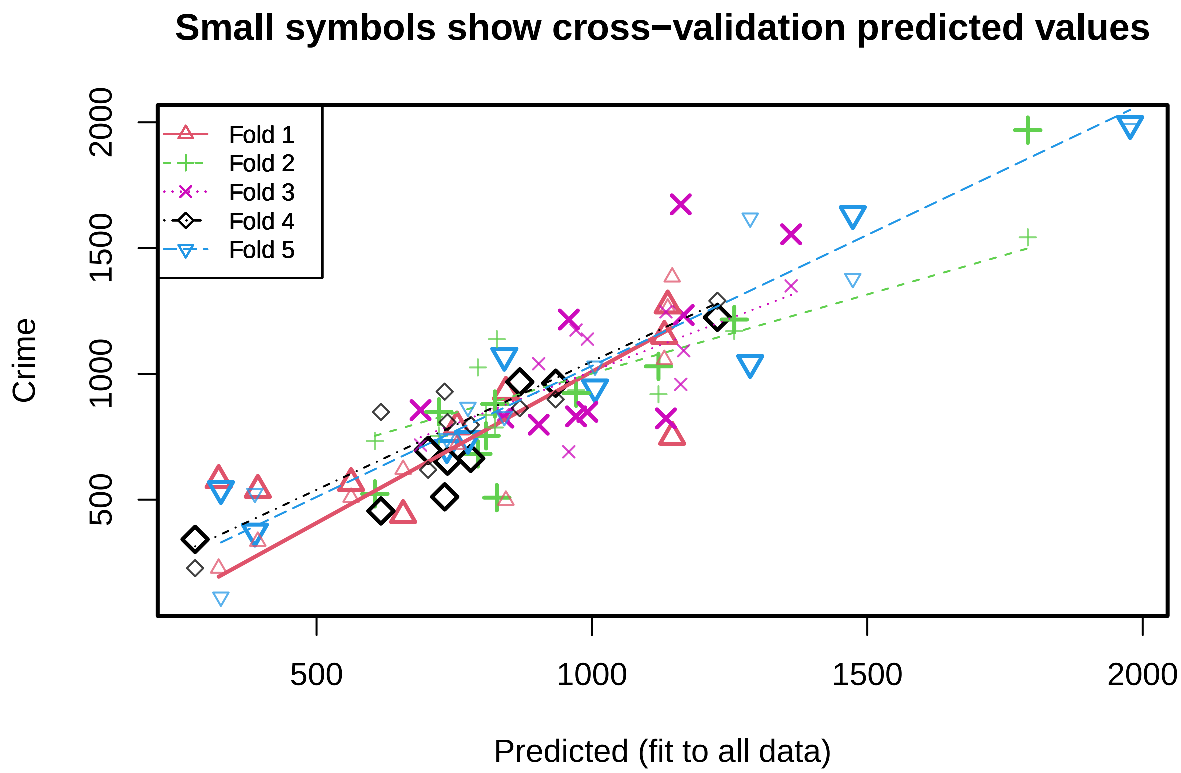
```
## [1] 0.5790789
```

After cross-validation, the R-squared for the model with 6 significant variables decreased from 76.6 to 63.4. This is a sign that our model before cross validation had overfitting, and it shows how important it is to validate even models that seem to make good predictions. **Adjusted R-squared decreased to 57.9 from 73**, which also suggests a lot of overfitting in the initial 6-factor model.

Let's repeat the procedure for the model with all parameters:

```r
cv_model1 <- cv.lm(data,
                  model_1,
                  m=5) #m is the number of folds
```

```
## Warning in cv.lm(data, model_1, 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
cv_model1
```

```
##       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  755.0322  719.4819    1
## 2  25.2999  1635 1473.6764 1379.5108    5
## 3  24.3006   578  322.2615  227.3811    1
```

```
## 4  29.9012  1969 1791.3619 1542.8663    2
## 5  21.2998  1234 1166.6840 1092.1924    3
## 6  20.9995   682  792.9301 1025.6864    2
## 7  20.6993   963  934.1637  898.5349    4
## 8  24.5988  1555 1361.7468 1349.7715    3
## 9  29.4001   856  688.8682  717.0401    3
## 10 19.5994   705  736.5080  743.2757    5
## 11 41.6000  1674 1161.3291  958.3058    3
## 12 34.2984   849  722.0408  752.8461    2
## 13 36.2993   511  732.6412  929.2776    4
## 14 21.5010   664  780.0401  797.4106    4
## 15 22.7008   798  903.3541 1040.2775    3
## 16 26.0991   946 1005.6569 1031.3568    5
## 17 19.1002   539  393.3633  334.2928    1
## 18 18.1996   929  843.8072  497.4904    1
## 19 24.9008   750 1145.7379 1384.9349    1
## 20 26.4010  1225 1227.8387 1290.4074    4
## 21 37.5998   742  774.8506  867.6315    5
## 22 37.0994   439  657.2092  620.1834    1
## 23 25.1989  1216  957.9918  690.2073    3
## 24 17.6000   968  868.9805  863.7702    4
## 25 21.9003   523  605.8824  733.1797    2
## 26 22.1005  1993 1977.3707 1975.1257    5
## 27 28.4999   342  279.4772  227.4408    4
## 28 25.8006  1216 1258.4842 1170.1042    2
## 29 36.7009  1043 1287.3917 1619.8299    5
## 30 28.3011   696  702.6945  618.7241    4
## 31 21.7998   373  388.0334  525.4791    5
## 32 30.9014   754  807.8167  836.6094    2
## 33 25.5005  1072  840.9992  830.6871    5
## 34 21.6997   923  971.4558  934.6280    2
## 35 37.4011   653  737.7888  808.0845    4
## 36 44.0004  1272 1137.6171 1261.6160    1
## 37 31.6995   831  971.1513 1174.2195    3
## 38 16.6999   566  562.6934  509.0826    1
## 39 27.3004   826  839.2864  838.1895    3
## 40 29.3004  1151 1131.4533 1057.0870    1
## 41 30.0001   880  823.7419  786.7404    2
## 42 12.1996   542  326.3324  112.9800    5
## 43 31.9989   823 1134.4172 1246.7022    3
## 44 30.0001  1030 1120.8227  919.1066    2
## 45 32.5996   455  616.8983  848.6350    4
## 46 16.6999   508  827.3543 1137.6778    2
## 47 16.0997   849  991.7629 1138.2873    3
```

The mean squared prediction error in cross-validated model with all predictors is significantly higher compared to the model with only significant parameters - 84948 instead of 53586.

Let's see how the R-squared and adjusted R-square compare to our prevous results:

```r
#sum of squared errors (residuals)
sseres_cv_model1 <- nrow(data)*attr(cv_model1,"ms")
sseres_cv_model1
```

```
## [1] 3992597
```

```r
#R-squared
R2_cv_model1 <- 1-sseres_cv_model1/sst
R2_cv_model1
```

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

```r
#adjusted R-squared
R2_adj_cv_model1 <- 1 - (((1-R2_cv_model1)*(nrow(data)-1))/(nrow(data)-6-1))
R2_adj_cv_model1
```

```
## [1] 0.3327228
```

There is a drastic difference compared to the non-validated model: R-squared reduced from 80.3 to 41.9, and Adjusted R-squared from 70.8 to 33.3. It shows that the initial model had a lot of overfitting, which is probably a reason why the initial R-squared was 80, higher than the same value for a 6-parameter model (76.6) - the model was fitting not only significant values, but also a lot of randomness.

This once again shows how important it is to: a) Build a linear regression model with significant parameters to give the model more power for prediction b) Validate a model after fitting it on a training data set to reduce overfitting and estimate its real quality.