**Automobile Vehicle MPG**

**Date – 10/15/2018**

**Ankur Patel**

**Table of Contents:**

1: Introduction ………..………………………………………………………………………..3

2: Read, View, Structure, Summary …..…………………………………………………...4

3: Clean Data ……………………………………….……………………………………...…6

4: Train Dataset & Test Dataset ………………………………….…………………………8

5: Multiple R-Squared ………………………………………………………………..……….9

6: Adjusted R-Squared …………………………………………………………………...…25

7: Complete Linear Regression Equation ……………………………………………….29

8: Predict & Compare ………………………………………………………………………30

9: Residual Plot ……………………………………………………………………………….32

10: Histogram …………………………………………………………………………...……33

11: Transaction Log ……………………………………………………………………….…35

12. Bonus ………………………………………………………………………………………38

13: Conclusion ………………………………………………………………………………..39

**INTRODUCTION**

This project is to investigate the impact of a number of automobile engine factors on the vehicle’s mpg. I will walk you through the process of analyzing and interpreting all the steps. I used the dataset "auto-mpg.csv" that contains information for 398 different automobile models. Variables analyzed: mpg, cylinder, displacement, horsepower, weight, acceleration, model.year, origin, and car.name.

Firstly, using 300 observations in 'auto-mpg.csv' as train dataset, I will run linear regression individually to determine the relationship between mpg and appropriate variables. As I analyze the data and the results with the train dataset, I will adjust the parameters to get the intended/best result. The remaining observations of the dataset will be in test dataset and will be analyzed using a complete linear regression equation gotten from the train dataset. All appropriate results, a log, and comments will be included in this report to better understand the methods and the journey.

**READ, VIEW, STRUCTURE, SUMMARY**

Firstly, I read the "auto-mpg" file using 'read.csv(file(choose))' function to get an access to the data for this project. I can also use 'read.csv("/Users/...path.../auto-mpg.csv")', but I'll use the first method here. I view it using a 'View()' function to get a picture for the data. I then get a statistical summary using 'summary()' function to better understand. I also like to use the 'str()' function to know the basic details of its structure in the start.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> auto\_mpg <- read.csv(file.choose())

> View(auto\_mpg)

> str(auto\_mpg)

'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 : Factor w/ 94 levels "?","100","102",..: 17 35 29 29 24 42 47 46 48 40 ...

$ 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 : Factor w/ 305 levels "amc ambassador brougham",..: 50 37 232 15 162 142 55 224 242 2 ...

> summary(auto\_mpg)

mpg cylinder displacement horsepower weight acceleration model.year origin

Min. : 9.00 Min. :3.000 Min. : 68.0 150 : 22 Min. :1613 Min. : 8.00 Min. :70.00 Min. :1.000

1st Qu.:17.50 1st Qu.:4.000 1st Qu.:104.2 90 : 20 1st Qu.:2224 1st Qu.:13.82 1st Qu.:73.00 1st Qu.:1.000

Median :23.00 Median :4.000 Median :148.5 88 : 19 Median :2804 Median :15.50 Median :76.00 Median :1.000

Mean :23.51 Mean :5.455 Mean :193.4 110 : 18 Mean :2970 Mean :15.57 Mean :76.01 Mean :1.573

3rd Qu.:29.00 3rd Qu.:8.000 3rd Qu.:262.0 100 : 17 3rd Qu.:3608 3rd Qu.:17.18 3rd Qu.:79.00 3rd Qu.:2.000

Max. :46.60 Max. :8.000 Max. :455.0 75 : 14 Max. :5140 Max. :24.80 Max. :82.00 Max. :3.000

(Other):288

car.name

ford pinto : 6

amc matador : 5

ford maverick : 5

toyota corolla: 5

amc gremlin : 4

amc hornet : 4

(Other) :369

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This dataset has 9 variables (mpg is 1; other 8 variables will be analyzed to study its feature). The types for this data are num, int, and Factor. The reason for horsepower being a factor is that it includes some '?'. The variables I will use for correlation between mpg and other variables will be numeric, the same type as mpg. The car.name variable values are non-numeric, and thus will not be included in the analysis.

This summary gives us the basic statistics such as Min, 1st Qu, Median, Mean, 3rd Qu, and Max for each variable. We should glance through it to get a good perspective. Statistics of mpg variable: IQR = 29.00 - 17.50 = 11.5; mpg dataset is well-within lower and upper boundaries because they are within 1.5\*IQR range. As I analyze each variable individually, I will comment on what I observe.

**CLEAN DATA**

Non-numeric variables need their types changed to numeric to avoid errors. We will change the 'horsepower' variable (factor type) to numeric. The 'horsepower' variable column also contains some '?' values, whose rows will be excluded in new dataset. I found the row numbers by putting the observations in 'horsepower' in ascending order, and that grouped the ‘?’ values together on top. The variable of car.name with non-numeric values will also be excluded. I'll get a summary of the linear regression on the full dataset in the start by using 'mpg~.' (. means all columns).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> auto\_mpg1 <- auto\_mpg[-c(33,127,331,337,355,375),]

> auto\_mpg\_clean <- auto\_mpg1[,-c(9)]

> View(auto\_mpg\_clean)

> auto\_mpg\_clean$horsepower <- as.numeric(auto\_mpg\_clean$horsepower)

> model <- lm(mpg~., data = auto\_mpg)

> summary(model)

# not including summary results because it includes all the 398 observations of ‘horsepower’ variable since it also includes ‘?’ values and car.names because they’re different strings

> model1 <- lm(mpg~., data = auto\_mpg\_clean)

> summary(model1)

Call:

lm(formula = mpg ~ ., data = auto\_mpg\_clean)

Residuals:

Min 1Q Median 3Q Max

-9.656 -2.069 -0.043 1.775 13.098

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -2.134e+01 4.292e+00 -4.971 1.00e-06 \*\*\*

cylinder -2.739e-01 3.398e-01 -0.806 0.4208

displacement 1.539e-02 7.275e-03 2.116 0.0350 \*

horsepower 1.072e-02 7.076e-03 1.515 0.1306

weight -6.756e-03 5.828e-04 -11.592 < 2e-16 \*\*\*

acceleration 1.489e-01 7.777e-02 1.914 0.0563 .

model.year 7.688e-01 4.972e-02 15.463 < 2e-16 \*\*\*

origin 1.344e+00 2.701e-01 4.975 9.89e-07 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.324 on 384 degrees of freedom

Multiple R-squared: 0.8218, Adjusted R-squared: 0.8186

F-statistic: 253.1 on 7 and 384 DF, p-value: < 2.2e-16

> str(auto\_mpg\_clean)

'data.frame': 392 obs. of 8 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 : num 17 35 29 29 24 42 47 46 48 40 ...

$ 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 ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The Multiple R-squared for the mpg vs. entire dataset was ~0.99. Without the color.name variable, it was ~0.82. We will not include them in our regression because we want to find a good correlation with a good R-squared value by using only the numeric values.

We will clean again after we run correlation for the variables individually to determine which ones to use according to their significance and R-squared.

**TRAIN DATASET & TEST DATASET**

Divide the "auto-mpg" dataset into 2 datasets: train dataset of 300 observations and test dataset of 301 to 392 observations. The test dataset was initially 301 to 398 observations, but I reduced it since we removed the 6 rows with ‘?’ values for ‘horsepower’. \*\*Could stores the randomly generated values but that will change the values every time, so not ideal for this report.\*\*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> View(auto\_mpg\_clean)

> train <- auto\_mpg\_clean[1:300,]

> View(train)

> # test <- auto\_mpg\_clean[301:398,]

> test <- auto\_mpg\_clean[301:392,]

> View(test)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Again, the train dataset will be used to determine a complete linear regression equation that will be used for the test dataset.

**MULTIPLE R-SQUARED:**

MPG vs. Cylinder

Scatterplot to find the relationship between 'mpg' and 'cylinder' in the training set.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> traincylinder <- lm(train$mpg ~ train$cylinder, data = train)

> summary(traincylinder)

Call:

lm(formula = train$mpg ~ train$cylinder, data = train)

Residuals:

Min 1Q Median 3Q Max

-11.1071 -2.3012 -0.4306 1.8282 16.9282

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 37.9130 0.7356 51.54 <2e-16 \*\*\*

train$cylinder -2.9353 0.1211 -24.24 <2e-16 \*\*\*

---

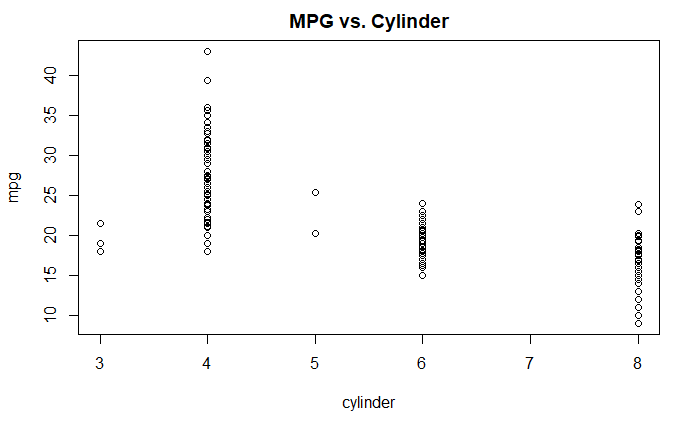
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.678 on 298 degrees of freedom

Multiple R-squared: 0.6636, Adjusted R-squared: 0.6624

F-statistic: 587.8 on 1 and 298 DF, p-value: < 2.2e-16

> plot(train$cylinder, train$mpg, main = "MPG vs. Cylinder", xlab = "cylinder", ylab = "mpg")



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This scatterplot shows us the distribution of cylinders. 4 cylinders have a higher range of mpg as well as higher values of mpg, containing the maximum of 46. 8 cylinders has some of the lowest mpg observations, containing the minimum of 9. 6 cylinders subset group seem to be around the mean ‘mpg’ of the full dataset (‘auto\_mpg’) of 23.51.

The linear model on a train dataset of 300 observations gave us a \*\*\* significance of <2e-16 p-value and ~0.66 Multiple R-squared. The \*\*\* tells us that it has 100% significance, and the extremely low p-value being close to 0 shows that it is sufficiently consistent. The coefficient of determination, denoted as r^2 and pronounced “R-squared”, is the proportion of the variance in the dependable variable (mpg) that is predictable from the independent variable (cylinder); a multiple R-squared of 0.66 will be used to adjust, in case it gives us better results. The Adjusted R-squared value is also ~0.66 because it hasn't been adjusted yet.

MPG vs. Displacement

Scatterplot to find the relationship between 'mpg' and 'displacement' in the training set. From the table, 'displacement' values seem to decrease as mpg values increase.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> traindisplacement <- lm(train$displacement ~ train$mpg, data = train)

> summary(traindisplacement)

Call:

lm(formula = train$displacement ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-186.167 -35.216 -2.498 33.558 198.391

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 517.6038 11.6978 44.25 <2e-16 \*\*\*

train$mpg -14.5242 0.5372 -27.04 <2e-16 \*\*\*

---

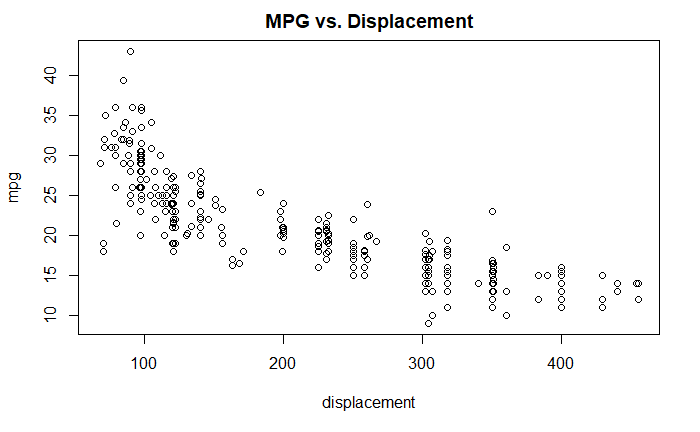
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 58.8 on 298 degrees of freedom

Multiple R-squared: 0.7104, Adjusted R-squared: 0.7094

F-statistic: 731.1 on 1 and 298 DF, p-value: < 2.2e-16

> plot(train$displacement, train$mpg, main = "MPG vs. Displacement", xlab = "displacement", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This scatterplot shows us a linear negative-slope trend in 'displacement' when comparing mpg. It does have a variance, especially at lower 'displacement'. The results show its significance of \*\*\* with a p-value of 2.2e-16 and Multiple R-squared of ~0.71. It will be included in the adjusted regression.

MPG vs. Horsepower

Scatterplot to see the relationship between 'mpg' and 'horsepower' in the training dataset. I would assume that it would be significant.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> trainhorsepower <- lm(train$horsepower ~ train$mpg, data = train)

> summary(trainhorsepower)

Call:

lm(formula = train$horsepower ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-50.697 -18.444 -0.606 20.316 50.798

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.5888 5.2436 0.112 0.911

train$mpg 2.3119 0.2408 9.601 <2e-16 \*\*\*

---

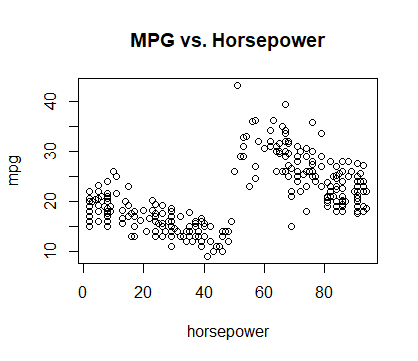
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26.36 on 298 degrees of freedom

Multiple R-squared: 0.2363, Adjusted R-squared: 0.2337

F-statistic: 92.18 on 1 and 298 DF, p-value: < 2.2e-16

> plot(train$horsepower, train$mpg, main = "MPG vs. Horsepower", xlab = "horsepower", ylab = "mpg")



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Although it shows a significance code of \*\*\*, since it's Multiple R-squared values are ~0.23, it is too variable. This ‘horsepower’ dataset can be split into 2 datasets according to horsepower. I will split it and test to see the difference.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> median(train$horsepower)

[1] 47.5

> train$horsepower1 <- train$horsepower[train$horsepower < 47.5]

> trainhorsepower1 <- lm(train$horsepower1 ~ train$mpg, data = train)

> summary(trainhorsepower1)

Call:

lm(formula = train$horsepower1 ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-20.209 -13.648 2.139 10.012 25.183

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 23.03172 2.69058 8.560 6.09e-16 \*\*\*

train$mpg -0.06326 0.12355 -0.512 0.609

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 13.53 on 298 degrees of freedom

Multiple R-squared: 0.000879, Adjusted R-squared: -0.002474

F-statistic: 0.2622 on 1 and 298 DF, p-value: 0.609

> train$horsepower2 <- train$horsepower[train$horsepower > 47.5]

> trainhorsepower2 <- lm(train$horsepower ~ train$mpg, data = train)

> summary(trainhorsepower2)

Call:

lm(formula = train$horsepower ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-50.697 -18.444 -0.606 20.316 50.798

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.5888 5.2436 0.112 0.911

train$mpg 2.3119 0.2408 9.601 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26.36 on 298 degrees of freedom

Multiple R-squared: 0.2363, Adjusted R-squared: 0.2337

F-statistic: 92.18 on 1 and 298 DF, p-value: < 2.2e-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This was an interesting study of difference between the 2 groups of ‘horsepower’. The ‘horsepower1’ set, which was the values under the median of 47.5, had a Multiple R-squared of 0.000879 and ‘ ‘ significance. The ‘horsepower2’ set, which was the values higher than the median of 47.5, had a Multiple R-squared of 0.2363 and \*\*\* significance. Since both their Multiple R-squared values are low, I would predict that they won’t be included in the complete linear regression equation.

MPG vs. Weight

Scatterplot to see the relationship between 'mpg' and 'weight' in the training dataset. Again, I would assume that it would be significant.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> trainweight <- lm(train$weight ~ train$mpg, data = train)

> summary(trainweight)

Call:

lm(formula = train$weight ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-1359.51 -273.50 -0.02 264.01 1579.43

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 5690.791 84.216 67.57 <2e-16 \*\*\*

train$mpg -122.627 3.867 -31.71 <2e-16 \*\*\*

---

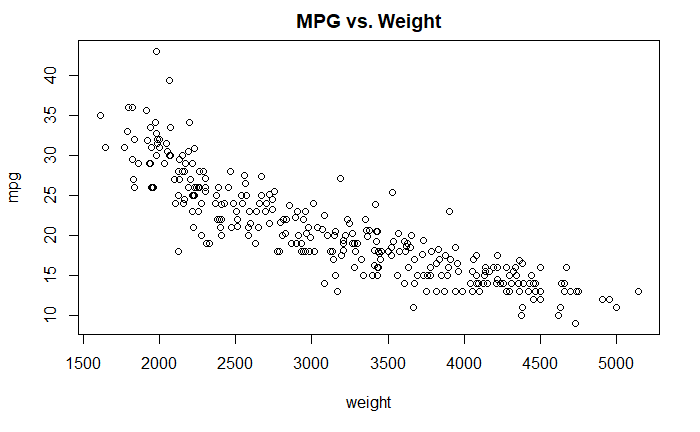
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 423.3 on 298 degrees of freedom

Multiple R-squared: 0.7714, Adjusted R-squared: 0.7706

F-statistic: 1005 on 1 and 298 DF, p-value: < 2.2e-16

> plot(train$weight, train$mpg, main = "MPG vs. Weight", xlab = "weight", ylab = "mpg")



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Voila, the highest significance so far. The results show \*\*\* significance of <2.2e-16 p-value and ~0.77 Multiple R-squared. Weight variable will be included in the adjusted regression.

MPG vs. Acceleration

Scatterplot to see the relationship between mpg and acceleration in the training dataset. Again, I would assume for this to be significant, but since the horsepower was insignificant, I wouldn't put my money on this.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> trainacceleration <- lm(train$acceleration ~ train$mpg, data = train)

> summary(trainacceleration)

Call:

lm(formula = train$acceleration ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-5.9084 -1.7107 -0.2921 1.4030 8.2824

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 11.14106 0.49082 22.70 <2e-16 \*\*\*

train$mpg 0.19767 0.02254 8.77 <2e-16 \*\*\*

---

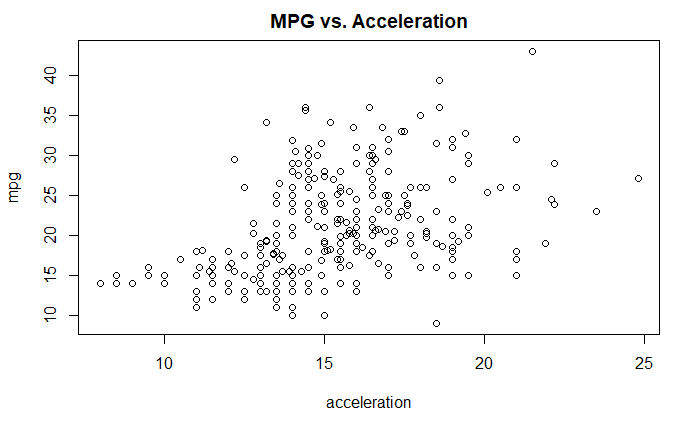
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.467 on 298 degrees of freedom

Multiple R-squared: 0.2052, Adjusted R-squared: 0.2025

F-statistic: 76.91 on 1 and 298 DF, p-value: < 2.2e-16

> plot(train$acceleration, train$mpg, main = "MPG vs. Acceleration", xlab = "acceleration", ylab = "mpg")



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

It looks like I saved my bet. Although the coefficients repeatedly have \*\*\* significance and p-value of <2.2e-16, the Multiple R-squared of ~0.2 is too low to be significant.

MPG vs. Model Car

Scatterplot to see the relationship between 'mpg' and 'model.car' in the training dataset. I wouldn't think it would be much significant, but I would assume that there was a continuous slow increase during that time period.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> trainmodel.year <- lm(train$model.year ~ train$mpg, data = train)

> summary(trainmodel.year)

Call:

lm(formula = train$model.year ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-5.3228 -1.9428 0.1113 2.1002 5.2194

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 71.75982 0.53199 134.890 < 2e-16 \*\*\*

train$mpg 0.13037 0.02443 5.337 1.88e-07 \*\*\*

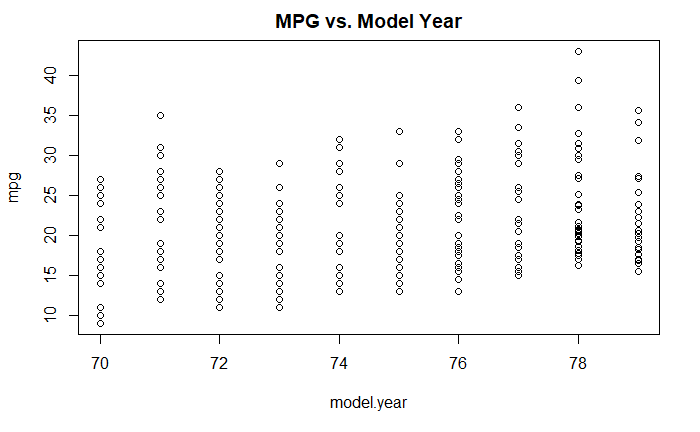
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.674 on 298 degrees of freedom

Multiple R-squared: 0.08723, Adjusted R-squared: 0.08417

F-statistic: 28.48 on 1 and 298 DF, p-value: 1.878e-07

> plot(train$model.year, train$mpg, main = "MPG vs. Model Year", xlab = "model.year", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Somewhat correct assumption. It can be seen as a tiny increasing wave, but it's variance is tremendously high. That can be proved by the Multiple R-squared value of ~0.08. Thus, it is quite unlikely to be included in the complete regression.

MPG vs. Origin

Scatterplot to see the relationship between 'mpg' and 'origin' in the training dataset.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> trainorigin <- lm(train$origin ~ train$mpg, data = train)

> summary(trainorigin)

Call:

lm(formula = train$origin ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-1.52042 -0.34713 -0.07268 0.20178 1.72148

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.043476 0.118852 0.366 0.715

train$mpg 0.068613 0.005458 12.572 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

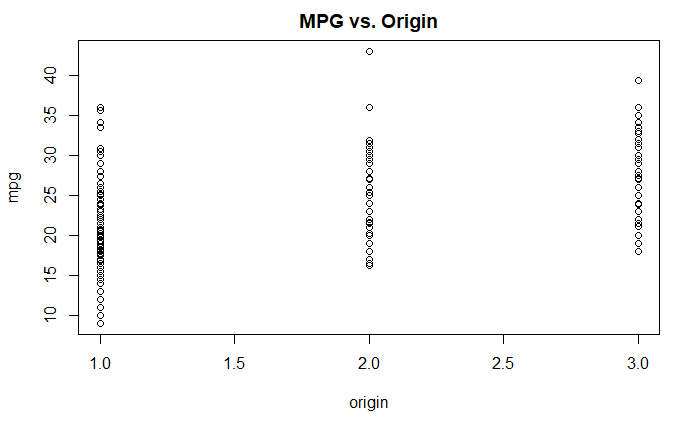
Residual standard error: 0.5975 on 298 degrees of freedom

Multiple R-squared: 0.3466, Adjusted R-squared: 0.3444

F-statistic: 158 on 1 and 298 DF, p-value: < 2.2e-16

> plot(train$origin, train$mpg, main = "MPG vs. Origin", xlab = "origin", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Last but not the least, the origin variable is insignificant because of its Multiple R-squared value of ~0.35.

MULTIPLE REGRESSION TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| **Regression** | **p-value** | **Significance** | **Multiple R-squared** |
| MPG vs. Cylinder | <2e-16 | \*\*\* | 0.6636 |
| MPG vs. Displacement | <2e-16 | \*\*\* | 0.7104 |
| MPG vs. Horsepower | <2e-16 | \*\*\* | 0.2363 |
| MPG vs. Horsepower1 (<47.5) | 0.609 |  | 0.000879 |
| MPG vs. Horsepower2 (>47.5) | < 2.2e-16 | \*\*\* | 0.2363 |
| MPG vs. Weight | < 2.2e-16 | \*\*\* | 0.7714 |
| MPG vs. Acceleration | < 2.2e-16 | \*\*\* | 0.2052 |
| MPG vs. Model Car | 1.878e-07 | \*\*\* | 0.08723 |
| MPG vs. Origin | < 2.2e-16 | \*\*\* | 0.3466 |

**ADJUSTED R-SQUARED**

Now, the variables will be combined in different ways to find the highest R-squared. What is R-squared and why is R-squared itself significant? R-squared simply explains the variability of a model; that is, how our dependent variable varies in proportion to the independent variable. We would love to get a R-squared value of 1.0 (or 100%), but there will always be variance caused by several factors. Here, we attempt to find the highest R-squared value by trying different combinations between numeric variables and "adjusting" the R-squared value.

Firstly, let's try and combine every variable with a R-squared of >0.60. The following variables had high enough significance to try: 'cylinder' (~0.66), 'displacement' (~0.71), and 'weight' (~0.77).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> attach(train)

> adjustedmodel <- lm(mpg ~ cylinder + displacement + weight, data = train)

> summary(adjustedmodel)

Call:

lm(formula = mpg ~ cylinder + displacement + weight, data = train)

Residuals:

Min 1Q Median 3Q Max

-9.559 -1.892 -0.151 1.712 15.113

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 39.0682163 1.1448111 34.126 <2e-16 \*\*\*

cylinder -0.0579278 0.3260901 -0.178 0.8591

displacement -0.0103839 0.0062078 -1.673 0.0954 .

weight -0.0049947 0.0005411 -9.231 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.009 on 296 degrees of freedom

Multiple R-squared: 0.7764, Adjusted R-squared: 0.7741

F-statistic: 342.6 on 3 and 296 DF, p-value: < 2.2e-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The cylinder variable had a 0.8591 p-value and ' ' significance, thus it has no significance. Let's try with just 'displacement' and 'weight'.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> attach(train)

> adjustedmodel2 <- lm(mpg ~ displacement + weight, data = train)

> summary(adjustedmodel2)

The following objects are masked from train (pos = 3):

acceleration, cylinder, displacement, horsepower, model.year, mpg, origin, weight

Call:

lm(formula = mpg ~ displacement + weight, data = train)

Residuals:

Min 1Q Median 3Q Max

-9.5232 -1.8741 -0.1765 1.6766 15.1041

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 38.9422109 0.8971239 43.408 <2e-16 \*\*\*

displacement -0.0111720 0.0043353 -2.577 0.0104 \*

weight -0.0050080 0.0005351 -9.360 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.004 on 297 degrees of freedom

Multiple R-squared: 0.7764, Adjusted R-squared: 0.7749

F-statistic: 515.6 on 2 and 297 DF, p-value: < 2.2e-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

There is a message that indicates that attach() is not necessary because it’s repetitive. To avoid it, I can either add detach() function or not attach() again. The adjusted regression between mpg vs. displacement and weight gives a Adjusted Regression of 0.7749 with displacement being \* (95%) significant and weight being \*\*\* (100%) significant. We will most likely run the test using this model.

The following was not intended at first, but I will run linear regression by combining every variable and adjust them according to their significance and R-squared.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> adjustedmodel3 <- lm(mpg ~ cylinder + displacement + horsepower + weight + acceleration + model.year + origin, data = train)

> summary(adjustedmodel3)

Call:

lm(formula = mpg ~ cylinder + displacement + horsepower + weight +

acceleration + model.year + origin, data = train)

Residuals:

Min 1Q Median 3Q Max

-9.2618 -1.6478 0.0955 1.6620 13.5492

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.0865676 4.6708904 0.019 0.98523

cylinder -0.2411595 0.3055229 -0.789 0.43056

displacement 0.0051518 0.0064685 0.796 0.42642

horsepower 0.0103672 0.0063847 1.624 0.10551

weight -0.0056497 0.0005166 -10.936 < 2e-16 \*\*\*

acceleration 0.0416447 0.0760485 0.548 0.58438

model.year 0.4897610 0.0594871 8.233 6.15e-15 \*\*\*

origin 0.7772528 0.2898790 2.681 0.00775 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.704 on 292 degrees of freedom

Multiple R-squared: 0.8219, Adjusted R-squared: 0.8176

F-statistic: 192.5 on 7 and 292 DF, p-value: < 2.2e-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Using all the variables for the regression resulted in ~0.82 R-squared and \*\*\* significance in ‘weight’, ‘model.year’, and ’origin’. Although I’m not an expert in car mechanics, I would assume those to be significant factors as well. I will run a regression using only those variables.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> adjustedmodel4 <- lm(mpg ~ weight + model.year + origin, data = train)

> summary(adjustedmodel4)

Call:

lm(formula = mpg ~ weight + model.year + origin, data = train)

Residuals:

Min 1Q Median 3Q Max

-9.1293 -1.6271 0.0525 1.6617 13.5735

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.260259 4.396926 0.514 0.60760

weight -0.005739 0.000225 -25.502 < 2e-16 \*\*\*

model.year 0.475641 0.056329 8.444 1.39e-15 \*\*\*

origin 0.778639 0.268320 2.902 0.00399 \*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.706 on 296 degrees of freedom

Multiple R-squared: 0.8191, Adjusted R-squared: 0.8173

F-statistic: 446.9 on 3 and 296 DF, p-value: < 2.2e-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Surprisingly, this regression can be considered the best one with the highest R-squared of ~0.82 with high significance, but since they were not significant and had high variance when analyzed individually, I wouldn’t trust including them. Thinking practically, the car’s model year and origin can give only temporary results since their mechanics change over time.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> adjustedmodel5 <- lm(mpg ~ displacement + horsepower2 + weight, data = train)

> summary(adjustedmodel5)

Call:

lm(formula = mpg ~ displacement + horsepower2 + weight, data = train)

Residuals:

Min 1Q Median 3Q Max

-9.3605 -1.8773 -0.1186 1.7334 15.1167

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 39.6025456 1.3821871 28.652 <2e-16 \*\*\*

displacement -0.0110485 0.0043441 -2.543 0.0115 \*

horsepower2 -0.0085922 0.0136715 -0.628 0.5302

weight -0.0050193 0.0005359 -9.366 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.007 on 296 degrees of freedom

Multiple R-squared: 0.7767, Adjusted R-squared: 0.7744

F-statistic: 343.1 on 3 and 296 DF, p-value: < 2.2e-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I tried to see the difference in the best R-squared (adjusted R-squared 0.7749, mpg ~ displacement + weight) after including ‘horsepower2’. Since it increased the R-squared by a bare-minimum of 0.0005, it confirmed my prediction of its insignificance.

**COMPLETE LINEAR REGRESSION EQUATION**

The adjustedmodel2 equation will be used as the complete linear regression equation for the test dataset.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> attach(train)

> adjustedmodel2 <- lm(mpg ~ displacement + weight, data = train)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PREDICT & COMPARE**

For the remaining 92 samples in the dataset, we will use the best linear models (mpg vs. displacement+weight, mpg vs. weight) to predict each automobile’s mpg. We will report how the predictions compare to the car’s actual reported mpg.

Let's first predict values using weight variable only. We will use the ‘coef()’ function to get the B0 and B1 values from the linear equation and use that to predict with the test observations. We will then compare to the actual values by calculating the difference.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> pred <- coef(trainweight)[1] + coef(trainweight)[2]\*test$mpg

> View(pred)

> test\_mpg <- test[,1]

> View(test\_mpg)

> error <- pred - test\_mpg

> summary(error)

Min. 1st Qu. Median Mean 3rd Qu. Max.

-70.22 1240.22 1728.55 1741.45 2263.24 3514.96

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The error was extremely high and ranged from ~(-70) to ~3500. Now, let's try to predict using the 'displacement' and 'weight' variables. We will find the coefficients in that adjusted model, then define them individually.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> pred1 <- coef(adjustedmodel2)[1] + coef(adjustedmodel2)[2]\*test$displacement + coef(adjustedmodel2)[3]\*test$weight

> View(pred1)

> test\_mpg1 <- test[,1]

> error1 <- test\_mpg1 - pred1

> View(error1)

> summary(error1)

Min. 1st Qu. Median Mean 3rd Qu. Max.

-2.341 3.519 6.431 6.764 9.304 19.185

> error1 <- error1 - 6.764

> summary(error1)

Min. 1st Qu. Median Mean 3rd Qu. Max.

-9.10483 -3.24530 -0.33306 -0.00035 2.54028 12.42144

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The predicted values ranged from ~16 to ~29 according to linear regression using displacement and weight variables. Error1 ranged from ~(-2) to ~19 and its mean was 6.764. To normalize that, I subtracted 6.764 from error1. The error1's IQR = ~5.8, and the Max value of error and some other higher values are outside the upper boundary of ~12. After normalizing it to the mean of 0, the errors are within the boundaries. It indicates that the data points are now in the normal curve.

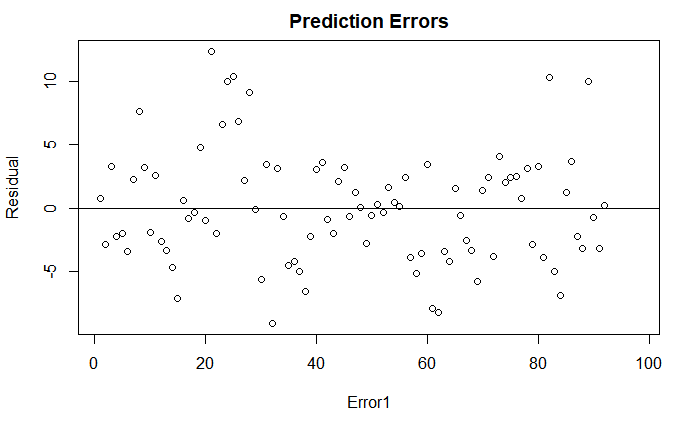
**RESIDUAL PLOT**

Following is a plot of the error differences between predicted and actual values. Let's also plot a line with intercept and slope at 0 for a clear view.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> plot(error1, main = 'Prediction Errors', xlab = 'Error1', ylab = 'Residual')

> abline(0,0)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The plot distributed nicely after normalizing the residuals to 0. At first, it was mainly positive with only some negative values.

**HISTOGRAM**

We will also create a histogram with the error1 data, having the x-axis showing error residual and y-axis to show density. The red line on the histogram shows the error density. The mean, variance, and standard deviation of error1 will be calculated for a blue normal distribution line to compare with the red line.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> plot(error1, main = 'Prediction Errors', xlab = 'Error1', ylab = 'Residual')

> abline(0,0)

> hist(error1, prob = T, breaks = 20, main = 'Error1- pred vs. test', xlab = 'Residual', ylab = 'Density')

> lines(density(error1), col='red')

> mean\_e <- mean(error1)

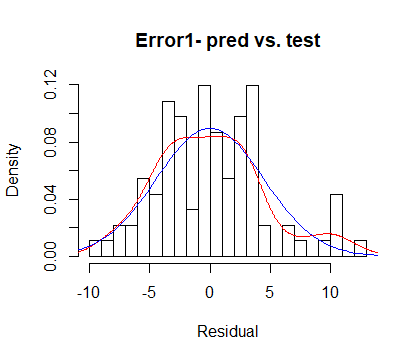
> var\_e <- var(error1)

> sd\_e <- sqrt(var\_e)

> x\_e <- seq(-20,20, length=92)

> y\_e <- dnorm(x\_e, mean\_e, sd\_e)

> lines(x\_e, y\_e, col = 'blue')



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Histogram looks like it's normally distributed overall. Breaks was increased from 10 to 20 to view the density (0.03) of error residuals between 5 and 6. Setting breaks at 10 showed the density (0.06) of error residuals between 4 and 6. A red line curve of the differences between predicted and actual error values was created on the plot. A normalized distribution curve was created to compare with the predicted curve using 92 x-values from a range of -20 to 20. The y-values were calculated using the 'dnorn()' function, which used the calculated mean and standard deviations (from variance). The red error curve is similar to blue normal predicted curve, but not accurately close – hence, it falls with the R-squared value of ~0.78.

**TRANSACTION LOG**

Following is the order for all the necessary code lines of this report. This log serves as a perfect example of the code lines of a project. The lines will have a description comment to save time in remembering or understanding. I've removed some lines to improve efficiency and only kept if essential for use. This log can also be used to efficiently review and make changes in the future.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> auto\_mpg <- read.csv(file.choose())

> View(auto\_mpg)

> str(auto\_mpg)

> summary(auto\_mpg)

> auto\_mpg1 <- auto\_mpg[-c(33,127,331,337,355,375),]

> auto\_mpg\_clean <- auto\_mpg1[,-c(9)]

> View(auto\_mpg\_clean)

> auto\_mpg\_clean$horsepower <- as.numeric(auto\_mpg\_clean$horsepower)

> model <- lm(mpg~., data = auto\_mpg)

> summary(model)

> model1 <- lm(mpg~., data = auto\_mpg\_clean)

> summary(model1)

> str(auto\_mpg\_clean)

> View(auto\_mpg\_clean)

> train <- auto\_mpg\_clean[1:300,]

> View(train)

> # test <- auto\_mpg\_clean[301:398,]

> test <- auto\_mpg\_clean[301:392,]

> View(test)

> traincylinder <- lm(train$mpg ~ train$cylinder, data = train)

> summary(traincylinder)

> plot(train$cylinder, train$mpg, main = "MPG vs. Cylinder", xlab = "cylinder", ylab = "mpg")

> traindisplacement <- lm(train$displacement ~ train$mpg, data = train)

> summary(traindisplacement)

> plot(train$displacement, train$mpg, main = "MPG vs. Displacement", xlab = "displacement", ylab = "mpg")

> trainhorsepower <- lm(train$horsepower ~ train$mpg, data = train)

> summary(trainhorsepower)

> median(train$horsepower)

> train$horsepower1 <- train$horsepower[train$horsepower < 47.5]

> trainhorsepower1 <- lm(train$horsepower1 ~ train$mpg, data = train)

> summary(trainhorsepower1)

> train$horsepower2 <- train$horsepower[train$horsepower > 47.5]

> trainhorsepower2 <- lm(train$horsepower ~ train$mpg, data = train)

> summary(trainhorsepower2)

> trainweight <- lm(train$weight ~ train$mpg, data = train)

> summary(trainweight)

> plot(train$weight, train$mpg, main = "MPG vs. Weight", xlab = "weight", ylab = "mpg")

> trainacceleration <- lm(train$acceleration ~ train$mpg, data = train)

> summary(trainacceleration)

> plot(train$acceleration, train$mpg, main = "MPG vs. Acceleration", xlab = "acceleration", ylab = "mpg")

> trainmodel.year <- lm(train$model.year ~ train$mpg, data = train)

> summary(trainmodel.year)

> plot(train$model.year, train$mpg, main = "MPG vs. Model Year", xlab = "model.year", ylab = "mpg")> trainorigin <- lm(train$origin ~ train$mpg, data = train)

> summary(trainorigin)

> plot(train$origin, train$mpg, main = "MPG vs. Origin", xlab = "origin", ylab = "mpg")

> attach(train)

> adjustedmodel <- lm(mpg ~ cylinder + displacement + weight, data = train)

> summary(adjustedmodel)

> attach(train)

> adjustedmodel2 <- lm(mpg ~ displacement + weight, data = train)

> summary(adjustedmodel2)

> adjustedmodel3 <- lm(mpg ~ cylinder + displacement + horsepower + weight + acceleration + model.year + origin, data = train)

> summary(adjustedmodel3)

> adjustedmodel4 <- lm(mpg ~ weight + model.year + origin, data = train)

> summary(adjustedmodel4)

> adjustedmodel5 <- lm(mpg ~ displacement + horsepower2 + weight, data = train)

> summary(adjustedmodel5)

> attach(train)

> adjustedmodel2 <- lm(mpg ~ displacement + weight, data = train)

> pred <- coef(trainweight)[1] + coef(trainweight)[2]\*test$mpg

> View(pred)

> test\_mpg <- test[,1]

> View(test\_mpg)

> error <- pred - test\_mpg

> summary(error)

> pred1 <- coef(adjustedmodel2)[1] + coef(adjustedmodel2)[2]\*test$displacement + coef(adjustedmodel2)[3]\*test$weight

> View(pred1)

> test\_mpg1 <- test[,1]

> error1 <- test\_mpg1 - pred1

> View(error1)

> summary(error1)

> error1 <- error1 - 6.764

> summary(error1)

> plot(error1, main = 'Prediction Errors', xlab = 'Error1', ylab = 'Residual')

> abline(0,0)

> plot(error1, main = 'Prediction Errors', xlab = 'Error1', ylab = 'Residual')

> abline(0,0)

> hist(error1, prob = T, breaks = 20, main = 'Error1- pred vs. test', xlab = 'Residual', ylab = 'Density')

> lines(density(error1), col='red')

> mean\_e <- mean(error1)

> var\_e <- var(error1)

> sd\_e <- sqrt(var\_e)

> x\_e <- seq(-20,20, length=92)

> y\_e <- dnorm(x\_e, mean\_e, sd\_e)

> lines(x\_e, y\_e, col = 'blue')

**BONUS**

The 'esquisse' package in R is new and includes a tableau-like drag and drop GUI visualization. It sits right inside RStudio and helps build ggplot2 package for visualizations. "It allows you to draw bar graphs, curves, scatter plots, histograms, then export the graph or retrieve the code generating the graph."

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

> source("https://install-github.me/dreamRs/esquisse")

> ggplot(data = test) +

+ aes() +

+ geom\_blank() +

+ geom\_smooth(span = 1) +

+ labs(title = "MPG vs Weight, Displacement",

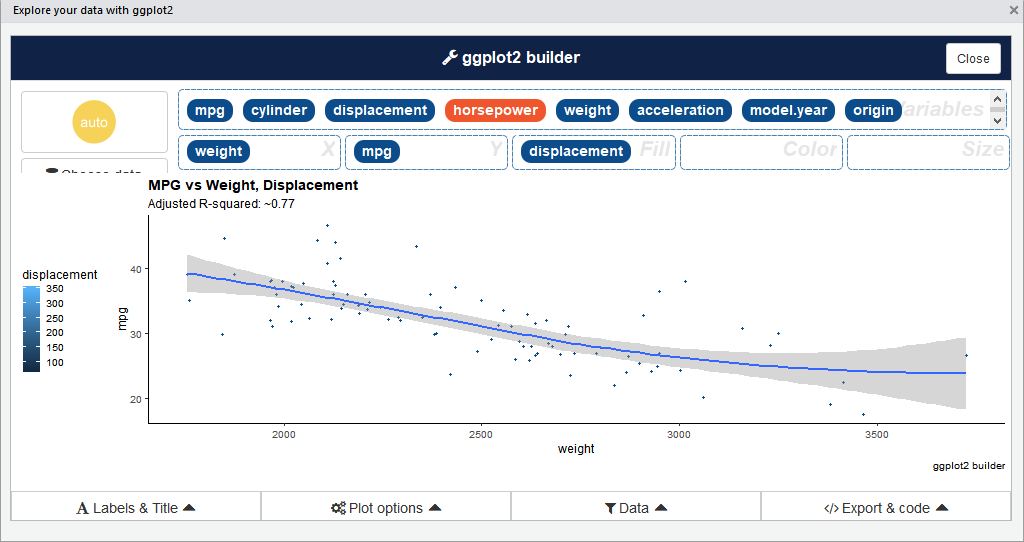
+ x = "weight",

+ y = "mpg",

+ caption = "ggplot2 builder",

+ subtitle = "Adjusted R-squared: 0.77") +

+ theme\_classic()



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The 'esquisse' package pulled open a 'ggplot2 builder' window and allowed to choose a dataframe and which variables to create a graph out of. It also allowed to change settings of the graph. Then we can insert the code in script.

**CONCLUSION**

In conclusion, we analyzed two variables from the 'auto-mpg' dataset to be significant factors for the vehicles studied in this dataset. The car names were not used for analysis, but all the other variables were used since they were numeric and the same type as mpg. We analyzed each variable individually to test its significance and multiple R-squared. R-squared value tells us how dependable it is. All the variables had \*\*\* or 100% significance, but many had a low R-squared so their high variance was too risky to include when adjusting. Although some variable can have observations that can cause a significant change of R-squared, it's multiple R-squared being very low proves its high variance and can be a gamble to include. The variables were then adjusted to find the best fit with the highest adjusted R-squared value. The significant variables -cylinder, displacement, and weight- were combined in a linear regression. Since only displacement and weight showed significance, \* and \*\*\* respectively, the complete equation of linear regression was found using those two variables.

Then, the best linear model obtained from the training dataset was used to predict each test automobile's mpg. The predictions were compared to the car's actual reported mpg values. The error residuals were also mostly positive values, but were normalized to 0. They were within the upper and lower boundaries after calculating IQR. The predicted error difference curve was compared to the actual reported mpg's predicted distribution curve, and they were similar indicating that they were normal. The highest R-squared I got after adjusting the parameters was ~0.78, which I can believe to be good for the subject of cars. Since data from cars can vary extensively, I chose the factors that showed the R-squared >0.6 because the lower R-squared variables vary greatly to trust.

This was a great exercise to use this automobile dataset to analyze and adjust to reach the best outcome. While doing so, I learned about the effects of the different factors on mpg and how to adjust the parameters. Appropriate car companies can use such methods to modify their processes to reach their desired outcomes.

Thank you for taking the time to read this report!

---

Hiding within those mounds of data is knowledge that could change the life of a patient, or change the world.

**Table of Contents:**

1: Introduction ………..………………………………………………………………………..3

2: Read, View, Structure, Summary …..…………………………………………………...4

3: Clean Data ……………………………………….……………………………………...…6

4: Train Dataset & Test Dataset ………………………………….…………………………8

5: Multiple R-Squared ………………………………………………………………..……….9

6: Adjusted R-Squared …………………………………………………………………...…23

7: Complete Linear Regression Equation ……………………………………………….26

8: Predict & Compare ………………………………………………………………………27

9: Residual Plot ……………………………………………………………………………….29

10: Histogram …………………………………………………………………………...……30

11: Transaction Log ……………………………………………………………………….…32

12. Bonus ………………………………………………………………………………………35

13: Conclusion ………………………………………………………………………………..37

**INTRODUCTION**

This project is to investigate the impact of a number of automobile engine factors on the vehicle’s mpg. I will walk you through the process of analyzing and interpreting all the steps. I used the dataset "auto-mpg.csv" that contains information for 398 different automobile models. Variables analyzed: mpg, cylinder, displacement, horsepower, weight, acceleration, model.year, origin, and car.name.

Firstly, using 300 observations in 'auto-mpg.csv' as train dataset, I will run linear regression individually to determine the relationship between mpg and appropriate variables. As I analyze the data and the results with the train dataset, I will adjust the parameters to get the intended/best result. The remaining observations of the dataset will be in test dataset and will be analyzed using a complete linear regression equation gotten from the train dataset. All appropriate results, a log, and comments will be included in this report to better understand the methods and the journey.

**READ, VIEW, STRUCTURE, SUMMARY**

Firstly, I read the "auto-mpg" file using 'read.csv(file(choose))' function to get an access to the data for this project. I can also use 'read.csv("/Users/...path.../auto-mpg.csv")', but I'll use the first method here. I view it using a 'View()' function to get a picture for the data. I then get a statistical summary using 'summary()' function to better understand. I also like to use the 'str()' function to know the basic details of its structure in the start.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

auto\_mpg <- read.csv(file.choose())

# auto\_mpg\_read <- read.csv("/Users/apatel8/Documents/DS 510/Labs/auto-mpg.csv")

View(auto\_mpg)

str(auto\_mpg)

summary(auto\_mpg)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

'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 : Factor w/ 94 levels "?","100","102",..: 17 35 29 29 24 42 47 46 48 40 ...

$ 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 : Factor w/ 305 levels "amc ambassador brougham",..: 50 37 232 15 162 142 55 224 242 2 ...

mpg cylinder displacement horsepower weight acceleration model.year

Min. : 9.00 Min. :3.000 Min. : 68.0 150 : 22 Min. :1613 Min. : 8.00 Min. :70.00

1st Qu.:17.50 1st Qu.:4.000 1st Qu.:104.2 90 : 20 1st Qu.:2224 1st Qu.:13.82 1st Qu.:73.00

Median :23.00 Median :4.000 Median :148.5 88 : 19 Median :2804 Median :15.50 Median :76.00

Mean :23.51 Mean :5.455 Mean :193.4 110 : 18 Mean :2970 Mean :15.57 Mean :76.01

3rd Qu.:29.00 3rd Qu.:8.000 3rd Qu.:262.0 100 : 17 3rd Qu.:3608 3rd Qu.:17.18 3rd Qu.:79.00

Max. :46.60 Max. :8.000 Max. :455.0 75 : 14 Max. :5140 Max. :24.80 Max. :82.00

(Other):288

origin car.name

Min. :1.000 ford pinto : 6

1st Qu.:1.000 amc matador : 5

Median :1.000 ford maverick : 5

Mean :1.573 toyota corolla: 5

3rd Qu.:2.000 amc gremlin : 4

Max. :3.000 amc hornet : 4

(Other) :369 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This dataset has 9 variables (mpg is 1; other 8 variables will be analyzed to study its feature). The types for this data are num, int, and Factor. The reason for horsepower being a factor is that it includes some '?'. The variables we will use for correlation between mpg and other variables should to be numeric, the same type as mpg. The car.name variable values are non-numeric, and thus will not be included in the analysis.

This summary gives us the basic statistics such as Min, 1st Qu, Median, Mean, 3rd Qu, and Max for each variable. We should glance through it to get a good perspective. Statistics of mpg variable: IQR = 29.00 - 17.50 = 11.5; mpg dataset is well-within lower and upper boundaries because they are within 1.5\*IQR range. As I analyze each variable individually, I will comment on what I observe.

**CLEAN DATA**

Non-numeric variables need their types changed to numeric to avoid errors. We will change the 'horsepower' variable (factor type) to numeric. The 'horsepower' variable column also contains some '?' values, whose rows will be excluded in new dataset. I found the row numbers by putting the observations in 'horsepower' in ascending order, and that grouped the ‘?’ values together on top. The variable of car.name with non-numeric values will also be excluded. I'll get a summary of the linear regression on the full dataset in the start by using 'mpg~.' (. means all columns).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

auto\_mpg1 <- auto\_mpg[-c(33,127,331,337,355,375),]

auto\_mpg\_clean <- auto\_mpg1[,-c(9)]

View(auto\_mpg\_clean)

auto\_mpg\_clean$horsepower <- as.numeric(auto\_mpg\_clean$horsepower)

model <- lm(mpg~., data = auto\_mpg)

summary(model)

model1 <- lm(mpg~., data = auto\_mpg\_clean)

summary(model1)

str(auto\_mpg\_clean)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = mpg ~ ., data = auto\_mpg\_clean)

Residuals:

Min 1Q Median 3Q Max

-9.656 -2.069 -0.043 1.775 13.098

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -2.134e+01 4.292e+00 -4.971 1.00e-06 \*\*\*

cylinder -2.739e-01 3.398e-01 -0.806 0.4208

displacement 1.539e-02 7.275e-03 2.116 0.0350 \*

horsepower 1.072e-02 7.076e-03 1.515 0.1306

weight -6.756e-03 5.828e-04 -11.592 < 2e-16 \*\*\*

acceleration 1.489e-01 7.777e-02 1.914 0.0563 .

model.year 7.688e-01 4.972e-02 15.463 < 2e-16 \*\*\*

origin 1.344e+00 2.701e-01 4.975 9.89e-07 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.324 on 384 degrees of freedom

Multiple R-squared: 0.8218, Adjusted R-squared: 0.8186

F-statistic: 253.1 on 7 and 384 DF, p-value: < 2.2e-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The Multiple R-squared for the mpg vs. entire dataset was ~0.99. Without the color.name variable, it was ~0.90. We will not include them in our regression because we want to find a good correlation with a good R-squared value by using only the numeric values.

We will clean again after we run correlation for the variables individually to determine which ones to use according to their significance and R-squared.

**TRAIN DATASET & TEST DATASET**

Divide the "auto-mpg" dataset into 2 datasets: train dataset of 300 observations and test dataset of 301 to 392 observations. The test dataset was initially 301 to 398 observations, but I reduced it since we removed the 6 rows with ‘?’ values for ‘horsepower’. \*\*Could stores the randomly generated values but that will change the values every time, so not ideal for this report.\*\*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

View(auto\_mpg\_clean)

train <- auto\_mpg\_clean[1:300,]

View(train)

# test <- auto\_mpg\_clean[301:398,]

test <- auto\_mpg\_clean[301:392,]

View(test)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Again, the train dataset will be used to determine a complete linear regression equation that will be used for the test dataset.

**MULTIPLE R-SQUARED:**

MPG vs. Cylinder

Scatterplot to find the relationship between 'mpg' and 'cylinder' in the training set.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

traincylinder <- lm(train$mpg ~ train$cylinder, data = train)

summary(traincylinder)

plot(train$cylinder, train$mpg, main = "MPG vs. Cylinder", xlab = "cylinder", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = train$mpg ~ train$cylinder, data = train)

Residuals:

Min 1Q Median 3Q Max

-11.1071 -2.3012 -0.4306 1.8282 16.9282

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 37.9130 0.7356 51.54 <2e-16 \*\*\*

train$cylinder -2.9353 0.1211 -24.24 <2e-16 \*\*\*

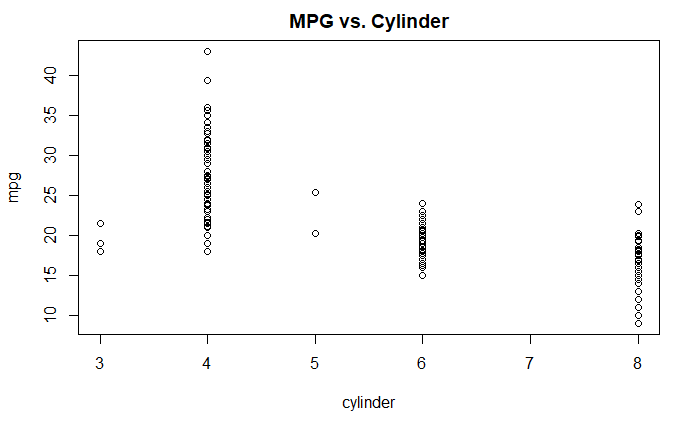
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.678 on 298 degrees of freedom

Multiple R-squared: 0.6636, Adjusted R-squared: 0.6624

F-statistic: 587.8 on 1 and 298 DF, p-value: < 2.2e-16



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This scatterplot shows us the distribution of cylinders. 4 cylinders have a higher range of mpg as well as higher values of mpg, containing the maximum of 46. 8 cylinders has some of the lowest mpg observations, containing the minimum of 9. 6 cylinders subset group seem to be around the mean ‘mpg’ of the full dataset (‘auto\_mpg’) of 23.51.

The linear model on a train dataset of 300 observations gave us a \*\*\* significance of <2e-16 p-value and ~0.66 Multiple R-squared. The \*\*\* tells us that it has 100% significance, and the extremely low p-value being close to 0 shows that it is sufficiently consistent. The coefficient of determination, denoted as r^2 and pronounced “R-squared”, is the proportion of the variance in the dependable variable (mpg) that is predictable from the independent variable (cylinder); a multiple R-squared of 0.66 will be used to adjust, in case it gives us better results. The Adjusted R-squared value is also ~0.66 because it hasn't been adjusted yet.

MPG vs. Displacement

Scatterplot to find the relationship between 'mpg' and 'displacement' in the training set. From the table, 'displacement' values seem to decrease as mpg values increase.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

traindisplacement <- lm(train$displacement ~ train$mpg, data = train)

summary(traindisplacement)

plot(train$displacement, train$mpg, main = "MPG vs. Displacement", xlab = "displacement", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = train$displacement ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-186.167 -35.216 -2.498 33.558 198.391

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 517.6038 11.6978 44.25 <2e-16 \*\*\*

train$mpg -14.5242 0.5372 -27.04 <2e-16 \*\*\*

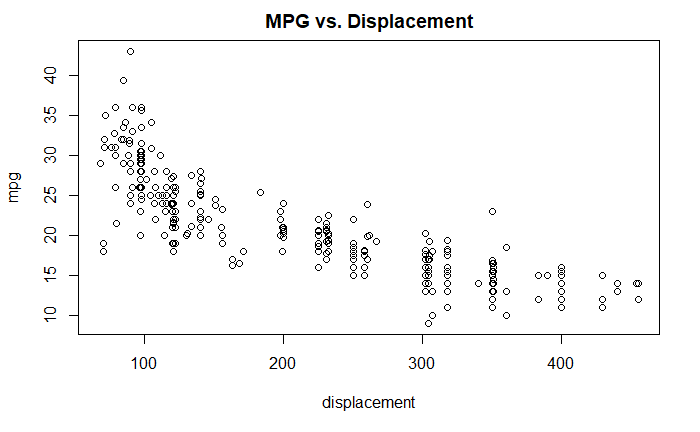
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 58.8 on 298 degrees of freedom

Multiple R-squared: 0.7104, Adjusted R-squared: 0.7094

F-statistic: 731.1 on 1 and 298 DF, p-value: < 2.2e-16



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This scatterplot shows us a linear negative-slope trend in 'displacement' when comparing mpg. It does have a variance, especially at lower 'displacement'. The results show its significance of \*\*\* with a p-value of 2.2e-16 and Multiple R-squared of ~0.71. It will be included in the adjusted regression.

MPG vs. Horsepower

Scatterplot to see the relationship between 'mpg' and 'horsepower' in the training dataset. I would assume that it would be significant.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

trainhorsepower <- lm(train$horsepower ~ train$mpg, data = train)

summary(trainhorsepower)

plot(train$horsepower, train$mpg, main = "MPG vs. Displacement", xlab = "displacement", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = train$horsepower ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-50.697 -18.444 -0.606 20.316 50.798

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.5888 5.2436 0.112 0.911

train$mpg 2.3119 0.2408 9.601 <2e-16 \*\*\*

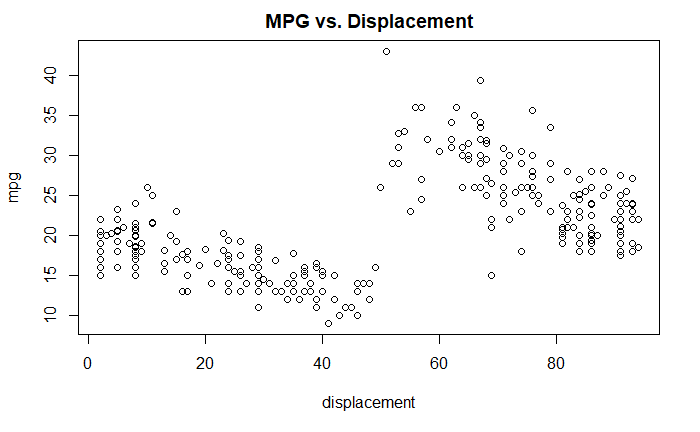
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26.36 on 298 degrees of freedom

Multiple R-squared: 0.2363, Adjusted R-squared: 0.2337

F-statistic: 92.18 on 1 and 298 DF, p-value: < 2.2e-16



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This is why I did not choose my career in cars. Although it shows a significance code of \*\*\*, since it's Multiple R-squared values are ~0.23, it is too variable and they will not be included in the adjusted training set.

MPG vs. Weight

Scatterplot to see the relationship between 'mpg' and 'weight' in the training dataset. Again, I would assume that it would be significant.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

trainweight <- lm(train$weight ~ train$mpg, data = train)

summary(trainweight)

plot(train$weight, train$mpg, main = "MPG vs. Weight", xlab = "weight", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = train$weight ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-1359.51 -273.50 -0.02 264.01 1579.43

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 5690.791 84.216 67.57 <2e-16 \*\*\*

train$mpg -122.627 3.867 -31.71 <2e-16 \*\*\*

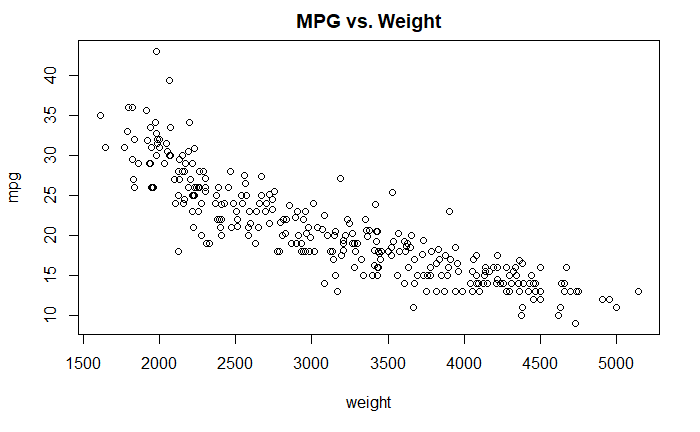
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 423.3 on 298 degrees of freedom

Multiple R-squared: 0.7714, Adjusted R-squared: 0.7706

F-statistic: 1005 on 1 and 298 DF, p-value: < 2.2e-16



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Voila, the highest significance so far. The results show \*\*\* significance of <2.2e-16 p-value and ~0.77 Multiple R-squared. Weight variable will be included in the adjusted regression.

MPG vs. Acceleration

Scatterplot to see the relationship between mpg and acceleration in the training dataset. Again, I would assume for this to be significant, but since the horsepower was insignificant, I wouldn't put my money on this.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

trainacceleration <- lm(train$acceleration ~ train$mpg, data = train)

summary(trainacceleration)

plot(train$acceleration, train$mpg, main = "MPG vs. Acceleration", xlab = "acceleration", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = train$acceleration ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-5.9084 -1.7107 -0.2921 1.4030 8.2824

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 11.14106 0.49082 22.70 <2e-16 \*\*\*

train$mpg 0.19767 0.02254 8.77 <2e-16 \*\*\*

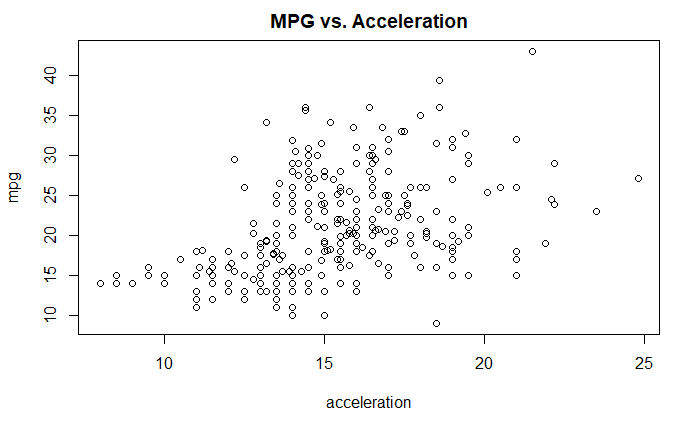
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.467 on 298 degrees of freedom

Multiple R-squared: 0.2052, Adjusted R-squared: 0.2025

F-statistic: 76.91 on 1 and 298 DF, p-value: < 2.2e-16



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

It looks like I saved my bet. Although the coefficients repeatedly have \*\*\* significance and p-value of <2.2e-16, the Multiple R-squared of ~0.2 is too low to be significant.

MPG vs. Model Car

Scatterplot to see the relationship between 'mpg' and 'model.car' in the training dataset. I wouldn't think it would be much significant, but I would assume that there was a continuous slow increase during that time period.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

trainmodel.year <- lm(train$model.year ~ train$mpg, data = train)

summary(trainmodel.year)

plot(train$model.year, train$mpg, main = "MPG vs. Model Year", xlab = "model.year", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = train$model.year ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-5.3228 -1.9428 0.1113 2.1002 5.2194

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 71.75982 0.53199 134.890 < 2e-16 \*\*\*

train$mpg 0.13037 0.02443 5.337 1.88e-07 \*\*\*

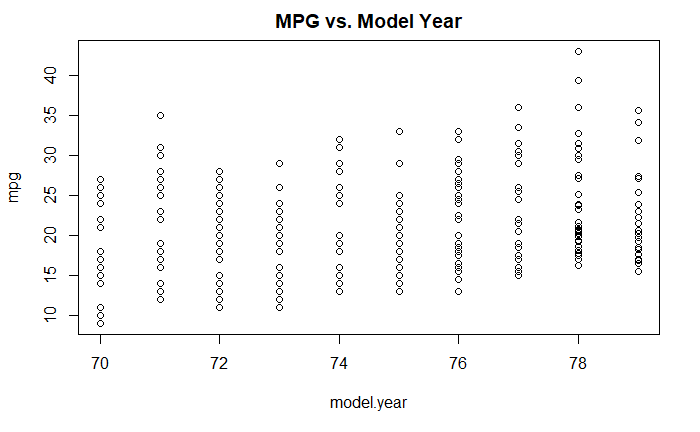
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.674 on 298 degrees of freedom

Multiple R-squared: 0.08723, Adjusted R-squared: 0.08417

F-statistic: 28.48 on 1 and 298 DF, p-value: 1.878e-07



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Somewhat correct assumption. It can be seen as a tiny increasing wave, but it's variance is tremendously high. That can be proved by the Multiple R-squared value of ~0.08. Thus, it will not be included in the adjusted regression.

MPG vs. Origin

Scatterplot to see the relationship between 'mpg' and 'origin' in the training dataset.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

trainorigin <- lm(train$origin ~ train$mpg, data = train)

summary(trainorigin)

plot(train$origin, train$mpg, main = "MPG vs. Origin", xlab = "origin", ylab = "mpg")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = train$origin ~ train$mpg, data = train)

Residuals:

Min 1Q Median 3Q Max

-1.52042 -0.34713 -0.07268 0.20178 1.72148

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.043476 0.118852 0.366 0.715

train$mpg 0.068613 0.005458 12.572 <2e-16 \*\*\*

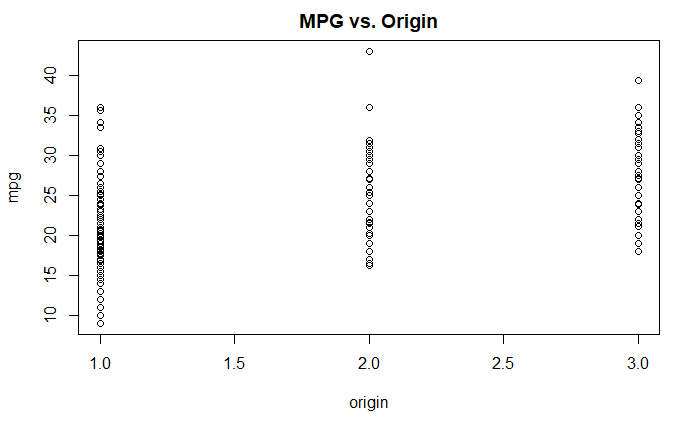
---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5975 on 298 degrees of freedom

Multiple R-squared: 0.3466, Adjusted R-squared: 0.3444

F-statistic: 158 on 1 and 298 DF, p-value: < 2.2e-16



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Last but not the least, the origin variable is insignificant because of its Multiple R-squared value of ~0.35, and it will not be included.

**ADJUSTED R-SQUARED**

Now, the variables will be combined in different ways to find the highest R-squared. What is R-squared and why is R-squared itself significant? R-squared simply explains the variability of a model; that is, how our dependent variable varies in proportion to the independent variable. We would love to get a R-squared value of 1.0 (or 100%), but there will always be variance caused by several factors. Here, we attempt to find the highest R-squared value by trying different combinations between numeric variables and "adjusting" the R-squared value.

Firsly, let's try and combine every variable with a R-squared of >0.60. The following variables had high enough significance to try: 'cylinder' (~0.66), 'displacement' (~0.71), and 'weight' (~0.77).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

attach(train)

adjustedmodel <- lm(mpg ~ cylinder + displacement + weight, data = train)

summary(adjustedmodel)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = mpg ~ cylinder + displacement + weight, data = train)

Residuals:

Min 1Q Median 3Q Max

-9.559 -1.892 -0.151 1.712 15.113

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 39.0682163 1.1448111 34.126 <2e-16 \*\*\*

cylinder -0.0579278 0.3260901 -0.178 0.8591

displacement -0.0103839 0.0062078 -1.673 0.0954 .

weight -0.0049947 0.0005411 -9.231 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.009 on 296 degrees of freedom

Multiple R-squared: 0.7764, Adjusted R-squared: 0.7741

F-statistic: 342.6 on 3 and 296 DF, p-value: < 2.2e-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The cylinder variable had a 0.8591 p-value and a ' ' significance code, thus it has no significance. Let's try with just 'displacement' and 'weight'.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

attach(train)

adjustedmodel2 <- lm(mpg ~ displacement + weight, data = train)

summary(adjustedmodel2)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The following objects are masked from train (pos = 3):

acceleration, cylinder, displacement, horsepower, model.year, mpg, origin, weight

Call:

lm(formula = mpg ~ displacement + weight, data = train)

Residuals:

Min 1Q Median 3Q Max

-9.5232 -1.8741 -0.1765 1.6766 15.1041

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 38.9422109 0.8971239 43.408 <2e-16 \*\*\*

displacement -0.0111720 0.0043353 -2.577 0.0104 \*

weight -0.0050080 0.0005351 -9.360 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.004 on 297 degrees of freedom

Multiple R-squared: 0.7764, Adjusted R-squared: 0.7749

F-statistic: 515.6 on 2 and 297 DF, p-value: < 2.2e-16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The adjusted regression between mpg vs. displacement and weight gives a Multiple and Adjusted Regression of ~0.78 each with displacement being \* (95%) significant and weight being \*\*\* (100%) significant. We will run the test using this model.

The following was not part of the plan, but I will run linear regression by combining more factors to see their effect and their significance.

It shows the “following objects are masked from train” message because both the packages have functions with the same name. To avoid it, I can either ‘detach()’ or not ‘attach()’ again.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

adjustedmodel3 <- lm(mpg ~ cylinder + displacement + horsepower + weight + acceleration, data = train)

summary(adjustedmodel3)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Call:

lm(formula = mpg ~ cylinder + displacement + horsepower + weight +

acceleration, data = train)

Residuals:

Min 1Q Median 3Q Max

-9.3002 -1.8905 -0.0339 1.6557 14.8272

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 37.0518989 1.8963492 19.539 <2e-16 \*\*\*

cylinder 0.0107676 0.3383601 0.032 0.975

displacement -0.0085758 0.0068050 -1.260 0.209

horsepower 0.0063642 0.0070540 0.902 0.368

weight -0.0050961 0.0005701 -8.939 <2e-16 \*\*\*

acceleration 0.0809730 0.0815218 0.993 0.321

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Like expected, the other variables like horsepower and acceleration, which one would assume would have a big effect, are not significant. That is because those factors cause high variance in the results (which I would assume they do), so they are insignificant for a regression model. The Multiple and Adjusted R-Squared values remain to be ~0.78 because they are not significant. The displacement doesn't show any significance either this time but since it shows a significance with weight alone, I will include it in the test function.

**COMPLETE LINEAR REGRESSION EQUATION**

The adjustedmodel2 equation will be used as the complete linear regression equation for the test dataset.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

attach(train)

adjustedmodel2 <- lm(mpg ~ displacement + weight, data = train)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PREDICT & COMPARE**

For the remaining 92 samples in the dataset, we will use the best linear models (mpg vs. displacement+weight, mpg vs. weight) to predict each automobile’s mpg. We will report how the predictions compare to the car’s actual reported mpg.

Let's first predict values using weight variable only. We will use the ‘coef()’ function to get the B0 and B1 values from the linear equation and use that to predict with the test observations. We will then compare to the actual values by calculating the difference.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pred <- coef(trainweight)[1] + coef(trainweight)[2]\*test$mpg

View(pred)

test\_mpg <- test[,1]

View(test\_mpg)

error <- pred - test\_mpg

summary(error)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Min. 1st Qu. Median Mean 3rd Qu. Max.

-70.22 1240.22 1728.55 1741.45 2263.24 3514.96

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The error was extremely high and ranged from ~(-70) to ~3500. Now, let's try to predict using the 'displacement' and 'weight' variables. We will find the coefficients in that adjusted model, then define them individually.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

pred1 <- coef(adjustedmodel2)[1] + coef(adjustedmodel2)[2]\*test$displacement + coef(adjustedmodel2)[3]\*test$weight

View(pred1)

test\_mpg1 <- test[,1]

error1 <- test\_mpg1 - pred1

View(error1)

summary(error1)

error1 <- error1 - 6.764

summary(error1)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Min. 1st Qu. Median Mean 3rd Qu. Max.

-9.10483 -3.24530 -0.33306 -0.00035 2.54028 12.42144

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The predicted values ranged from ~16 to ~29 according to linear regression using displacement and weight variables. Error1 ranged from ~(-2) to ~19 and its mean was 6.764. To normalize that, I subtracted 6.764 from error1. The error1's IQR = ~5.8, and the Max value of error and some other higher values are outside the upper boundary of ~12. After normalizing it to the mean of 0, the errors are within the boundaries. It indicates that the data points are in the normal curve.

**RESIDUAL PLOT**

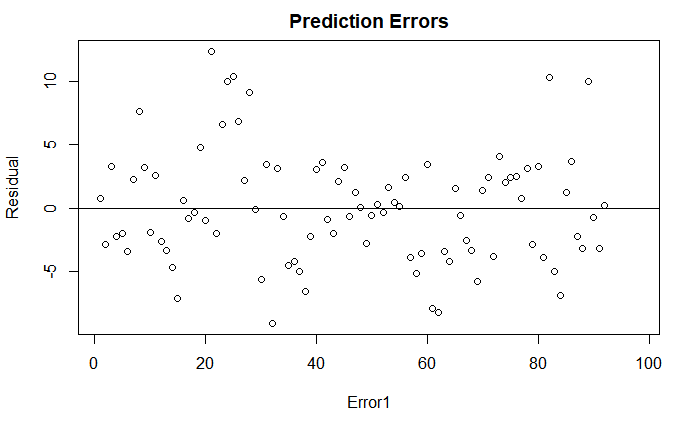
Following is a plot of the error differences between predicted and actual values. Let's also plot a line with intercept and slope at 0 for a clear view.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

plot(error1, main = 'Prediction Errors', xlab = 'Error1', ylab = 'Residual')

abline(0,0)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The plot distributed nicely after normalizing the residuals to 0. At first, it was mainly positive with only some negative values.

**HISTOGRAM**

We will also create a histogram with the error1 data, having the x-axis showing error residual and y-axis to show density. The red line on the histogram shows the error density. The mean, variance, and standard deviation of error1 will be calculated for a blue normal distribution line to compare with the red line.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

hist(error1, prob = T, breaks = 20, main = 'Error1- pred vs. test', xlab = 'Residual', ylab = 'Density')

lines(density(error1), col='red')

mean\_e <- mean(error1)

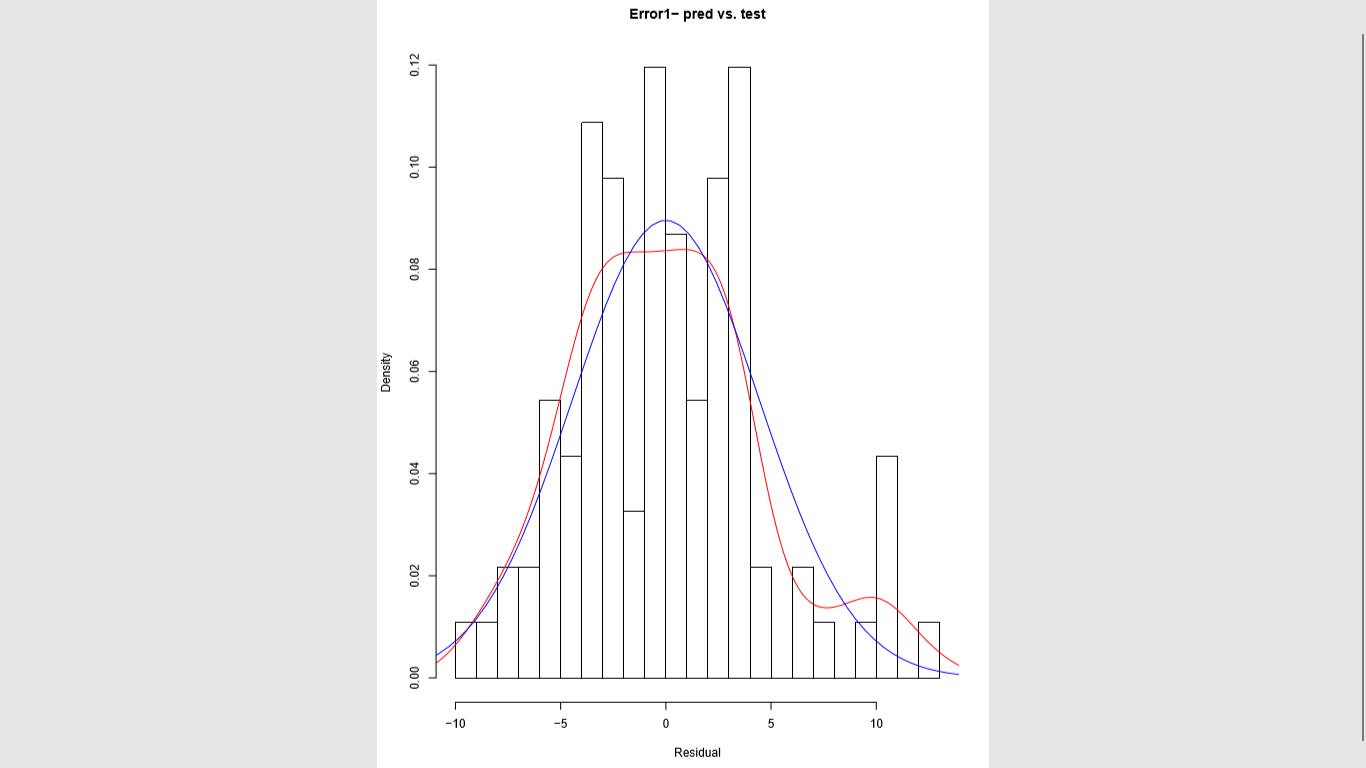
var\_e <- var(error1)

sd\_e <- sqrt(var\_e)

x\_e <- seq(-20,20, length=92)

y\_e <- dnorm(x\_e, mean\_e, sd\_e)

lines(x\_e, y\_e, col = 'blue')



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Histogram looks like it's normally distributed overall. Breaks was increased from 10 to 20 to view the density (0.03) of error residuals between 5 and 6. Setting breaks at 10 showed the density (0.06) of error residuals between 4 and 6. A red line curve of the differences between predicted and actual error values was created on the plot. A normalized distribution curve was created to compare with the predicted curve using 92 x-values from a range of -20 to 20. The y-values were calculated using the 'dnorn()' function, which used the calculated mean and standard deviations (from variance). The red error curve is similar to blue normal predicted curve, but not accurately close – hence, it falls with the R-squared value of ~0.78.

**TRANSACTION LOG**

Following is the order for all the necessary code lines of this report. This log serves as a perfect example of the code lines of a project. The lines will have a description comment to save time in remembering or understanding. I've removed some lines to improve efficiency and only kept if essential for use. This log can also be used to efficiently review and make changes in the future.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

auto\_mpg <- read.csv(file.choose()) #read

auto\_mpg\_read <- read.csv("/Users/apatel8/Documents/DS 510/Labs/auto-mpg.csv") #2nd read option

View(auto\_mpg) #view dataset

str(auto\_mpg) #structure

summary(auto\_mpg) #statistics

auto\_mpg1 <- auto\_mpg[-c(33,127,331,337,355,375),] #remove ?

auto\_mpg\_clean <- auto\_mpg1[,-c(9)] #remove car.name

auto\_mpg\_clean$horsepower <- as.numeric(auto\_mpg\_clean$horsepower) #horsepower numeric

model <- lm(mpg~., data = auto\_mpg) #R-squared full dataset

train <- auto\_mpg\_clean[1:300,] #300 train dataset

test <- auto\_mpg\_clean[301:398,] #98 test dataset

traincylinder <- lm(train$mpg ~ train$cylinder, data = train) #train cylinder

summary(traincylinder) #significant

traindisplacement <- lm(train$displacement ~ train$mpg, data = train) #train displacement

summary(traindisplacement) #significant

trainhorsepower <- lm(train$horsepower ~ train$mpg, data = train) #train horsepower

summary(trainhorsepower) #insignificant

trainweight <- lm(train$weight ~ train$mpg, data = train) #train weight

summary(trainweight) #significant

trainacceleration <- lm(train$acceleration ~ train$mpg, data = train) #train acceleration

summary(trainacceleration) #insignificant

trainmodel.year <- lm(train$model.year ~ train$mpg, data = train) #train model.year

summary(trainmodel.year) #insignificant

trainorigin <- lm(train$origin ~ train$mpg, data = train) #train origin

summary(trainorigin) #insignificant

attach(train) #attach for chunk

adjustedmodel <- lm(mpg ~ cylinder + displacement + weight, data = train) #adjusted

summary(adjustedmodel) #significance

adjustedmodel2 <- lm(mpg ~ displacement + weight, data = train) #adjusted

summary(adjustedmodel2) #significant

adjustedmodel3 <- lm(mpg ~ cylinder + displacement + horsepower + weight + acceleration, data=train) #check

summary(adjustedmodel3) #insignificant

pred <- coef(trainweight)[1] + coef(trainweight)[2]\*test$mpg #predict from weight

test\_mpg <- test[,1] #test mpg set

error <- pred - test\_mpg #error difference

pred1 <- coef(adjustedmodel2)[1] + coef(adjustedmodel2)[2]\*test$displacement + coef(adjustedmodel2)[3]\*test$weight #predict from displacement, weight

test\_mpg1 <- test[,1] #test mpg1 set

error1 <- test\_mpg1 - pred1 #error difference

error1 <- error1 - 6.764 #normalize

plot(error1, xlab = 'Error1', ylab = 'Residual')

abline(0,0) #0 line for plot

hist(error1, prob = T, breaks = 20, main = 'Error1- mpg vs. displacement+weight', xlab = 'Residual', ylab = 'Density')

lines(density(error1), col='red')

mean\_e <- mean(error1)

var\_e <- var(error1)

sd\_e <- sqrt(var\_e)

x\_e <- seq(-20,20, length=20)

y\_e <- dnorm(x\_e, mean\_e, sd\_e)

lines(x\_e, y\_e, col = 'blue')

**BONUS**

The 'esquisse' package in R is new and includes a tableau-like drag and drop GUI visualization. It sits right inside RStudio and helps build ggplot2 package for visualizations. "It allows you to draw bar graphs, curves, scatter plots, histograms, then export the graph or retrieve the code generating the graph."

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

source("https://install-github.me/dreamRs/esquisse")

esquisse::esquisser()

ggplot(data = test) +

aes() +

geom\_blank() +

geom\_smooth(span = 1) +

labs(title = "MPG vs Weight, Displacement",

x = "weight",

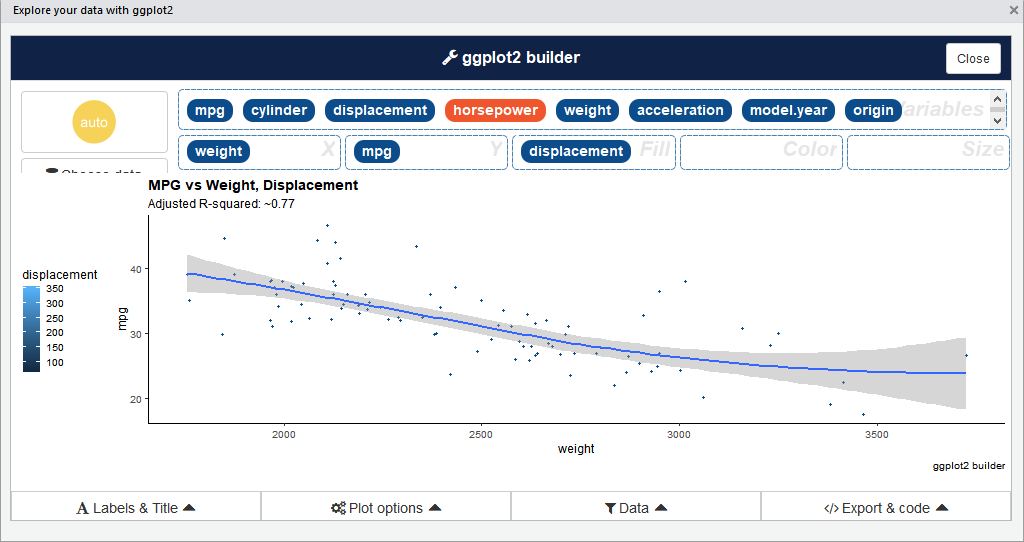
y = "mpg",

caption = "ggplot2 builder",

subtitle = "Adjusted R-squared: 0.77") +

theme\_classic()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The 'esquisse' package pulled open a 'ggplot2 builder' window and allowed to choose a dataframe and which variables to create a graph out of. It also allowed to change settings of the graph. Then we can insert the code in script.

~I can't see the plot printed, but it can be seen in the 'ggplot2 builder' as mentioned earlier.

**CONCLUSION**

In conclusion, we analyzed two variables from the 'auto-mpg' dataset to be significant factors for the vehicles studied in this dataset. The car names were not used for analysis, but all the other variables were used since they were numeric and the same type as mpg. We analyzed each variable individually to test its significance and multiple R-squared. R-squared value tells us how dependable it is. All the variables had \*\*\* or 100% significance, but many had a low R-squared so their high variance was too risky to include when adjusting. Although some variable can have observations that can cause a significant change of R-squared, it's multiple R-squared being very low proves its high variance and can be a gamble to include. The variables were then adjusted to find the best fit with the highest adjusted R-squared value. The significant variables -cylinder, displacement, and weight- were combined in a linear regression. Since only displacement and weight showed significance, \* and \*\*\* respectively, the complete equation of linear regression was found using those two variables.

Then, the best linear model obtained from the training dataset was used to predict each test automobile's mpg. The predictions were compared to the car's actual reported mpg values. The error residuals were also mostly positive values, but were normalized to 0. They were within the upper and lower boundaries after calculating IQR. The predicted error difference curve was compared to the actual reported mpg's predicted distribution curve, and they were similar indicating that they were normal. The highest R-squared I got after adjusting the parameters was ~0.78, which I can believe to be good for the subject of cars. Since data from cars can vary extensively, I chose the factors that showed the R-squared >0.6 because the lower R-squared variables vary greatly to trust.

This was a great exercise to use this automobile dataset to analyze and adjust to reach the best outcome. While doing so, I learned about the effects of the different factors on mpg and how to adjust the parameters. Appropriate car companies can use such methods to modify their processes to reach their desired outcomes.

Thank you for taking the time to read this report!

---

Hiding within those mounds of data is knowledge that could change the life of a patient, or change the world.