Data 605 - DB12

Avery Davidowitz

2022-11-10

Introduction

mtcars is a built in dataset consisting of: A data frame with 32 observations on 11 (numeric) variables.

[, 1] mpg Miles/(US) gallon [, 2] cyl Number of cylinders [, 3] disp Displacement (cu.in.) [, 4] hp Gross horsepower [, 5] drat Rear axle ratio [, 6] wt Weight (1000 lbs) [, 7] qsec 1/4 mile time [, 8] vs Engine (0 = V-shaped, 1 = straight) [, 9] am Transmission (0 = automatic, 1 = manual) [,10] gear Number of forward gears [,11] carb Number of carburetors

I will attempt to build a multiple linear regression model to predict an unknown car's miles per gallon.

Load Data

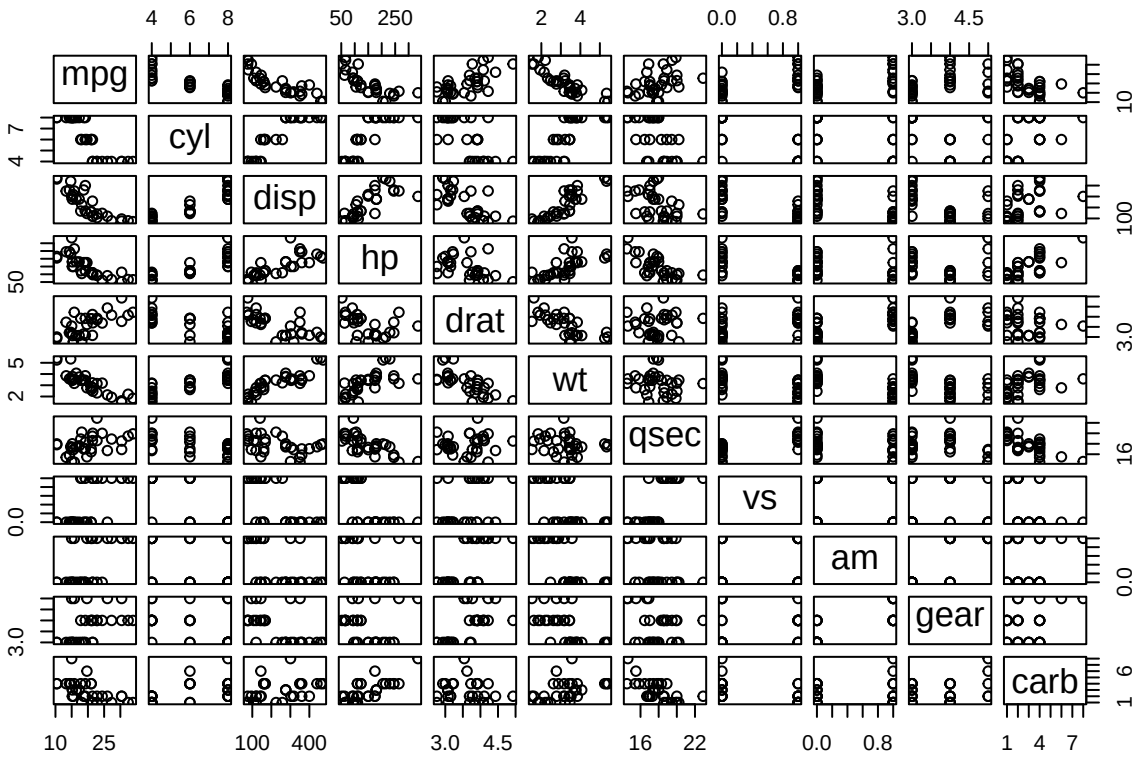
```
data(mtcars)
car_df <- mtcars
head(car_df)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
set.seed(1237)
```

Visualize Data Relationships

```
pairs(mtcars, gap=.5)
```



Divide Dataset into Train and Test Goups

```
rows <- nrow(car_df)
f <- .5
upper_bound <- floor(f * rows)
permuted_car_df <- car_df[sample(rows), ]
train <- permuted_car_df[1:upper_bound, ]
test <- permuted_car_df[(upper_bound+1):rows, ]
```

Backward Elimination of Variables

```
car_lm_full <- lm(mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb, data=train)
summary(car_lm_full)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + drat + wt + qsec + vs +
##     am + gear + carb, data = train)
##
## Residuals:
##           Valiant         Mazda RX4     Toyota Corona        Duster 360
```

```
##           -1.8051            1.1241            0.7026           -1.3948
##       Merc 450SLC Chrysler Imperial        Merc 450SE         Merc 280C
##           -0.1719            1.0178            1.6296           -0.7580
##        Datsun 710 Hornet Sportabout      Ferrari Dino        Volvo 142E
##           -1.0672           -0.2582           -0.7623           -0.3332
##          Merc 230     Porsche 914-2      Lotus Europa    Hornet 4 Drive
##            0.4817           -1.1843            2.2229            0.5563
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.47443   28.25083   0.831   0.4439
## cyl         -0.89287    1.54183  -0.579   0.5876
## disp         0.02245    0.02059   1.090   0.3253
## hp          -0.01853    0.03293  -0.563   0.5980
## drat        -0.97533    2.57539  -0.379   0.7205
## wt          -2.22607    1.91853  -1.160   0.2983
## qsec        -0.07328    0.90761  -0.081   0.9388
## vs          -0.05865    2.12352  -0.028   0.9790
## am          -0.83463    2.94533  -0.283   0.7882
## gear         3.82286    1.71670   2.227   0.0765 .
## carb        -0.85255    0.97728  -0.872   0.4229
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.002 on 5 degrees of freedom
## Multiple R-squared:  0.9267, Adjusted R-squared:  0.7802
## F-statistic: 6.324 on 10 and 5 DF,  p-value: 0.02754
```

The engine shape represented by the dichotomous categorical variable vs 0 = V shaped and 1 = straight has the highest p value. I remove eliminate that variable and rerun the regression.

```
car_lm2 <- update(car_lm_full, .~. - vs, data =
train)
summary(car_lm2)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + drat + wt + qsec + am +
##     gear + carb, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8015 -0.8621 -0.2114  0.7925  2.2071
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.24627   24.66418   0.943   0.3823
## cyl         -0.87355    1.25443  -0.696   0.5123
## disp         0.02249    0.01876   1.198   0.2759
## hp          -0.01849    0.03004  -0.615   0.5609
## drat        -0.94510    2.12833  -0.444   0.6726
## wt          -2.22873    1.74930  -1.274   0.2498
## qsec        -0.07552    0.82527  -0.092   0.9301
```

```
## am          -0.82140    2.65310  -0.310   0.7673
## gear         3.82651    1.56261   2.449   0.0499 *
## carb        -0.85397    0.89096  -0.958   0.3748
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.828 on 6 degrees of freedom
## Multiple R-squared:  0.9267, Adjusted R-squared:  0.8168
## F-statistic: 8.431 on 9 and 6 DF,  p-value: 0.008665
```

This process is repeated until there are no variables with a p value over .05

```
car_lm3 <- update(car_lm_full, .~. - vs - qsec, data =
train)
summary(car_lm3)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + drat + wt + am + gear +
##     carb, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7835 -0.8405 -0.2484  0.8478  2.2126
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.29157   11.42472   1.864   0.1047
## cyl         -0.80566    0.93719  -0.860   0.4184
## disp         0.02289    0.01689   1.355   0.2175
## hp          -0.01749    0.02593  -0.675   0.5216
## drat        -0.88757    1.88385  -0.471   0.6519
## wt          -2.33166    1.24117  -1.879   0.1024
## am          -0.64060    1.64061  -0.390   0.7078
## gear         3.82570    1.44768   2.643   0.0333 *
## carb        -0.84794    0.82319  -1.030   0.3372
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.693 on 7 degrees of freedom
## Multiple R-squared:  0.9266, Adjusted R-squared:  0.8428
## F-statistic: 11.05 on 8 and 7 DF,  p-value: 0.002407
```

```
car_lm4 <- update(car_lm_full, .~. - vs - qsec - am, data =
train)
summary(car_lm4)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + drat + wt + gear + carb,
##     data = train)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8373 -0.8471 -0.2060  0.8604  2.2971
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.08818   10.62896   2.078   0.0713 .
## cyl         -0.82527    0.88488  -0.933   0.3783
## disp         0.02292    0.01597   1.435   0.1892
## hp          -0.01826    0.02445  -0.747   0.4766
## drat        -0.98982    1.76397  -0.561   0.5901
## wt          -2.20730    1.13428  -1.946   0.0875 .
## gear         3.58978    1.24393   2.886   0.0203 *
## carb        -0.83918    0.77807  -1.079   0.3122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.601 on 8 degrees of freedom
## Multiple R-squared:  0.925,  Adjusted R-squared:  0.8594
## F-statistic:  14.1 on 7 and 8 DF,  p-value: 0.0006354
```

```
car_lm5 <- update(car_lm_full, .~. - vs - qsec - am - drat, data =
train)
summary(car_lm5)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + wt + gear + carb, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4273 -0.9730 -0.3338  0.8226  2.5400
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.93194    7.32728   2.447   0.0369 *
## cyl         -0.56795    0.72742  -0.781   0.4550
## disp         0.02123    0.01508   1.408   0.1927
## hp          -0.01627    0.02325  -0.700   0.5017
## wt          -2.18143    1.08935  -2.003   0.0762 .
## gear         3.45881    1.17441   2.945   0.0163 *
## carb        -0.98710    0.70364  -1.403   0.1942
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.539 on 9 degrees of freedom
## Multiple R-squared:  0.9221, Adjusted R-squared:  0.8701
## F-statistic: 17.75 on 6 and 9 DF,  p-value: 0.0001613
```

```
car_lm6 <- update(car_lm_full, .~. - vs - qsec - am - drat - hp, data =
train)
summary(car_lm6)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + wt + gear + carb, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7229 -1.0953 -0.2846  1.0026  2.2777
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.42366    6.82208   2.407  0.03685 *
## cyl         -0.53468    0.70710  -0.756  0.46699
## disp         0.01418    0.01093   1.298  0.22349
## wt          -1.91426    0.99388  -1.926  0.08298 .
## gear         3.59498    1.12824   3.186  0.00971 **
## carb        -1.29504    0.53487  -2.421  0.03598 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.499 on 10 degrees of freedom
## Multiple R-squared:  0.9178, Adjusted R-squared:  0.8767
## F-statistic: 22.34 on 5 and 10 DF,  p-value: 3.951e-05
```

```
car_lm7 <- update(car_lm_full, .~. - vs - qsec - am - drat - hp - cyl, data =
train)
summary(car_lm7)
```

```
##
## Call:
## lm(formula = mpg ~ disp + wt + gear + carb, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6898 -1.0916 -0.4164  1.1995  2.1870
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.987505   4.988272   2.604  0.02454 *
## disp         0.008281   0.007499   1.104  0.29303
## wt          -1.702545   0.934884  -1.821  0.09587 .
## gear         4.033918   0.948435   4.253  0.00136 **
## carb        -1.577821   0.374887  -4.209  0.00146 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.47 on 11 degrees of freedom
## Multiple R-squared:  0.9131, Adjusted R-squared:  0.8815
## F-statistic: 28.91 on 4 and 11 DF,  p-value: 8.779e-06
```

```
car_lm8 <- update(car_lm_full, .~. - vs - qsec - am - drat - hp - cyl - disp, data =
train)
summary(car_lm8)
```

```
##
## Call:
## lm(formula = mpg ~ wt + gear + carb, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8137 -1.0901 -0.1061  0.8014  2.7118
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.2922     4.8905   2.922  0.01278 *
## wt           -1.1573     0.8011  -1.445  0.17416
## gear          3.6219     0.8799   4.116  0.00143 **
## carb         -1.4813     0.3679  -4.027  0.00168 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.483 on 12 degrees of freedom
## Multiple R-squared:  0.9035, Adjusted R-squared:  0.8794
## F-statistic: 37.45 on 3 and 12 DF,  p-value: 2.268e-06
```

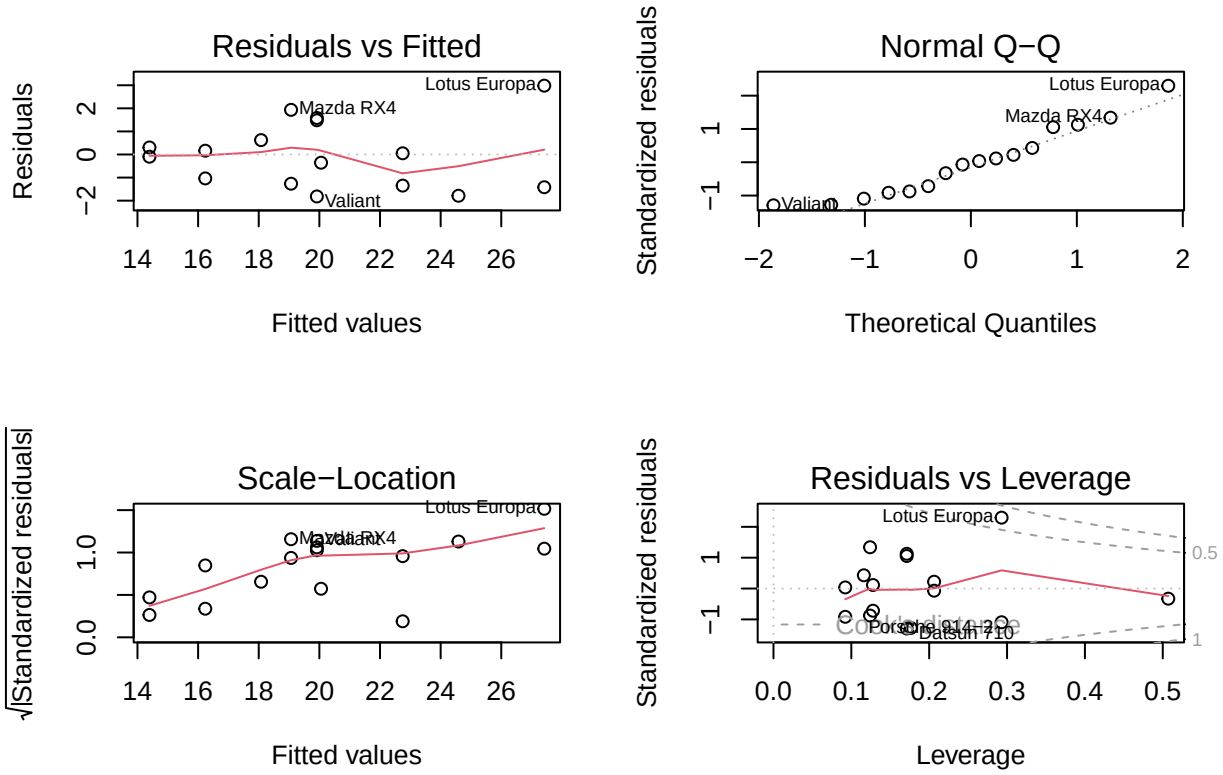
```
car_lm9 <- update(car_lm_full, .~. - vs - qsec - am - drat - hp - cyl - disp - wt, data =
train)
summary(car_lm9)
```

```
##
## Call:
## lm(formula = mpg ~ gear + carb, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.81747 -1.28651 -0.02147  0.83805  2.98169
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.7459     1.9143   4.046  0.00139 **
## gear          4.6708     0.5174   9.027 5.82e-07 ***
## carb         -1.8407     0.2821  -6.525 1.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.544 on 13 degrees of freedom
## Multiple R-squared:  0.8867, Adjusted R-squared:  0.8693
## F-statistic: 50.88 on 2 and 13 DF,  p-value: 7.112e-07
```

Residual Analysis

Our residuals are clustered around 0 and have a nearly normal distribution. This would indicate that our model is appropriate.

```
par(mfrow=c(2,2))
plot(car_lm9)
```



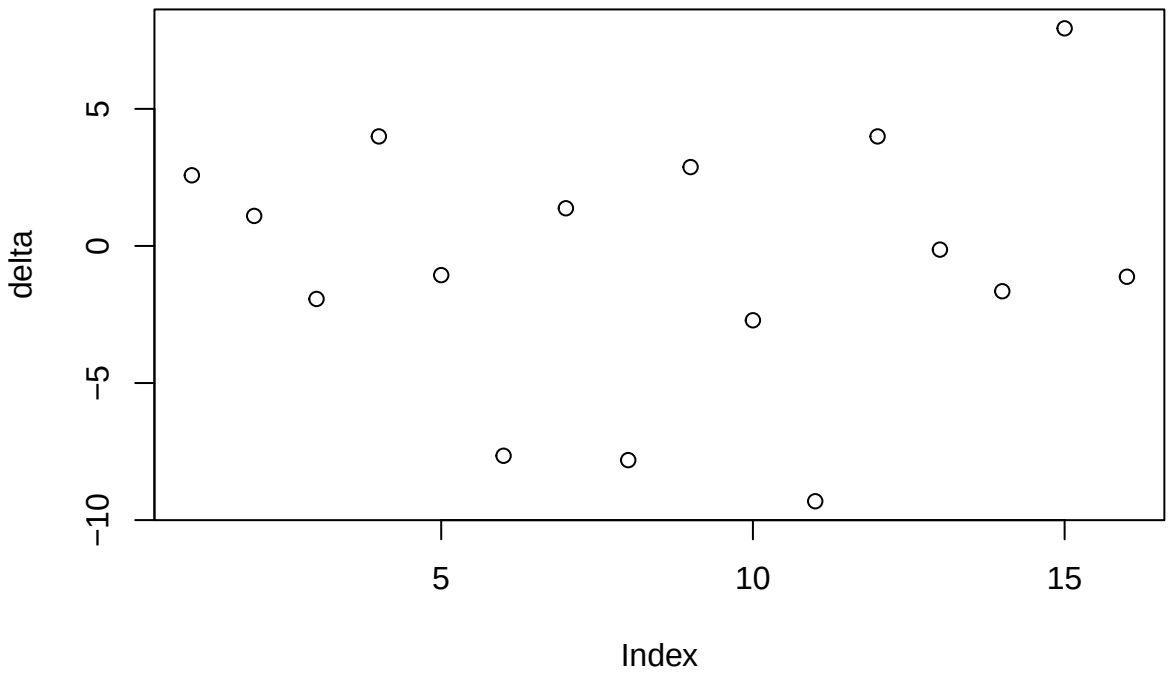
Model Testing

Since our t test is relatively tight interval that includes 0, the model is reasonably good at predicting the mpg of the test group based on the training data. Similarly, the deltas are clustered around 0 ignoring a few outliers.

```
predicted <- predict(car_lm9, newdata=test)
delta <- predicted - test$mpg
t.test(delta, conf.level = .95)
```

```
##
##  One Sample t-test
##
## data:  delta
## t = -0.50889, df = 15, p-value = 0.6182
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -3.094947  1.901929
## sample estimates:
##  mean of x
## -0.5965087
```

```
plot(delta)
```


delta
5
0
−5
−10
5
10
15
Index