Predicting used car prices with linear regression

Glenn Bruns

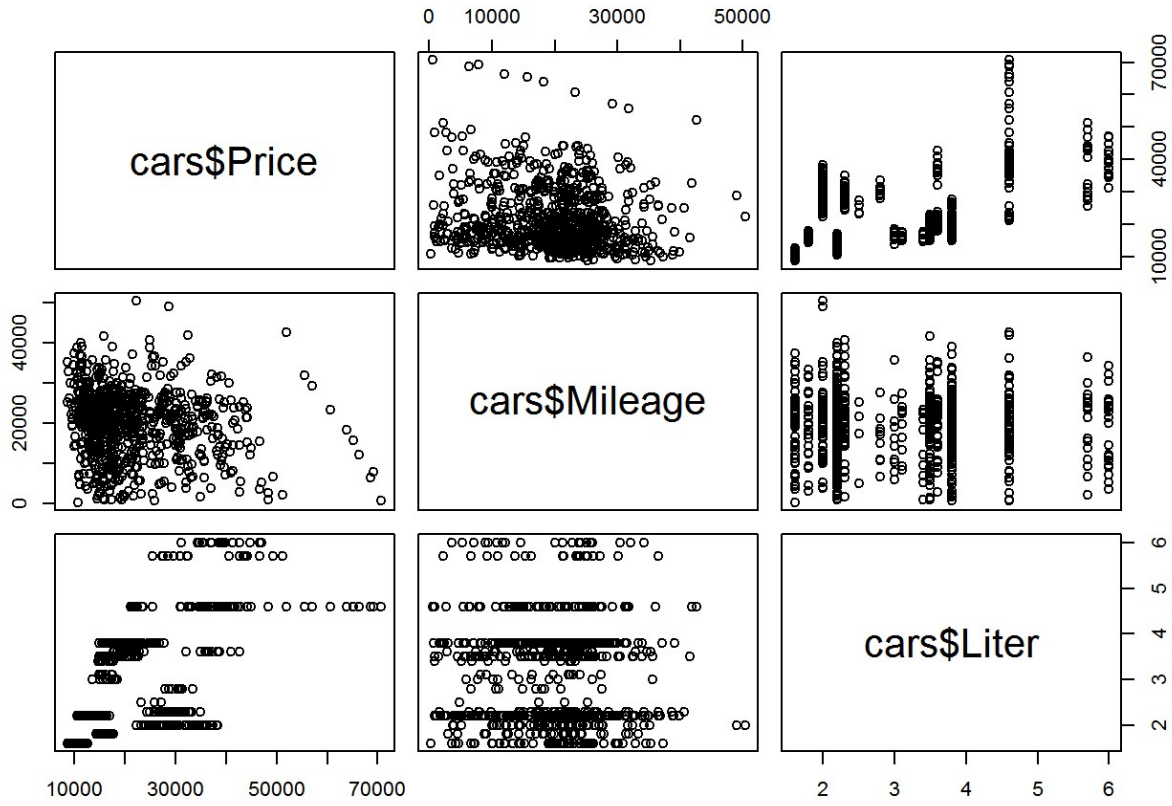
April 3, 2016

```
knitr::opts_chunk$set(prompt=TRUE, comment="", echo=FALSE)
```

We will use the Kuipers 2008 used car data to try to predict the price of a used car from features like mileage, engine size, and trim.

Data exploration

In a real project to predict used car prices, we would spend substantial effort on pre-processing and exploring our data. Here we show only scatter plots of a few features.



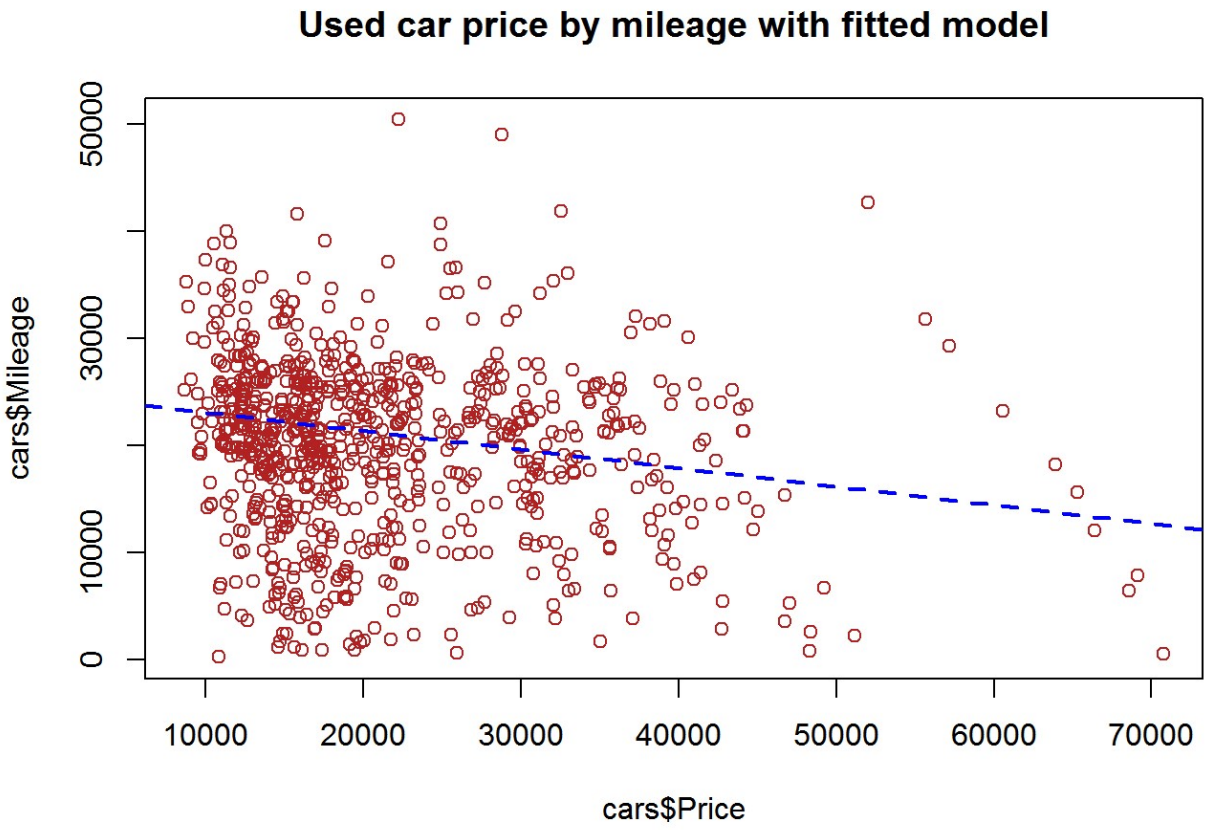
Simple linear regression to predict Price from Mileage

Let's try predicting price from a car's mileage, which means the number of miles the car has been

driven.

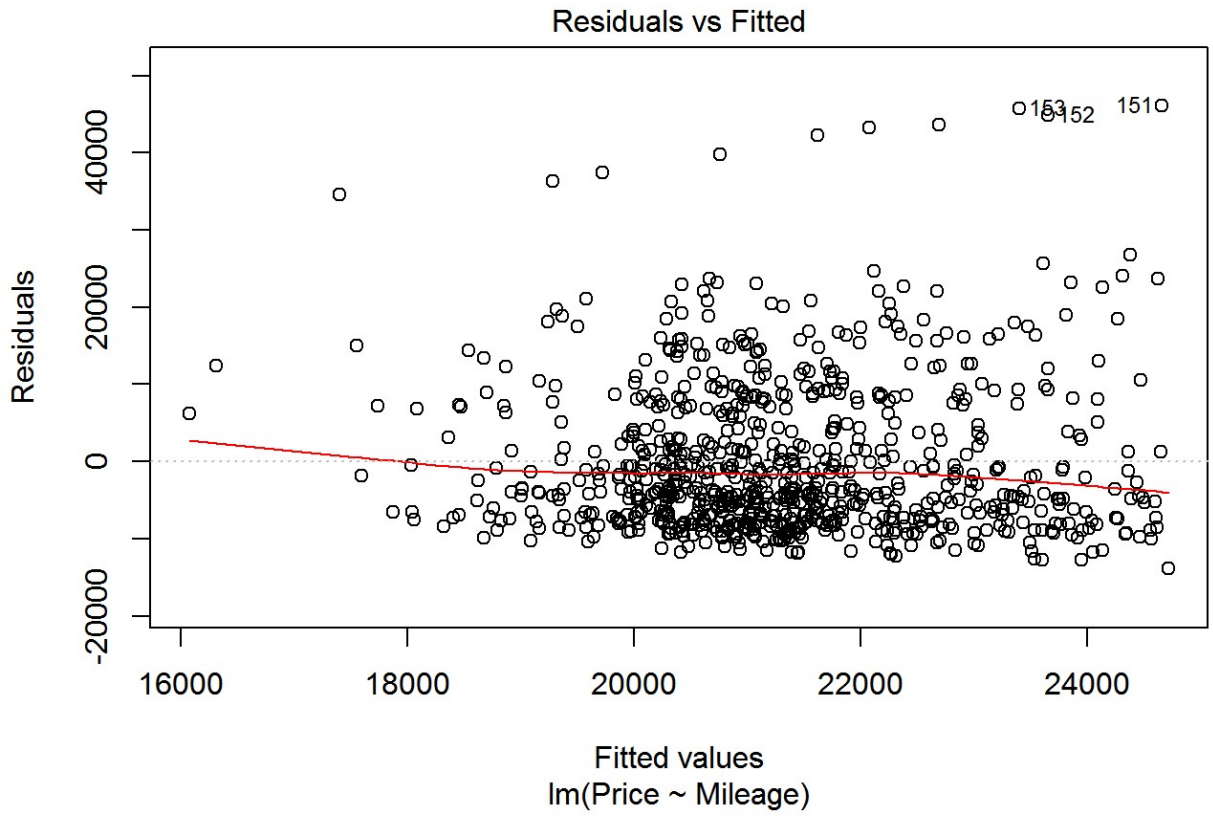
The fit shows decreasing price with increasing mileage, as we expect. However, the scatterplot shows we can't expect too much from using mileage along.

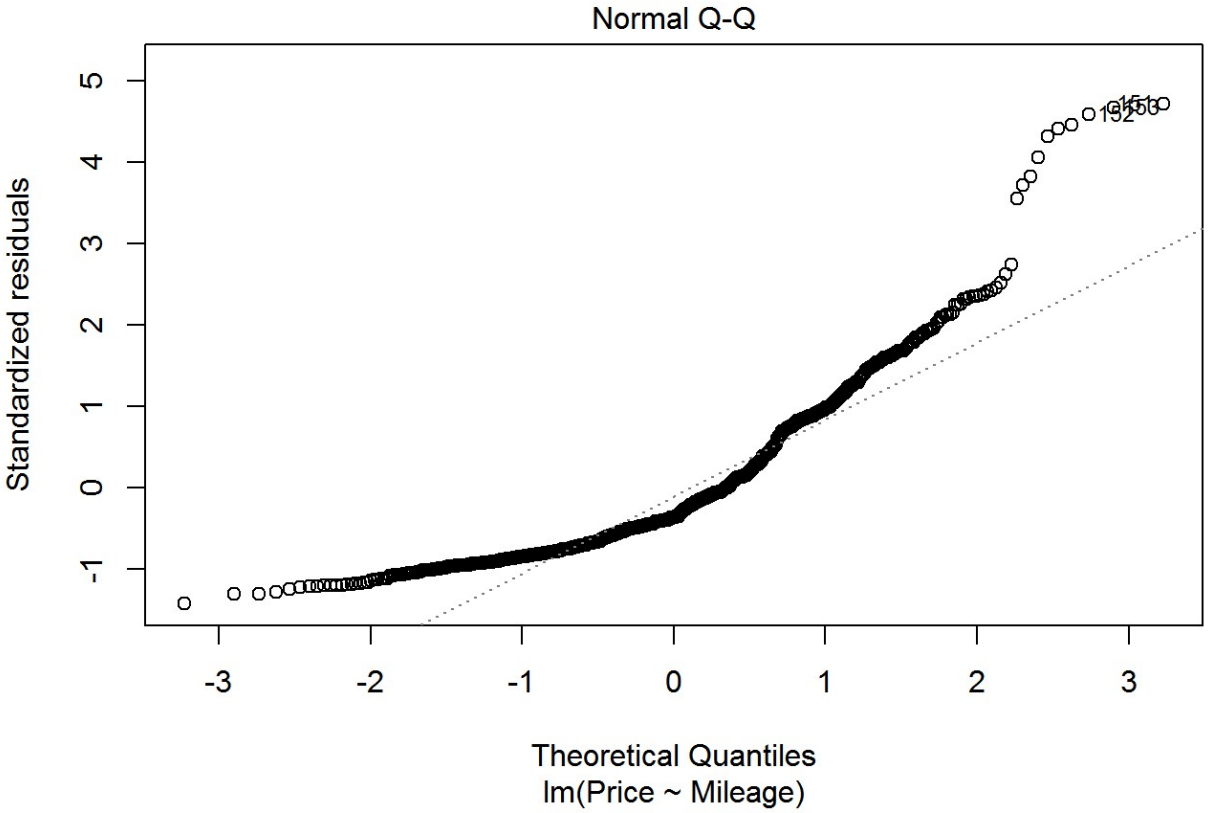
```
Warning in lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...):
extra arguments c("'ylab'", "'xlab'") are disregarded.
```

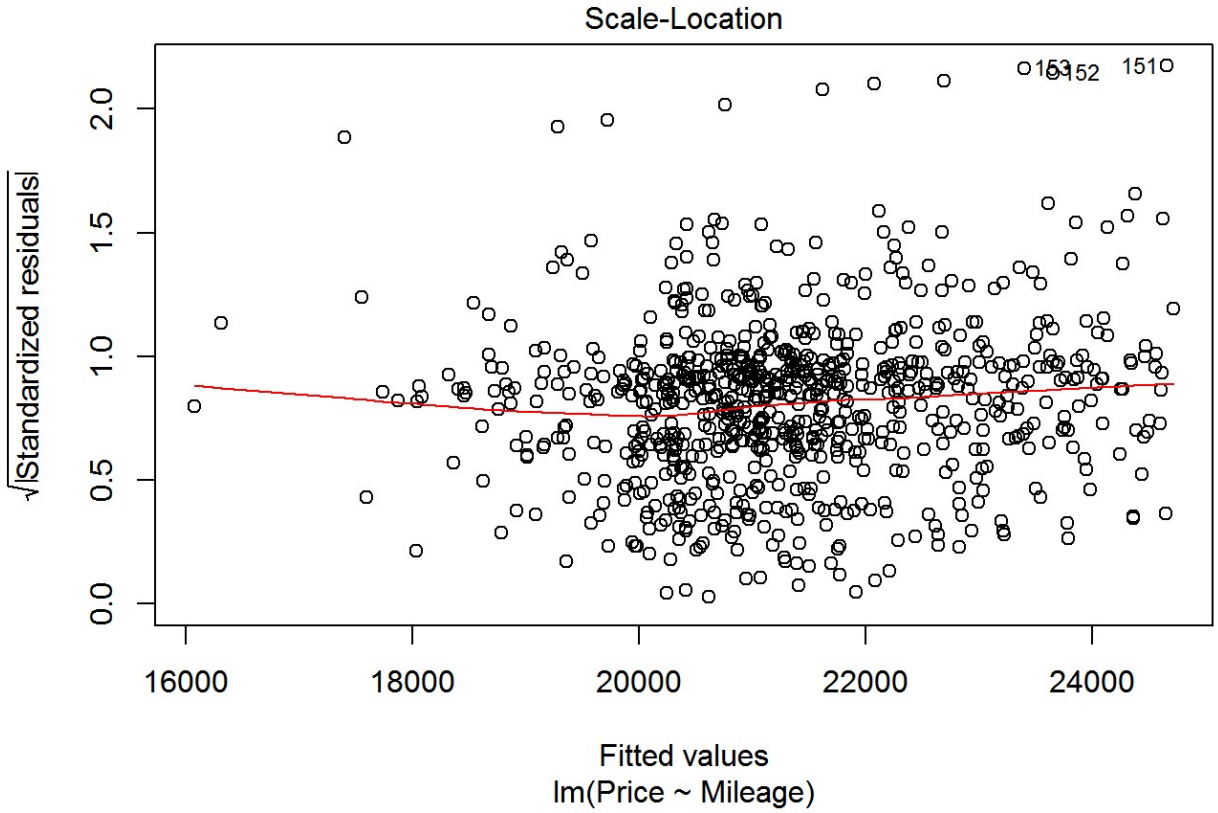


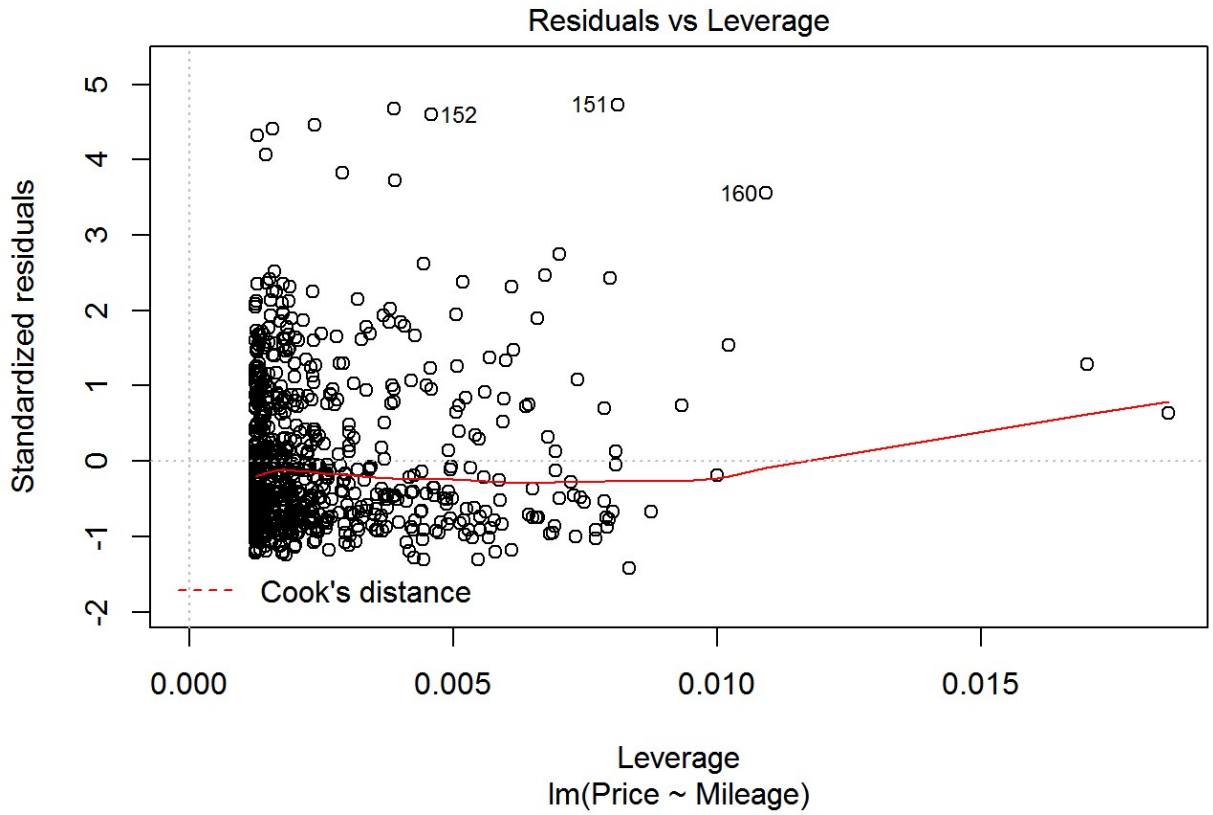
Summary information from the model fit shows large error and a small R-squared statistic.

```
Call:
lm(formula = Price ~ Mileage, data = cars, ylab = "Price", xlab = "Mileage")

Residuals:
   Min     1Q Median     3Q    Max
-13905  -7254  -3520   5188  46091

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.476e+04  9.044e+02  27.383  < 2e-16 ***
Mileage     -1.725e-01  4.215e-02  -4.093 4.68e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9789 on 802 degrees of freedom
Multiple R-squared:  0.02046,   Adjusted R-squared:  0.01924
F-statistic: 16.75 on 1 and 802 DF,  p-value: 4.685e-05
```




Normal Q-Q
Standardized residuals
5
4
3
2
1
0
-1
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Price ~ Mileage)


Scale-Location
153
152
151
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
16000
18000
20000
22000
24000
Fitted values
lm(Price ~ Mileage)

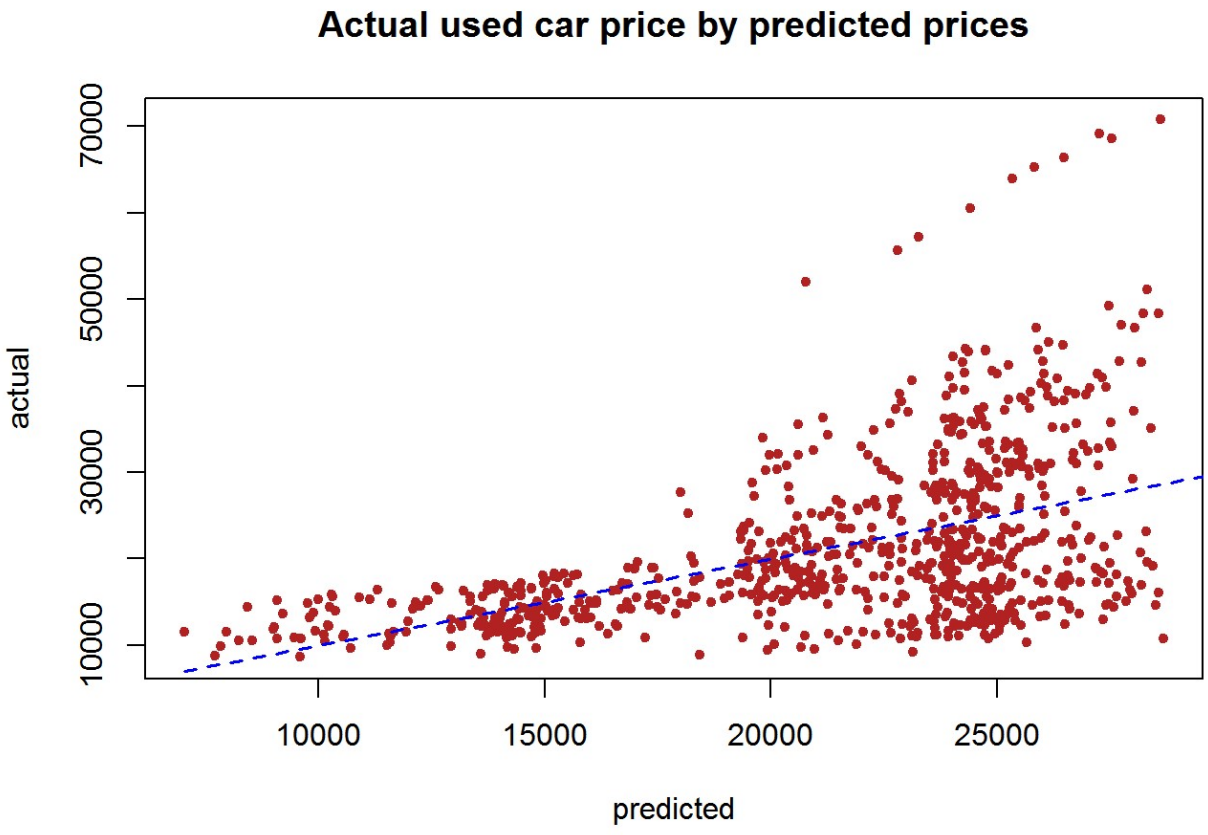


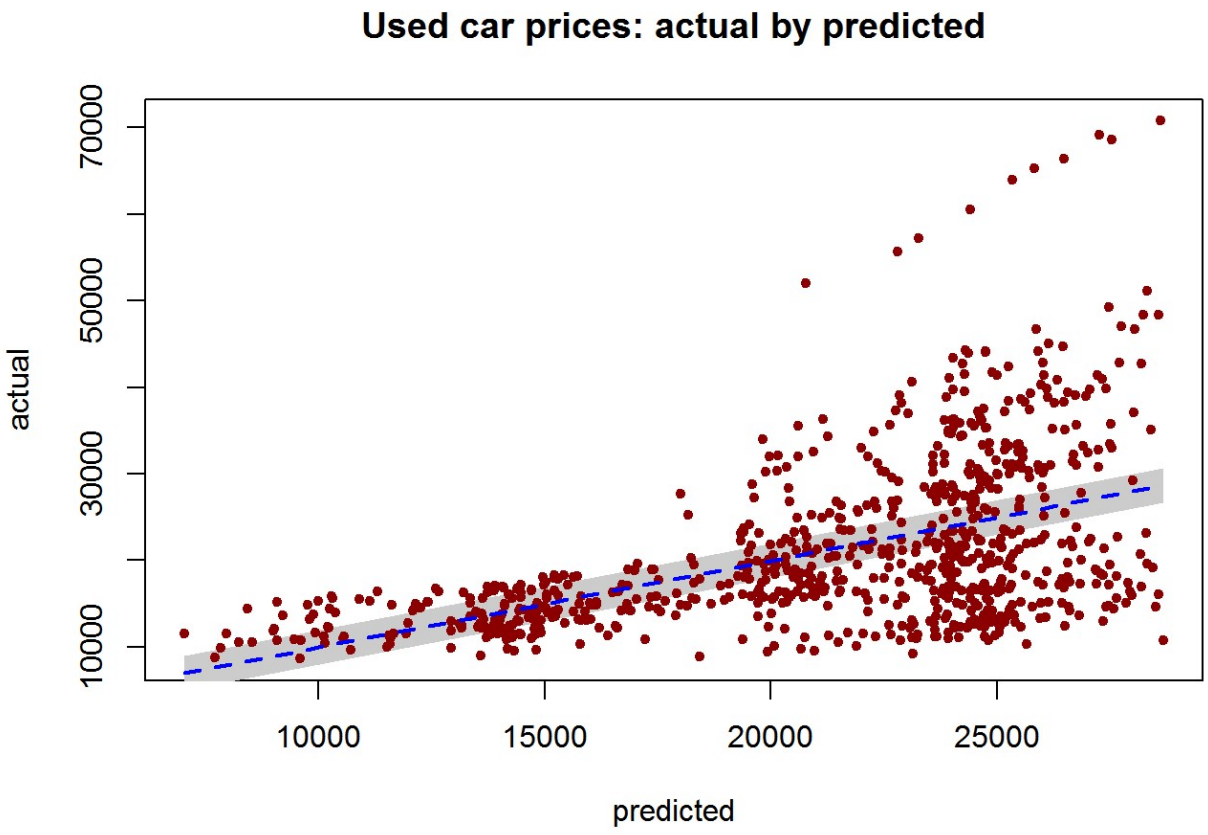
Predicting price from Mileage, Cruise, and Leather

We expand the model by incorporating whether a car has a leather interior and cruise control.

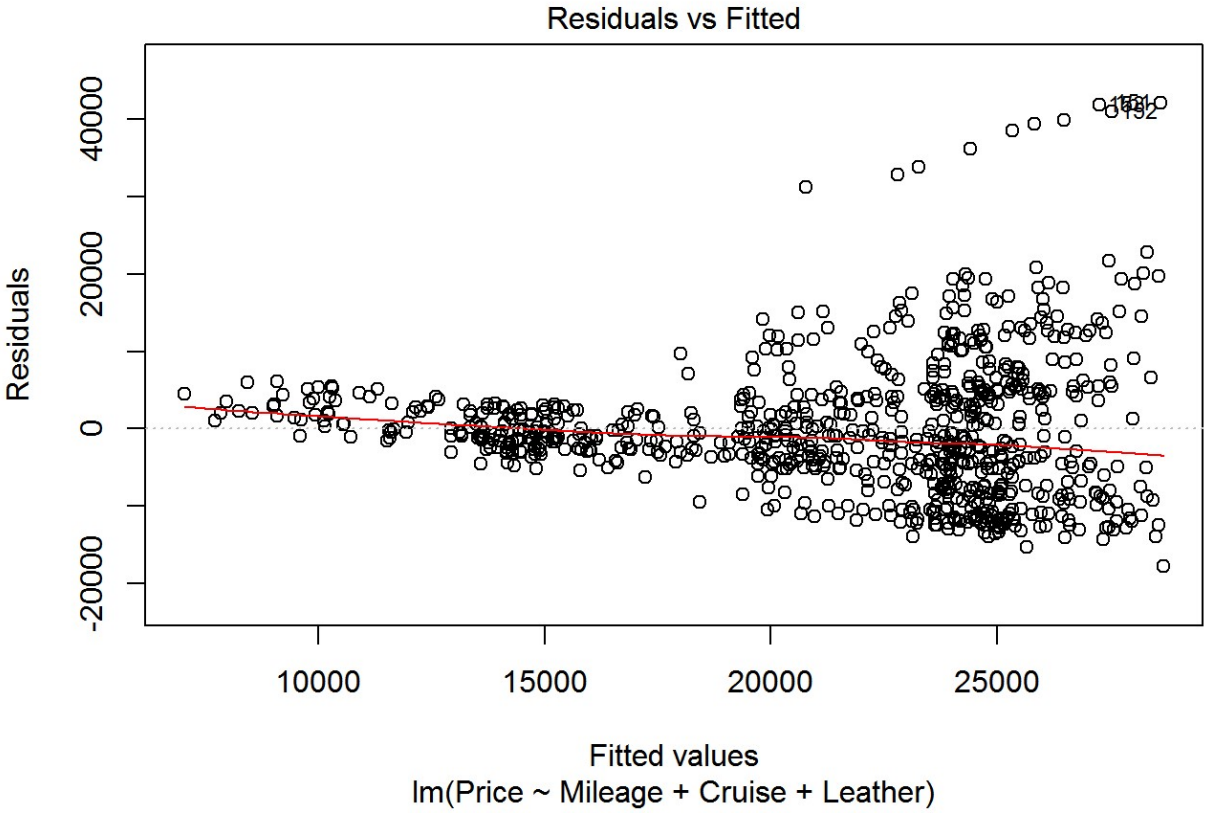
```
Call:
lm(formula = Price ~ Mileage + Cruise + Leather, data = cars)

Residuals:
   Min     1Q Median     3Q    Max
-17866  -5011   -972   4018  42135

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.430e+04  1.083e+03  13.206  < 2e-16 ***
Mileage     -1.863e-01  3.706e-02  -5.026 6.19e-07 ***
Cruise       1.026e+04  7.053e+02  14.542  < 2e-16 ***
Leather      4.176e+03  6.806e+02   6.135 1.34e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8606 on 800 degrees of freedom
Multiple R-squared:  0.2448,    Adjusted R-squared:  0.242
F-statistic: 86.46 on 3 and 800 DF,  p-value: < 2.2e-16
```

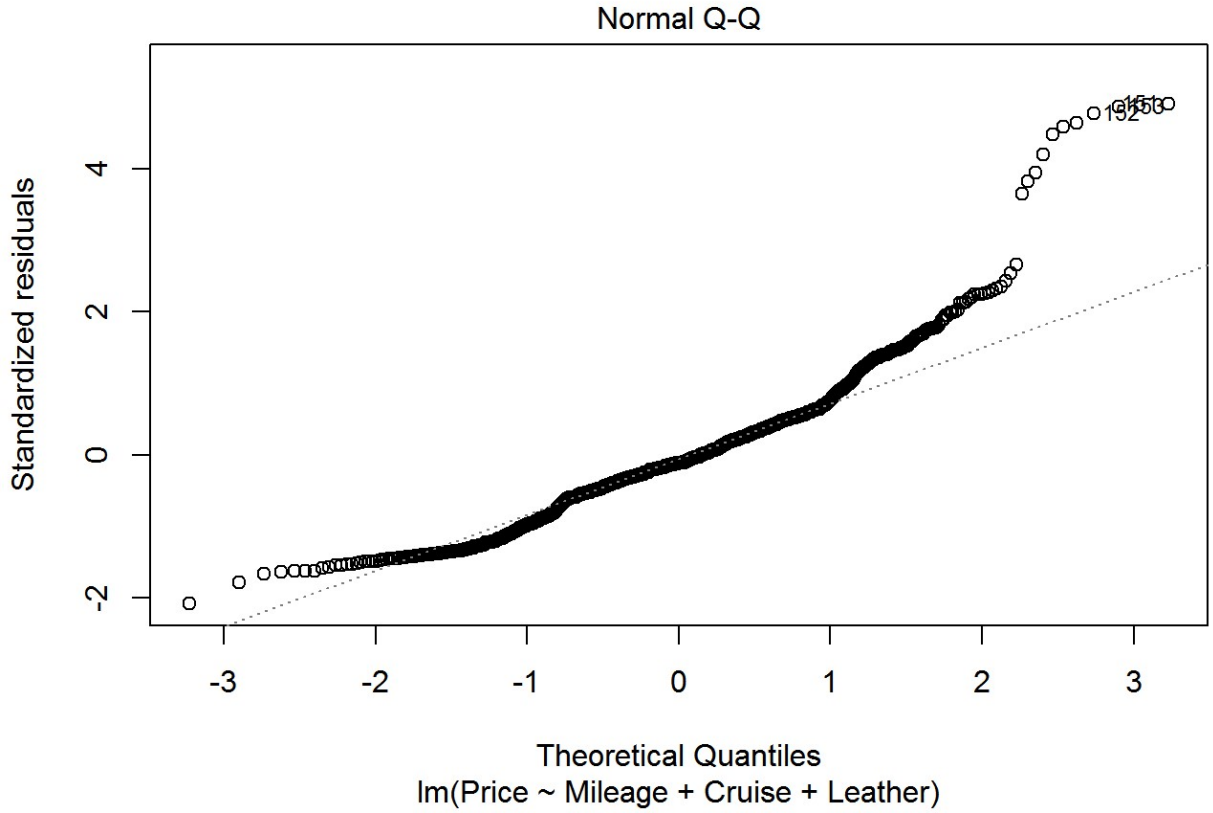
Plots of predicted vs. actual prices help in showing how the model does overall, and also whether the error is constant or not. In these plots we see that error tends to be bigger when the price is higher.

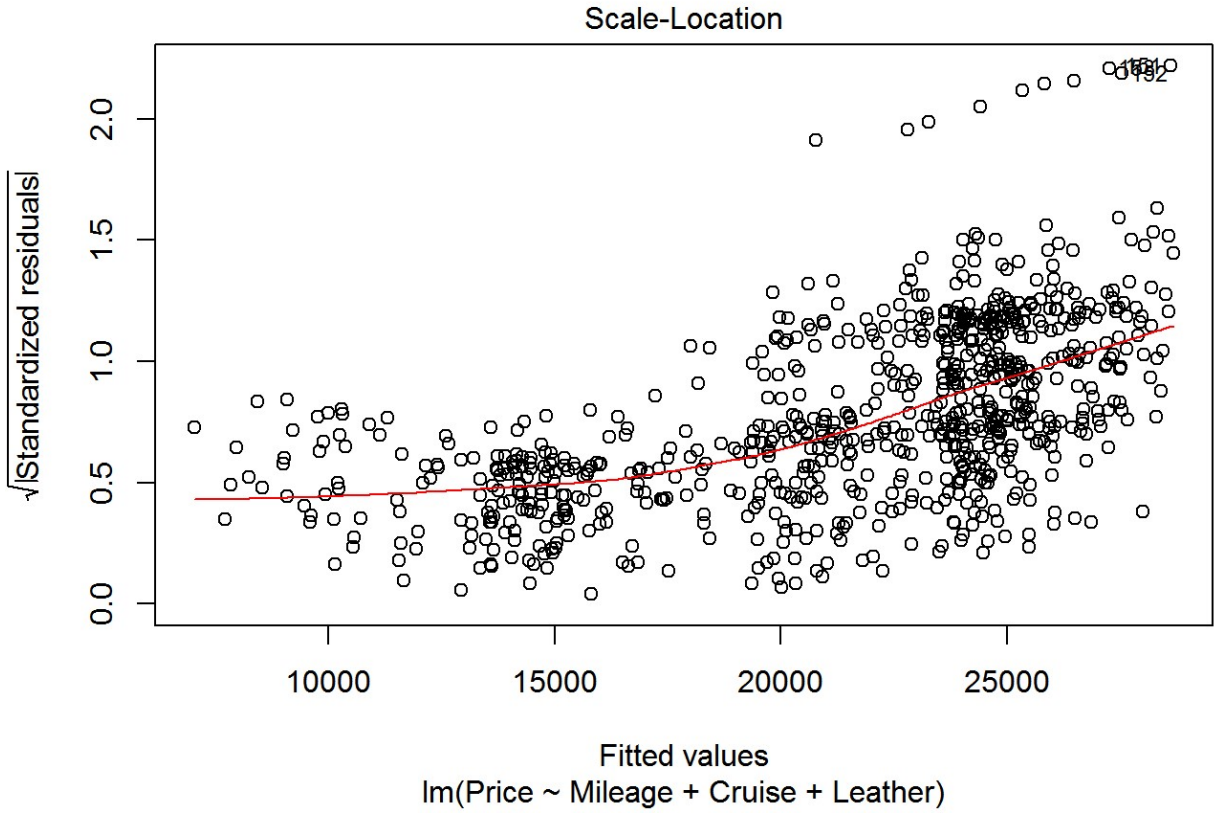


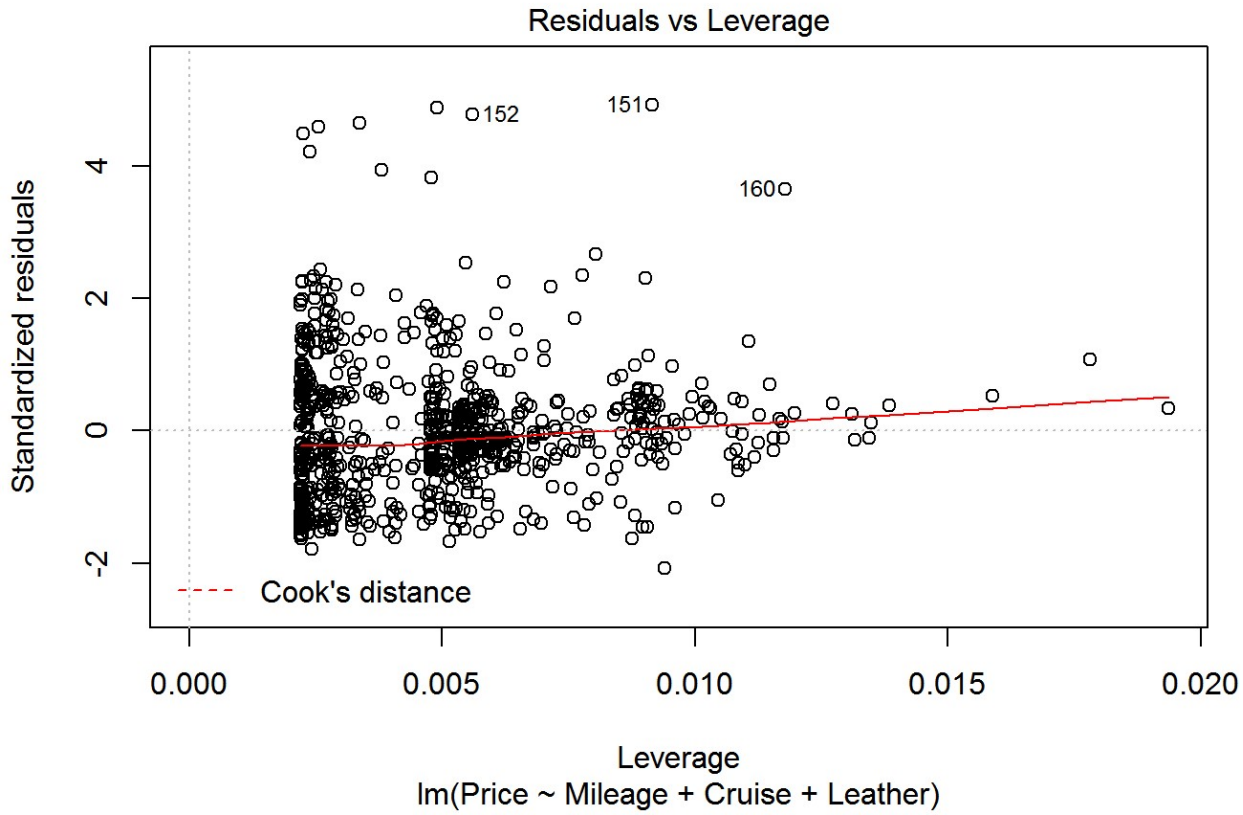


The Residuals vs Fitted plot is another way of seeing how errors increase with price. This tells us our linear model is not very good yet. The Q-Q plot shows that the distrbution of errors is not too far from a normal distribution.


Residuals vs Fitted
Residuals
40000
20000
0
-20000
10000
15000
20000
25000
Fitted values
lm(Price ~ Mileage + Cruise + Leather)

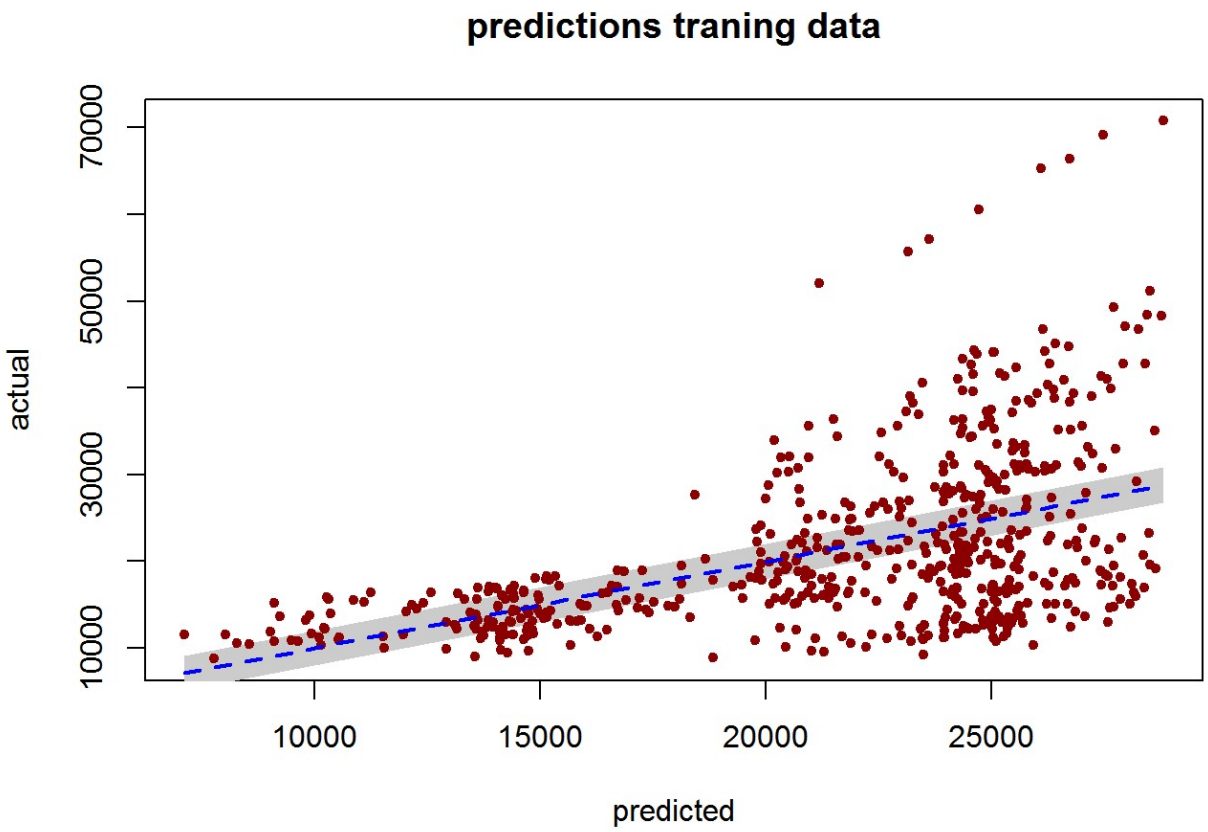

Normal Q-Q
Standardized residuals
4
2
0
-2
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Price ~ Mileage + Cruise + Leather)

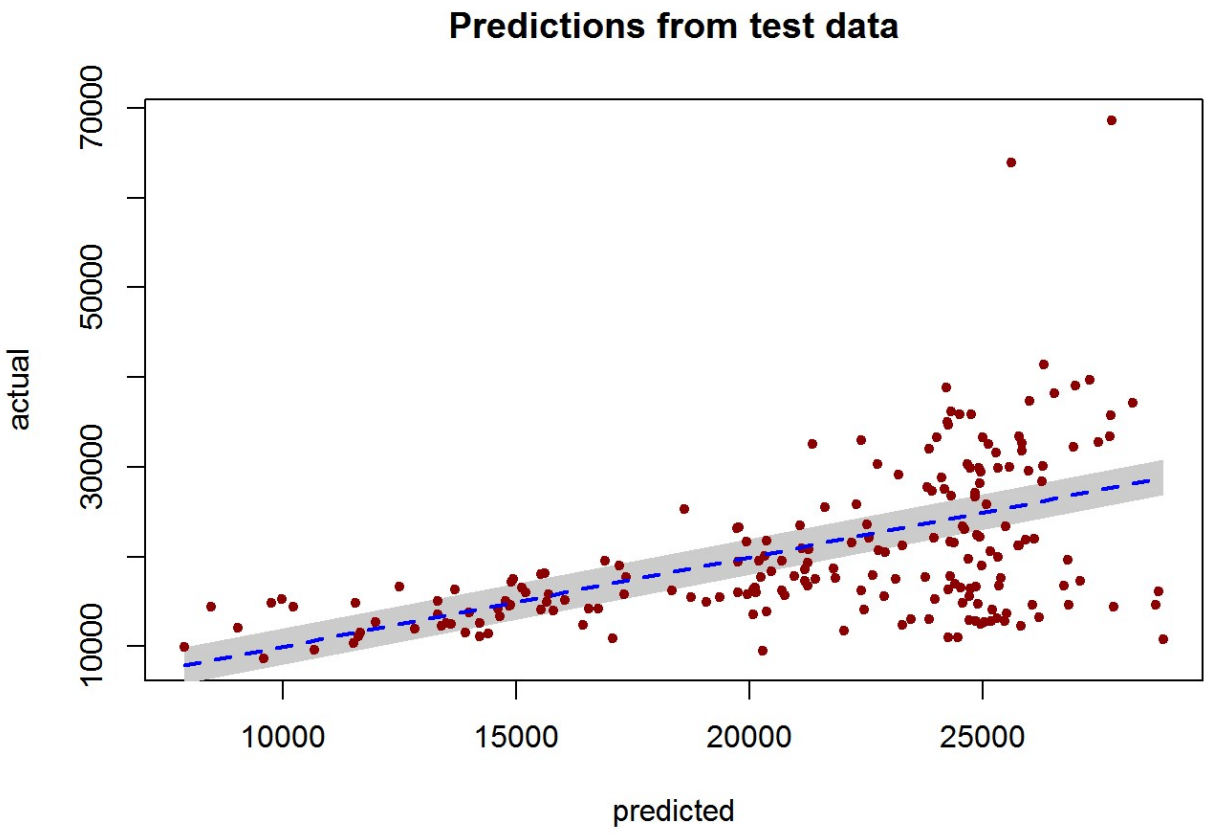

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
10000
15000
20000
25000
Fitted values
lm(Price ~ Mileage + Cruise + Leather)



Predictions from training data and test data

Looking at prediction errors using the training data is not very useful. A model that has low error on training data could still have high error on test data. In this case we see the error on the test data is similar to the error on the training data. This suggests we're not overfitting when we build our model.


predictions traning data
actual
predicted
10000
30000
50000
70000
10000
15000
20000
25000



The RMSE value computed from our test set confirms that our model is not yet very useful.

```
[1] "RMSE for Price prediction from Mileage, Cruise, Leather, on test data: 83
5"
```

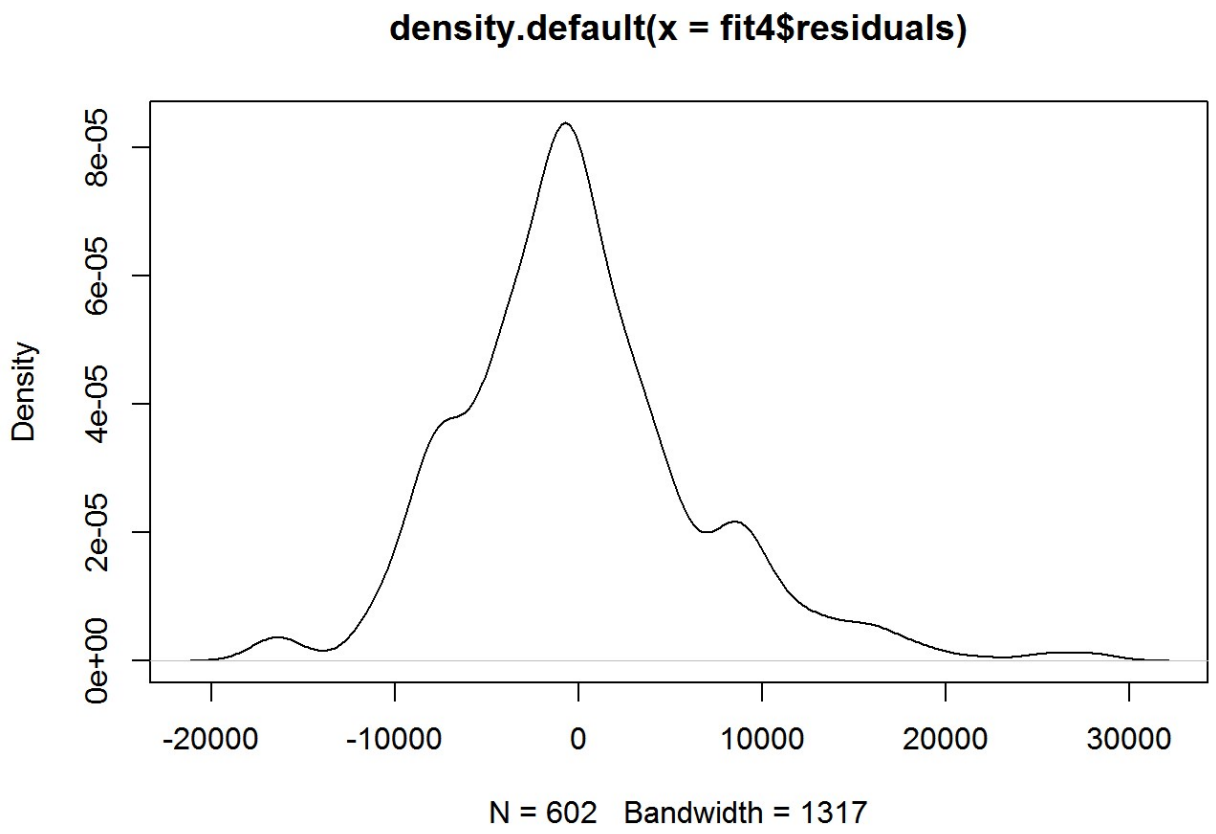
Adding input feature cyl8 to the linear model

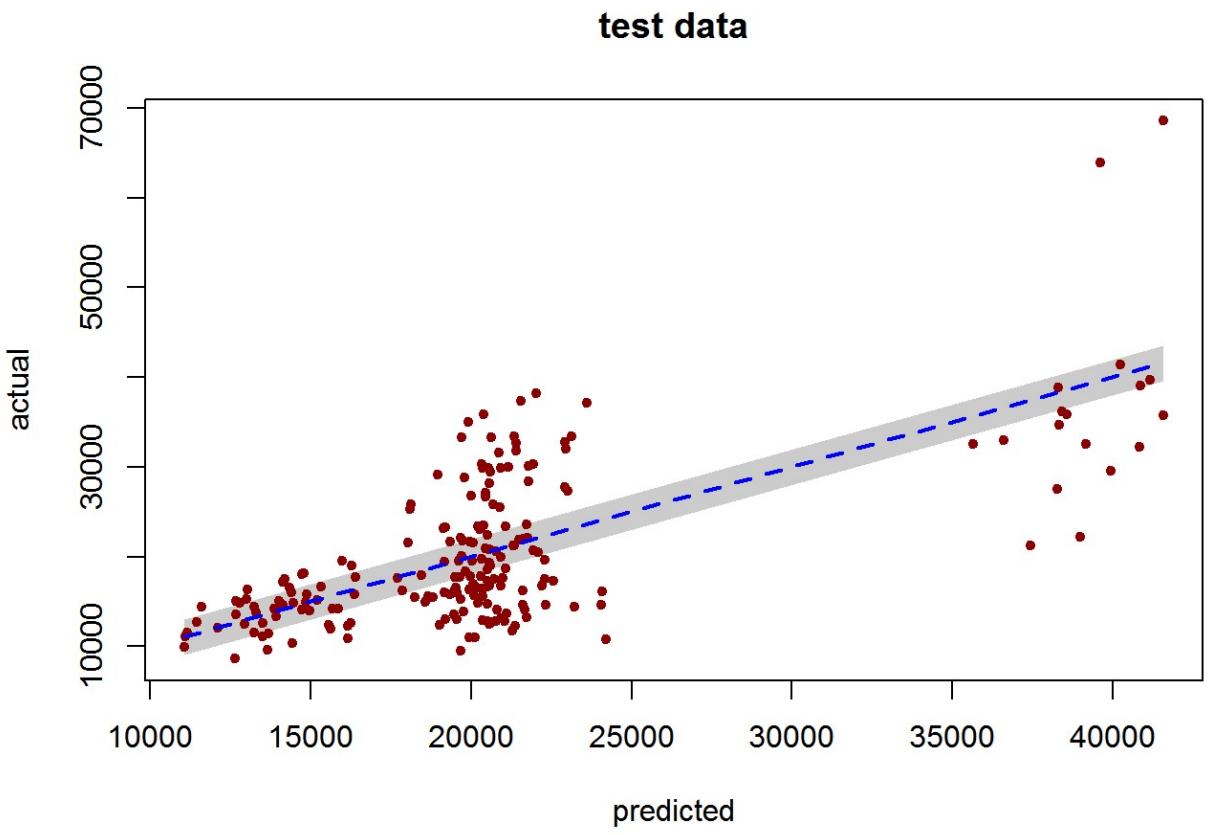
We add to our model a feature that indicates whether a car has an eight-cylinder engine.

The R-squared statistic is improving. It looks like leather isn't very useful for predicting price (at least, in the presence of the other features we're using).

```
Call:
lm(formula = Price ~ Mileage + Cruise + Leather + cyl8, data = tr_dat)

Residuals:
     Min       1Q   Median       3Q      Max
-17242.3  -4057.3   -593.3   2997.1  28203.6

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.686e+04  9.818e+02  17.169  < 2e-16 ***
Mileage     -1.673e-01  3.279e-02  -5.102 4.53e-07 ***
Cruise       6.974e+03  6.475e+02  10.769  < 2e-16 ***
Leather      4.150e+02  6.344e+02   0.654    0.513
cyl8         1.841e+04  8.362e+02  22.010  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6605 on 597 degrees of freedom
Multiple R-squared:  0.5824,    Adjusted R-squared:  0.5796
F-statistic: 208.2 on 4 and 597 DF,  p-value: < 2.2e-16
```

The distribution of the residuals is improving, but it is still not very much like a normal distribution.





Using a validation data set

A problem is that we can't use the test data to pick the "best" set of features, because then we can't use the test data to give a fair idea of the how our model will do on future data.

We also can't use the training data for this purpose, as we already know.

An alternative is to split our data set into *three* pieces, one for training, another (called the "validation" data set) for comparing alternative sets of features, and the last for doing the final test of our model.

A problem with splitting the data three ways is that the size of each of the three pieces is not very big. This means our validation results are not very reliable.

```
Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length
```

```
Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
```

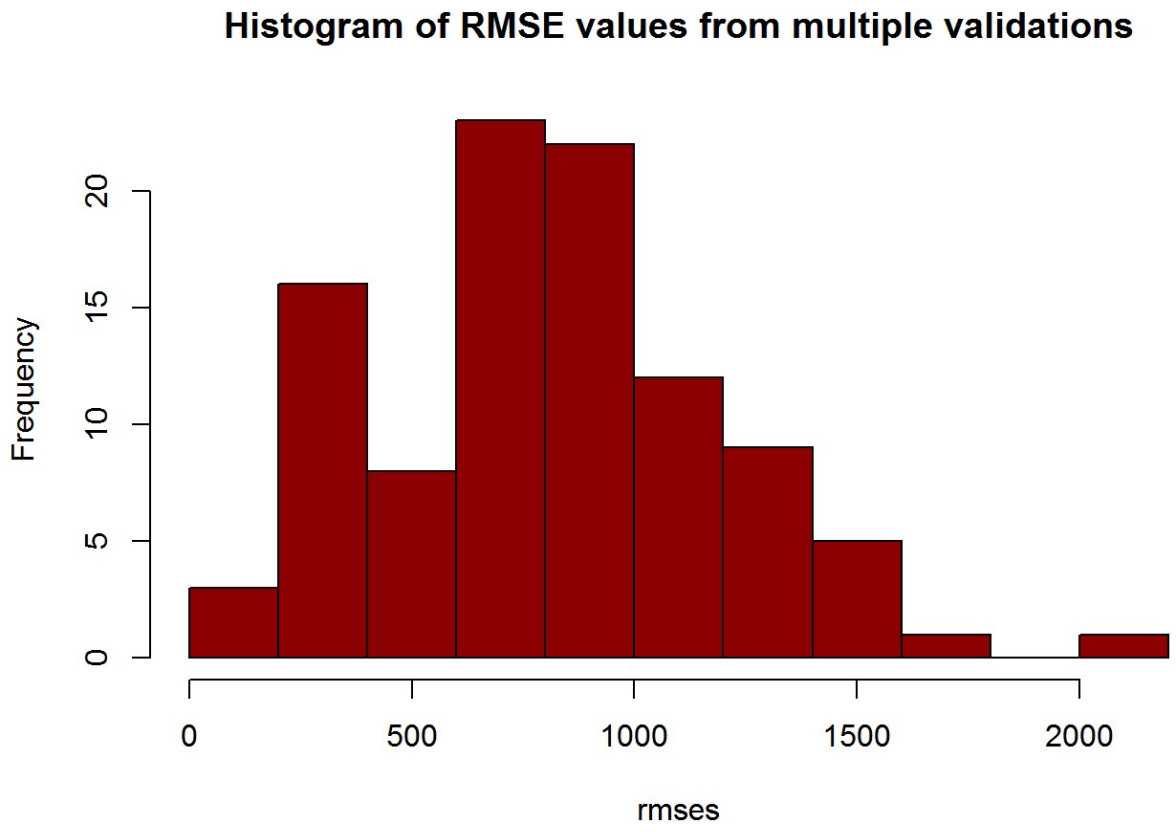
```
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length
```

```
Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length
```

```
Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
```

```
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length
```

```
Warning in actual - predicted: longer object length is not a multiple of
shorter object length

Warning in actual - predicted: longer object length is not a multiple of
shorter object length
```



Finding best features using cross validation

A way to get more reliable validation results is to “cross validate”, which basically allows the training data to be re-used. One cross validation method is called 10-fold cross validation. In this method you split the training data into 10 pieces. To see how well a set of features works, you train the model on 9 of the 10 pieces, and validate on the remaining piece. You then repeat this 9 more times, each time using a different subset of 9 of the 10 pieces. In each of the 10 validation runs, you compute a result, like the RMSE.

```
[1] "RMSE from cross validation: 8684"
```

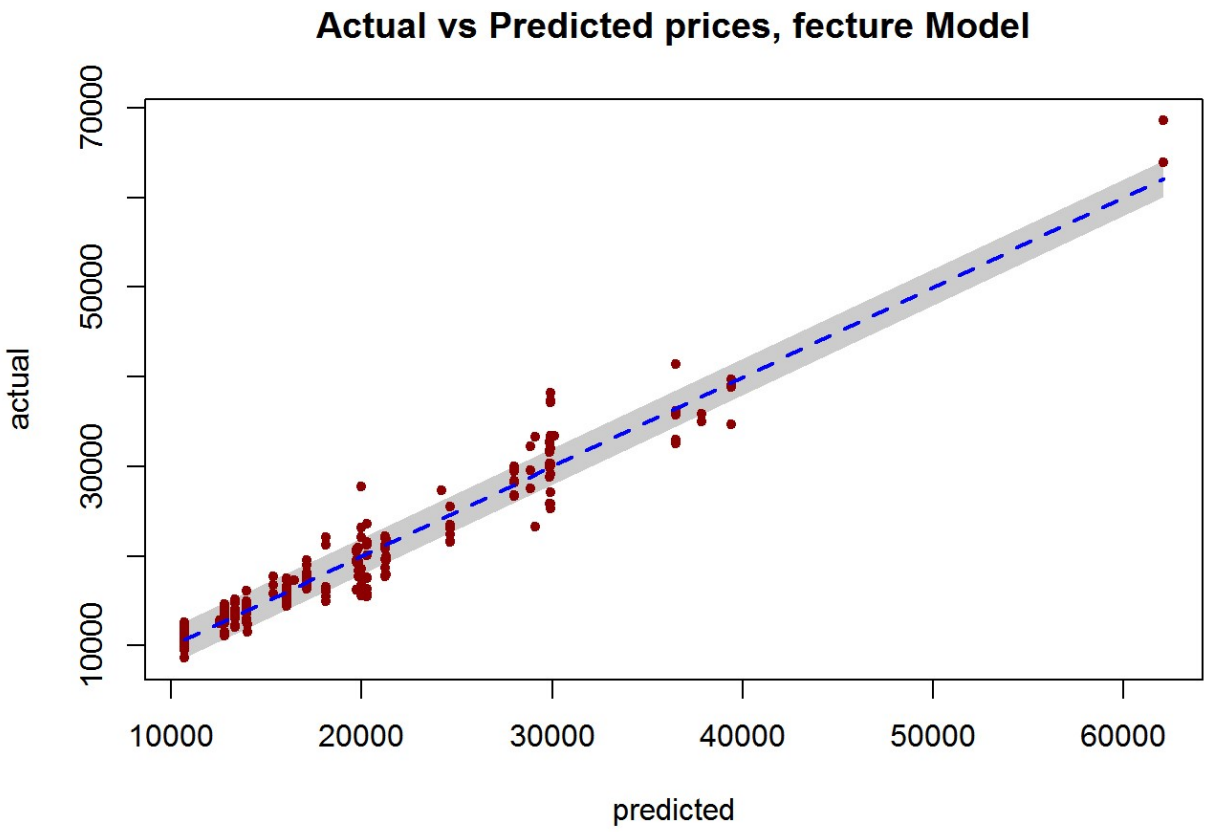
Notes on cross validation:

- Cross validation is used for all kinds of machine learning problems, not just feature selection in linear regression.

- Please read the cross_validate_lm code and understand how it works.
- There are special methods that allow for very cheap cross validation with linear regression. We're using a general-purpose cross validation method here so that you can see the idea.

By computing RMSE for each single feature (using cross validation), we can find the single most effective feature in predicting price.

```
[1] "best feature to predict Price: 'Model'; RMSE = 2386"
```



```
Call:
lm(formula = ff, data = tr_dat)

Residuals:
     Min       1Q   Median       3Q      Max
-10111.5  -1237.3     -4.8   1177.1   8642.0

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               16053.65    1031.93  15.557  < 2e-16 ***
ModelBuick Lacrosse        5239.12    1138.58   4.601 5.18e-06 ***
ModelBuick Lesabre         3757.24    1191.57   3.153 0.001700 **
ModelBuick Park Avenue     8595.65    1228.24   6.998 7.30e-12 ***
ModelCadillac CST-V       28969.28    1263.85  22.922  < 2e-16 ***
ModelCadillac CTS         14072.30    1287.04  10.934  < 2e-16 ***
ModelCadillac Deville     20467.74    1138.58  17.977  < 2e-16 ***
ModelCadillac STS-V6      21805.85    1315.45  16.577  < 2e-16 ***
ModelCadillac STS-V8      26930.80    1263.85  21.309  < 2e-16 ***
ModelCadillac XLR-V8      46059.86    1315.45  35.014  < 2e-16 ***
ModelChevrolet AVEO       -5375.62    1088.98  -4.936 1.05e-06 ***
ModelChevrolet Cavalier   -3272.36    1083.30  -3.021 0.002635 **
ModelChevrolet Classic    -2051.38    1315.45  -1.559 0.119446
ModelChevrolet Cobalt     -2097.96    1107.35  -1.895 0.058654 .
ModelChevrolet Corvette   23367.87    1182.22  19.766  < 2e-16 ***
ModelChevrolet Impala      3918.74    1143.19   3.428 0.000652 ***
ModelChevrolet Malibu      1057.33    1086.56   0.973 0.330920
ModelChevrolet Monte Carlo 4238.20    1148.22   3.691 0.000245 ***
ModelPontiac Bonneville    5157.71    1143.19   4.512 7.81e-06 ***
ModelPontiac G6            3677.45    1202.16   3.059 0.002325 **
ModelPontiac Grand Am      -684.25    1173.91  -0.583 0.560202
ModelPontiac Grand Prix    2040.61    1143.19   1.785 0.074792 .
ModelPontiac GTO          12794.70    1351.11   9.470  < 2e-16 ***
ModelPontiac Sunfire      -3524.32    1351.11  -2.608 0.009334 **
ModelPontiac Vibe            -5.92    1153.73  -0.005 0.995908
ModelSAAB 9-2X AWD         8133.96    1685.13   4.827 1.78e-06 ***
ModelSAAB 9_3             13080.68    1166.48  11.214  < 2e-16 ***
ModelSAAB 9_3 HO          13877.59    1120.28  12.388  < 2e-16 ***
ModelSAAB 9_5             13813.66    1148.22  12.030  < 2e-16 ***
ModelSAAB 9_5 HO          11965.84    1214.27   9.854  < 2e-16 ***
ModelSaturn Ion           -2690.99    1096.08  -2.455 0.014382 *
ModelSaturn L Series        411.60    1287.04   0.320 0.749236
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2307 on 570 degrees of freedom
Multiple R-squared:  0.9513,    Adjusted R-squared:  0.9487
F-statistic: 359.5 on 31 and 570 DF,  p-value: < 2.2e-16
```

Having found the single best feature, we can look for the best feature out of the remaining features. This is a kind of greedy algorithm for feature selection: we find the best feature, then the best of the remaining features, etc.

This approach is not guaranteed to find the two best features, however. Think of trying to find the best basketball team by finding the best basketball player, then the next-best basketball player, etc. The top five people may not work together to be the best team.