DS_Final_Code

Sarah Milstein

2023-12-15

```r
library(MASS)

vehicles <- read.csv("/Users/zissimilstein/Downloads/auto-mpg.csv")
str(vehicles)
```

```
## 'data.frame':    398 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinder    : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : chr  "130" "165" "150" "150" ...
##  $ weight      : int  3504 3693 3436 3433 3449 4341 4354 4312 4425
3850 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ model.year  : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ car.name    : chr  "chevrolet chevelle malibu" "buick skylark
320" "plymouth satellite" "amc rebel sst" ...
```

```r
# Convert horsepower to an integer
vehicles$horsepower <- as.integer(vehicles$horsepower)
```

```
## Warning: NAs introduced by coercion
```

```r
# Split the data into training and testing
vehicles_train <- vehicles[1:300,]
vehicles_test <- vehicles[301:398,]

# Using the first 300 samples in the auto-mpg.csv, run a simple linear
regression and multiple linear regression
# to determine the relationship between mpg and appropriate
independent variable/(s).
# Report all the appropriate information regarding your regression.

full_model <- lm(mpg ~ ., data=vehicles_train)
# summary(full_model)
```

```
# Multiple R-squared: .9789
# Adjusted R-squared:.9022
# Complete Linear Regression equation: Including the car names has the
best R value but is very complicated to create a linear model
# since each car has it's own value. Additionally the testing set will
have other car names that can't be predicted based on these.
# This is without the car names:

full_model <- lm(mpg ~
cylinder+displacement+horsepower+weight+acceleration+model.year+origin
, data=vehicles_train)
summary(full_model)

##
## Call:
## lm(formula = mpg ~ cylinder + displacement + horsepower + weight +
##     acceleration + model.year + origin, data = vehicles_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.298 -1.641  0.089  1.578 13.587
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.8118427  4.9722754   1.169  0.24342
## cylinder     -0.4562389  0.2996507  -1.523  0.12896
## displacement  0.0101272  0.0067125   1.509  0.13246
## horsepower   -0.0172493  0.0120542  -1.431  0.15351
## weight       -0.0053282  0.0005719  -9.316  < 2e-16 ***
## acceleration -0.0278409  0.0956550  -0.291  0.77122
## model.year    0.4439943  0.0605543   7.332 2.27e-12 ***
## origin        0.9931335  0.2993730   3.317  0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.683 on 290 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.823,  Adjusted R-squared:  0.8187
## F-statistic: 192.7 on 7 and 290 DF,  p-value: < 2.2e-16
```

```r
# Multiple R-squared: .823
# Adjusted R-squared:.8187
# Complete Linear Regression equation: Y = 5.8118427 + -.4562389*B1 +
0.0101272*B2 + -0.0172493*B3 + -0.0053282*B4 + -0.0278409*B5 +
0.4439943*B6 + 0.9931335*B7

backward_model <- stepAIC(full_model, direction = "backward")

## Start:  AIC=596.09
## mpg ~ cylinder + displacement + horsepower + weight + acceleration
+
##     model.year + origin
##
##                Df Sum of Sq    RSS    AIC
## - acceleration  1      0.61 2088.0 594.17
## <none>                      2087.4 596.09
## - horsepower    1     14.74 2102.2 596.18
## - displacement  1     16.38 2103.8 596.42
## - cylinder      1     16.69 2104.1 596.46
## - origin        1     79.21 2166.7 605.19
## - model.year    1    386.97 2474.4 644.77
## - weight        1    624.72 2712.2 672.10
##
## Step:  AIC=594.17
## mpg ~ cylinder + displacement + horsepower + weight + model.year +
##     origin
##
##                Df Sum of Sq    RSS    AIC
## <none>                      2088.0 594.17
## - cylinder      1     16.24 2104.3 594.48
## - displacement  1     17.35 2105.4 594.64
## - horsepower    1     17.36 2105.4 594.64
## - origin        1     79.89 2167.9 603.36
## - model.year    1    388.26 2476.3 642.99
## - weight        1    825.55 2913.6 691.45

summary(backward_model)

##
## Call:
## lm(formula = mpg ~ cylinder + displacement + horsepower + weight +
##     model.year + origin, data = vehicles_train)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.2324 -1.6368  0.1072  1.5954 13.4927
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.263056   4.593671   1.146 0.252853
## cylinder     -0.448111   0.297877  -1.504 0.133576
## displacement  0.010352   0.006657   1.555 0.121027
## horsepower   -0.015202   0.009773  -1.555 0.120930
## weight       -0.005406   0.000504 -10.726  < 2e-16 ***
## model.year    0.444528   0.060431   7.356 1.94e-12 ***
## origin        0.996595   0.298666   3.337 0.000958 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.679 on 291 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.823,  Adjusted R-squared:  0.8193
## F-statistic: 225.5 on 6 and 291 DF,  p-value: < 2.2e-16

# Multiple R-squared: .823
# Adjusted R-squared:.8193
# Complete Linear Regression equation: Y = 5.8118427 + -0.448111*B1 +
0.010352*B2 + -0.015202*B3 + -0.005406*B4 + 0.444528*B5 + 0.996595*B6

my_model <- lm(mpg ~ weight+model.year+origin, data=vehicles_train)
summary(my_model)

##
## Call:
## lm(formula = mpg ~ weight + model.year + origin, data =
vehicles_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1750 -1.5586  0.0463  1.6506 13.6915
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  3.2289898  4.3438467   0.743  0.45786
## weight      -0.0056852  0.0002215 -25.664  < 2e-16 ***
## model.year   0.4585332  0.0559760   8.192 7.82e-15 ***
## origin       0.8495008  0.2646593   3.210  0.00147 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.676 on 296 degrees of freedom
## Multiple R-squared:  0.8206, Adjusted R-squared:  0.8188
## F-statistic: 451.3 on 3 and 296 DF,  p-value: < 2.2e-16
```

```r
# Multiple R-squared: .8206
# Adjusted R-squared:.8188
# Complete Linear Regression equation: Y = 3.2289898 + -0.0056852*B1 +
0.4585332*B2 + 0.8495008*B3

# For the remaining 98 samples in the dataset, use your best linear
model(s) to predict each automobile’s mpg and
# report how your predictions compare to the car’s actual reported
mpg.

# Use my model to take in the values in the dataset and predict what
the mpg would be:
predictions <- predict(my_model, vehicles_test)
# Store the actual mpg in actual outcomes
actual_outcomes <- vehicles_test$mpg

# Compare the two:
residuals <- actual_outcomes - predictions

summary(residuals)
```

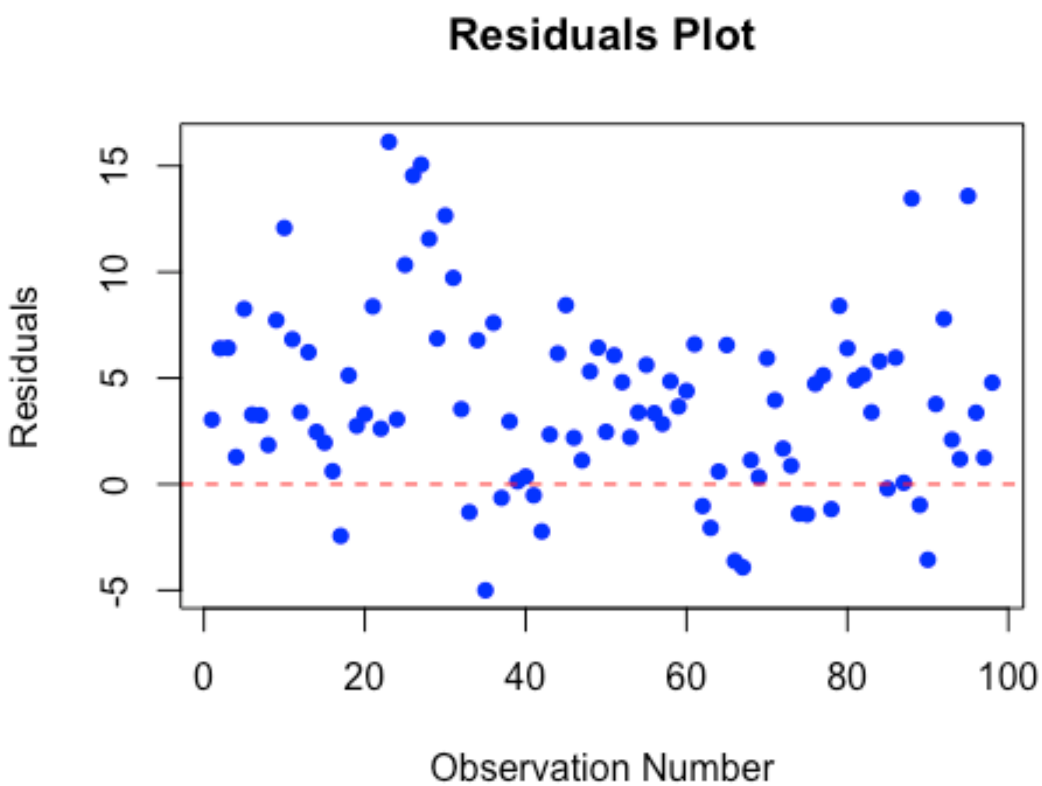
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -5.002   1.145   3.384   3.995   6.404  16.136
```

```r
# The closer to zero the better the model. Here our median residual is
3.995
summary(actual_outcomes)
```

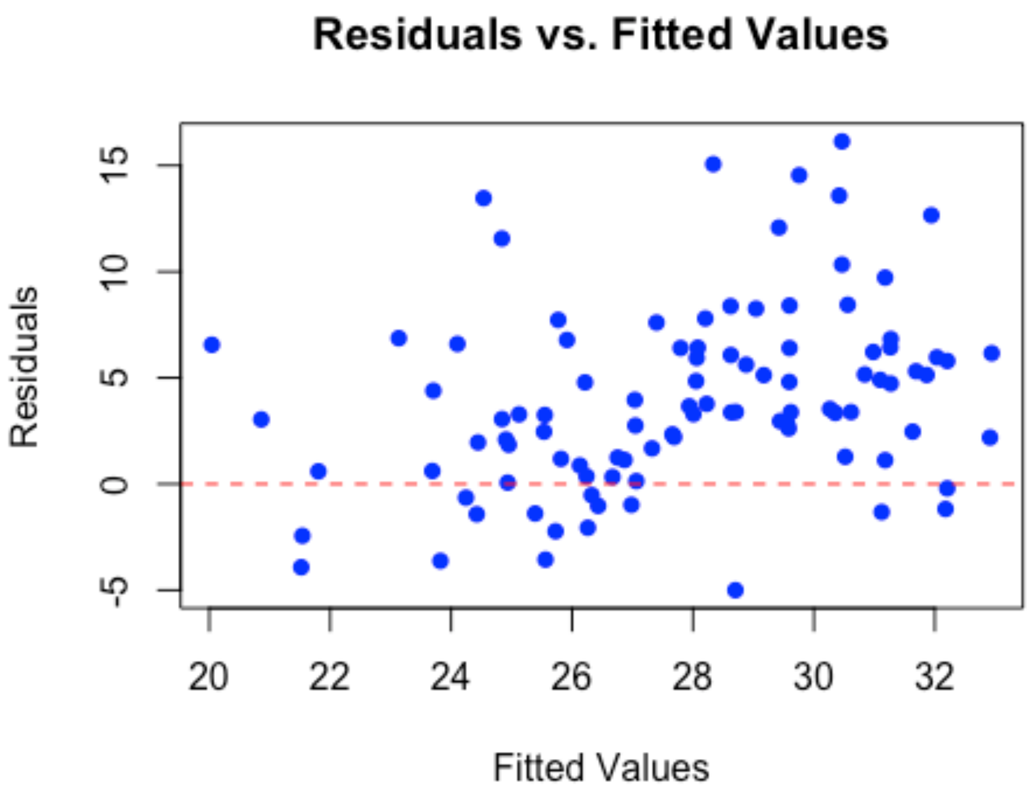
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.60   27.05   32.05   31.83   36.00   46.60
```

```r
summary(predictions)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.04   25.74   28.06   27.83   30.40   32.94

#Residual Plot:

# Checks the distribution of the residuals

plot(residuals,
     main = "Residuals Plot",
     xlab = "Observation Number",
     ylab = "Residuals",
     col = "blue",
     pch = 16)
abline(h = 0, col = "red", lty = 2)
```

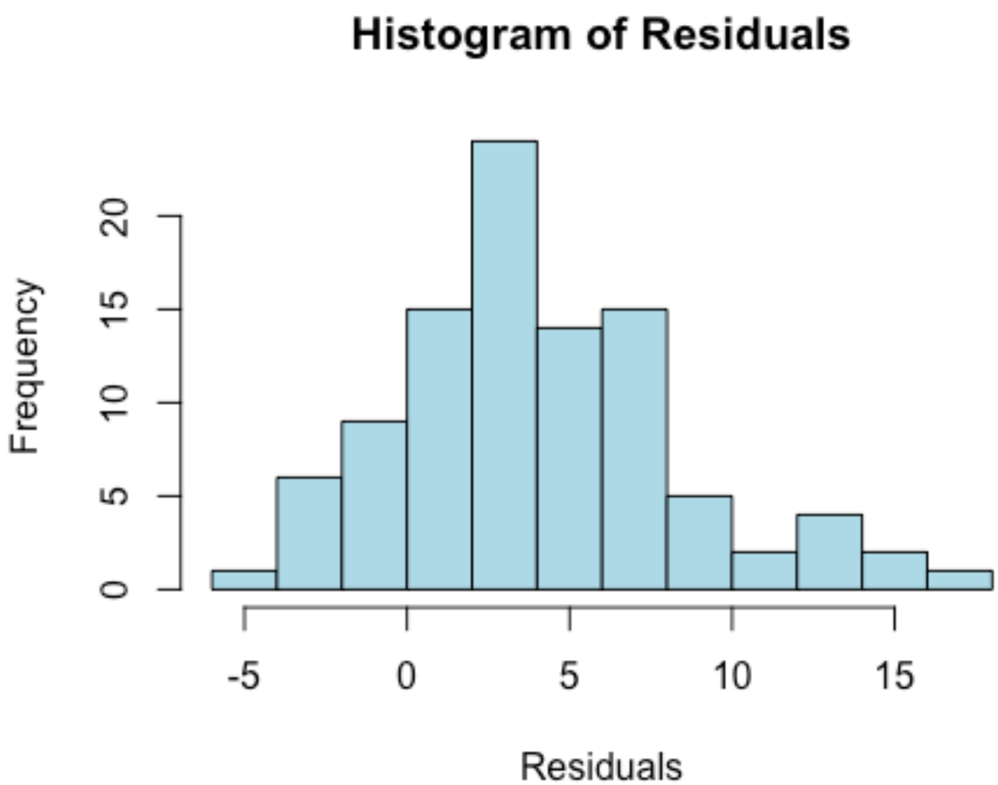


```r
# It doesn't look to great, there are many above the zero line and not many below it

# Compares the residuals to the predictions:

plot(x = predictions, y = residuals,
     main = "Residuals vs. Fitted Values",
     xlab = "Fitted Values",
     ylab = "Residuals",
     col = "blue",
     pch = 16)
abline(h = 0, col = "red", lty = 2)
```

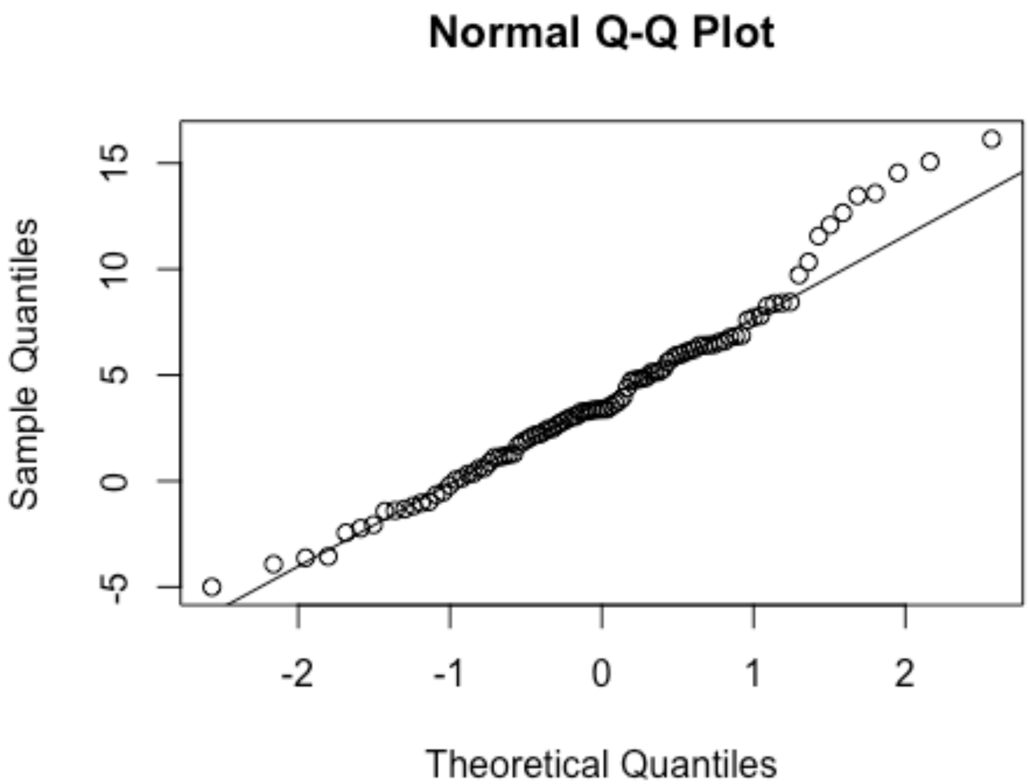



```r
# Histogram:

hist(residuals,
```

```
     main = "Histogram of Residuals",
     xlab = "Residuals",
     ylab = "Frequency",
     col = "lightblue",
     border = "black")
```




```
# The histogram shows that majority of the residuals are between 0 and 5. While not perfect, it is a good average.
# As we saw in the chart there are many more towards the positive side than towards the negative.

# QQ plot

qqnorm(residuals)
qqline(residuals)
```




```
# The points of the residuls fit the line but start shifting off by
the 1 point. There might be some outliers that are throwing the model
off.
```