*Investigate the impact of a number of automobile engine factors on the vehicle’s mpg. The dataset auto-mpg.csv contains information for 398 different automobile models. Information regarding the number of cylinders, displacement, horsepower, weight, acceleration, model year, origin, and car name as well as mpg are contained in the file.*

1. I checked if all the fields which should have been numeric were indeed numeric. The Horsepower field contained some ? values, so I deleted those rows, and then converted the rest of the values from string to int.
2. I split the data into a training set with 300 rows and a testing set with 92 rows (6 rows had been deleted because they contained non-numeric data).
3. I ran 3 models with many factors – a full model, a forwards stepwise model and a backwards stepwise model. However, in all the models I did not include the car.name predictor because it was split into many, insignificant predictors; and it could not be used on the test data because it had different car names. The results were:
   * *Full Model:*

Multiple R Squared: 0.8216

Adjusted R Squared: 0.8173

Complete Linear Regression Equation:

y = 4.870896 - 0.419028(cylinder) + 0.009598(displacement) - 0.017772(horsepower) - 0.005373(weight) - 0.035080(acceleration) + 0.461178(model.year) + 0.926860(origin)

*Forwards Model:*

Multiple R Squared: 0.8216

Adjusted R Squared: 0.8173

Complete Linear Regression Equation:

y = 4.870896 - 0.419028(cylinder) + 0.009598(displacement) - 0.017772(horsepower) - 0.005373(weight) - 0.035080(acceleration) + 0.461178(model.year) + 0.926860(origin)

* + *Backwards Model:*

Multiple R Squared: 0.8191

Adjusted R Squared: 0.8173

Complete Linear Regression Equation:

y = 2.260259 - 0.005739(weight) + 0.009598(displacement) + 0.475641(model.year) + 0.778639(origin)

The forwards model was the same as the full model. The backwards model took out some variables, but the adjusted R Squared was the same.

1. I then ran 4 more models, choosing my independent factors based on which ones were most significant in the previous models:
   * *By weight and model.year (model1):*

Multiple R Squared: 0.814

Adjusted R Squared: 0.8127

Complete Linear Regression Equation:

y = 5.0707696 - 0.0061381(weight) + 0.4701286(model.year)

* + *By weight, model.year and origin (model2):*

Multiple R Squared: 0.8191

Adjusted R Squared: 0.8173

Complete Linear Regression Equation:

y = 2.260259 - 0.005739(weight) - 0.0196961(acceleration) + 0.475641(model.year) + 0.778639(origin)

* + *By weight (model3):*

Multiple R Squared: 0.7714

Adjusted R Squared: 0.7706

Complete Linear Regression Equation:

y = 40.5619792 - 0.0062905(weight)

* + *By model.year (model4):*

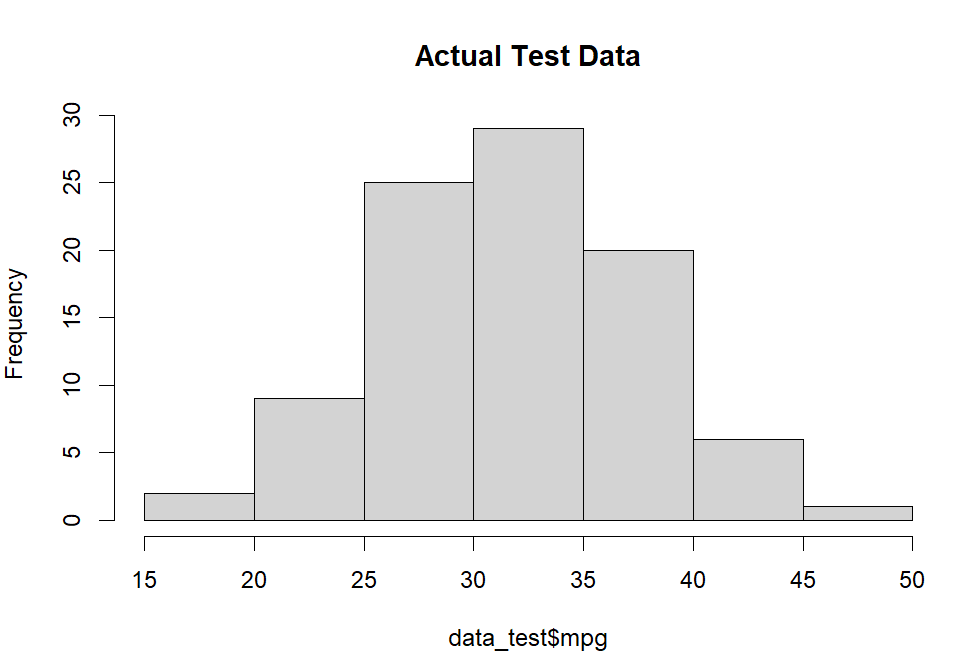
Multiple R Squared: 0.08723

Adjusted R Squared: 0.08417

Complete Linear Regression Equation:

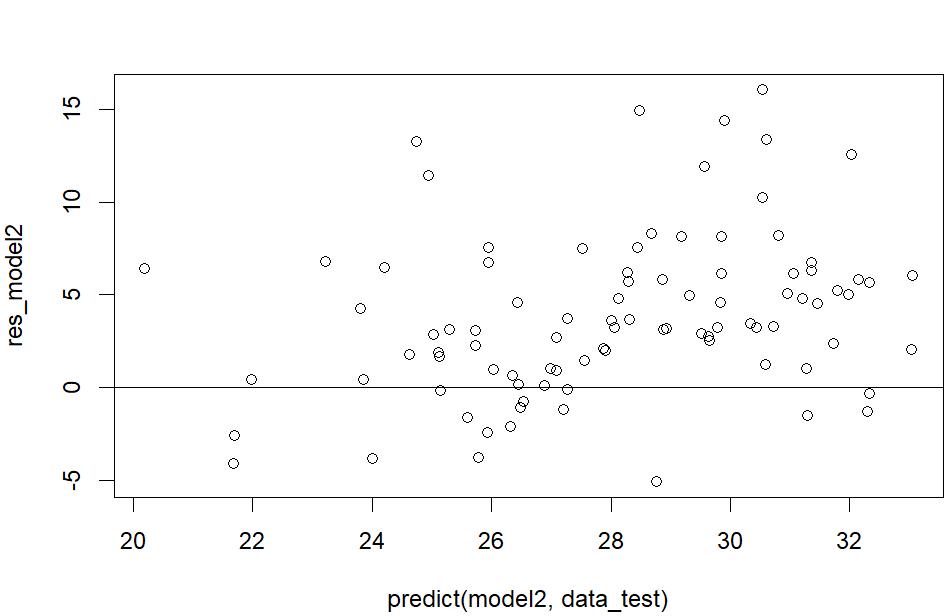
y = -28.9942 + 0.6691(model.year)

1. The best models appear to be the backwards model and by weight, model.year and origin (model2). I then tested these models on my test data.
   * The histogram from the actual data:

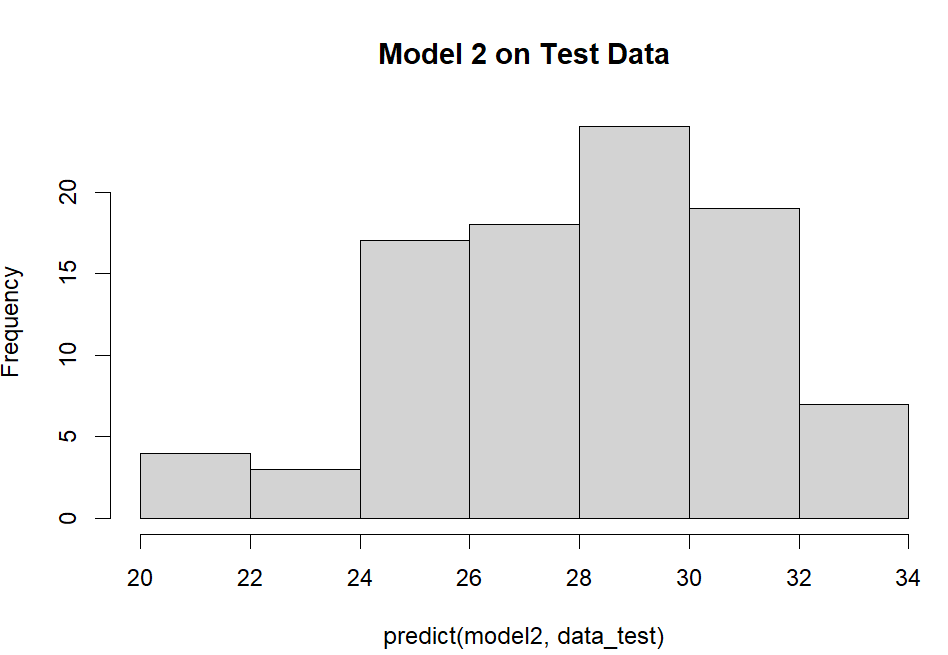


* + *model2:*

Residual Plot: A good model would be randomly scattered around 0. This is randomly scattered, but not really around 0.

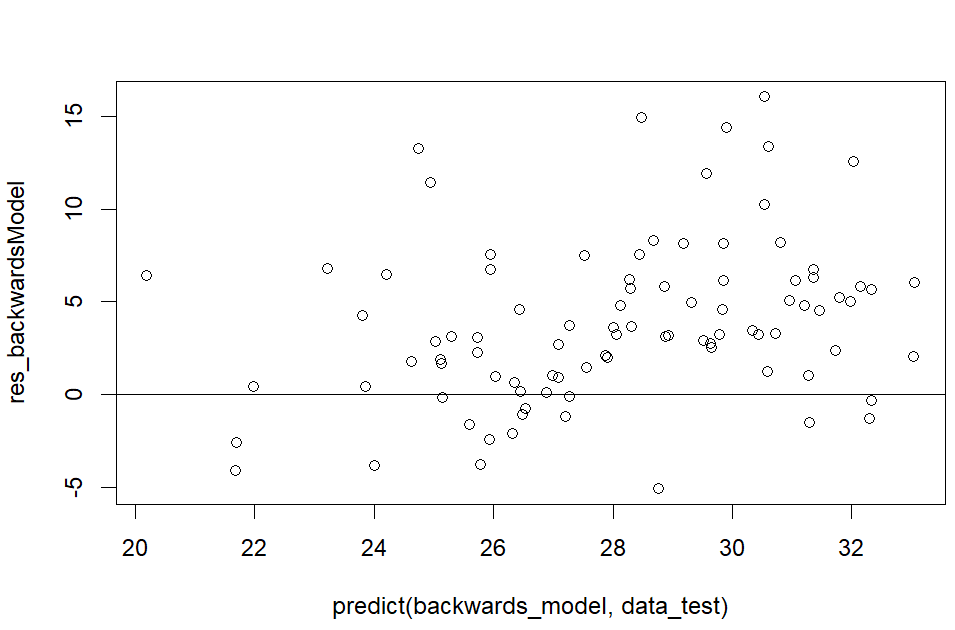


Histogram: The data appears to be skewed to the left compared to the actual data.

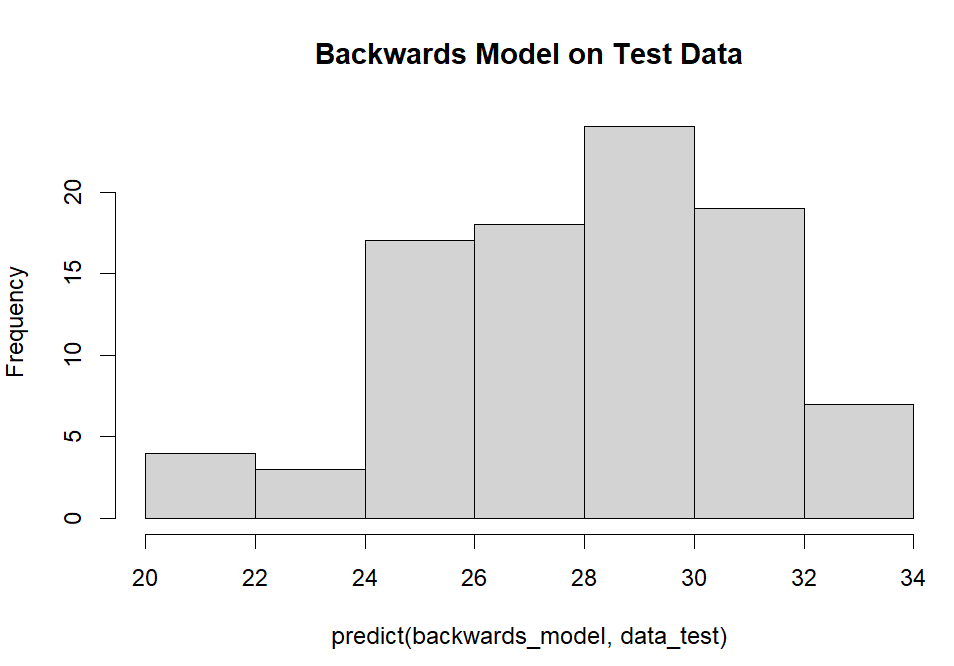


* + *Backwards Model:*

Residual Plot: Again, the plot, practically identical to the previous one, is randomly scattered, but not around 0



Histogram: The histogram also appears to be practically identical to that of model2.



1. Because the two models I chose gave out such similar outputs, I also tested model1, which was weight and model.year:
   * *model1:*

A graph with numbers and dots

Description automatically generatedResidual Plot: The plot is even less random than the previous ones.

Histogram: Appears to be even more skewed to the left.

