STAT350 Tutorial 3

22/09/2020

The tutorial this week will be similar to the one from last. The difference now is that we include multiple regressors in our linear regression model.

The Data

For this weeks tutorial we'll be using the auto-mpg dataset. The goal is to predict the miles per gallon (mpg) of a vehicle given other measurements for the vehicle such as the number of cylinders (cyl), engine displacement (disp), horsepower (hp), weight of the vehicle (wt), acceleration (acc), and model year (year).

This dataset does not include a header so we'll have to add variable names ourselves.

```
auto_mpg <- read.table(paste('https://archive.ics.uci.edu/ml/',
                             'machine-learning-databases/auto-mpg/auto-mpg.data',
                             sep = ''))

colnames(auto_mpg) = c("mpg", "cyl", "disp", "hp", "wt", "acc",
                       "year", "origin", "name")

head(auto_mpg)
```

```
##   mpg cyl disp    hp   wt  acc year origin                      name
## 1  18   8  307 130.0 3504 12.0   70      1 chevrolet chevelle malibu
## 2  15   8  350 165.0 3693 11.5   70      1         buick skylark 320
## 3  18   8  318 150.0 3436 11.0   70      1        plymouth satellite
## 4  16   8  304 150.0 3433 12.0   70      1             amc rebel sst
## 5  17   8  302 140.0 3449 10.5   70      1               ford torino
## 6  15   8  429 198.0 4341 10.0   70      1          ford galaxie 500
```

Below I use the `str()` function to check structure of the data. Making sure variables are formatted properly.

```
str(auto_mpg)
```

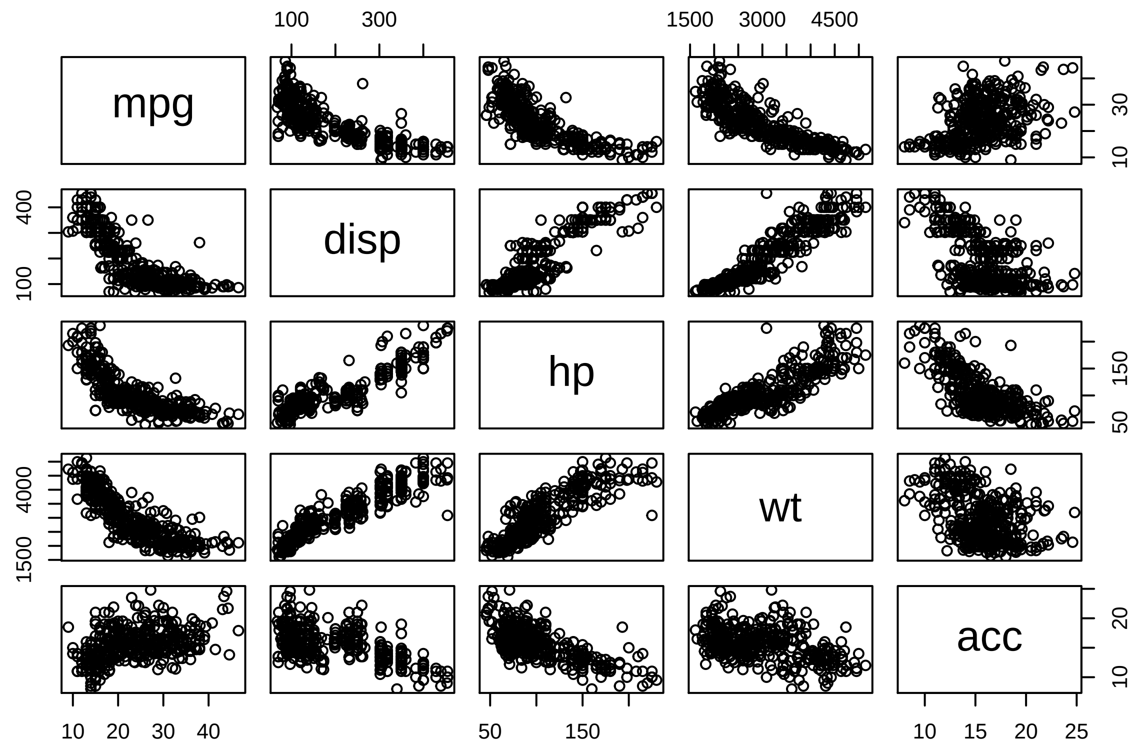
```
## 'data.frame':    398 obs. of  9 variables:
##  $ mpg   : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cyl   : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ disp  : num  307 350 318 304 302 429 454 440 455 390 ...
##  $ hp    : chr  "130.0" "165.0" "150.0" "150.0" ...
##  $ wt    : num  3504 3693 3436 3433 3449 ...
##  $ acc   : num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year  : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ name  : chr  "chevrolet chevelle malibu" "buick skylark 320" "plymouth satellite" "amc rebel sst"
```

So, we'll remove the second column and last three columns as we're currently only concerned with continuous predictors for a continuous response. Then we'll remove observations where hp="?" and change hp from a character to a numeric variable.

```
auto_mpg <- auto_mpg[,-c(2, 7, 8, 9)]
auto_mpg <- subset(auto_mpg, auto_mpg$hp != "?")
auto_mpg$hp <- as.numeric(auto_mpg$hp)
```

Next, we'll visualize the data with a scatter plot matrix using the `pairs()` function.

```
pairs(auto_mpg)
```



With this plot we can tell quite a bit about the relationship between variables in the dataset. Mainly, we see that a linear regression model seems appropriate. Another thing to take notice of is the multicollinearity present among the explanatory variables something we'll address in the coming weeks.

Multiple Linear Regression

To keep things simple, we'll have a model with only two predictors; wt and hp. So, our regression model equation will be:

$$mpg = \beta_0 + \beta_2(wt) + \beta_2(hp) + \epsilon$$

Fitting the Model

I now fit the model using the `lm()` function. This is the same as we've done before, where the response is on the left of the `~` and the predictors are on the right.

```r
mdl <- lm(mpg ~ wt + hp, data = auto_mpg)
(mdl_sum <- summary(mdl))
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp, data = auto_mpg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0762  -2.7340  -0.3312   2.1752  16.2601
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 45.6402108  0.7931958  57.540  < 2e-16 ***
## wt          -0.0057942  0.0005023 -11.535  < 2e-16 ***
## hp          -0.0473029  0.0110851  -4.267 2.49e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.24 on 389 degrees of freedom
## Multiple R-squared:  0.7064, Adjusted R-squared:  0.7049
## F-statistic: 467.9 on 2 and 389 DF,  p-value: < 2.2e-16
```

We could also compute the value of the coefficients directly from the data using the least squares equation:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}.$$

Here I use the functions `solve()` and `t()` to find the inverse and transpose of the design matrix $\boldsymbol{X}$, respectively. Making sure that the first column of $\boldsymbol{X}$ is a column of 1s.

```r
X <- cbind(rep(1, nrow(auto_mpg)), auto_mpg$wt, auto_mpg$hp)

solve(t(X) %*% X) %*% t(X) %*% auto_mpg$mpg
```

```
##              [,1]
## [1,] 45.640210840
## [2,] -0.005794157
## [3,] -0.047302863
```

Hypothesis Tests

Hypothesis Test for a Single β_j

Conducting hypothesis tests for a single β_j is in practice the same as simple linear regression, the difference is how we interpret the test. Since there are other regressors in the model we are doing a marginal test. That is, if we test the following hypothesis:

$$H_0: \ \beta_j = 0 \ \text{ vs. } \ \mathrm{H}_A : \beta_j \neq 0,$$

with $j = 1$, we are testing if there is a relationship between wt and mpg given that hp is included in the model. So, another way of stating the hypothesis is through the model equation:

$$H_o: \ mpg = \beta_0 + \beta_2(hp) + epsilon$$

vs.

$$H_A: \ mpg = \beta_0 + \beta_1(wt)\beta_2(hp) + epsilon$$

The test statistic for this hypothesis is given by:

$$t = \frac{\hat{\beta}_j}{se(\hat{\beta}_j)}.$$

Once this is computed, we find the corresponding p-value and compare to some level α.

Like before all the information we need to conduct the above hypothesis test is contained in the coefficients object of the model summary.

```
mdl_sum$coefficients
```

```
##                 Estimate  Std. Error    t value      Pr(>|t|)
## (Intercept) 45.640210840 0.793195833  57.539650 2.317113e-192
## wt          -0.005794157 0.000502327 -11.534633  1.124362e-26
## hp          -0.047302863 0.011085086  -4.267253  2.488482e-05
```

So, for our hypothesis test on the coefficient for wt we have a p-value that is practically zero. We therefore reject the null hypothesis that wt should not be included in our linear model for mpg, given that hp is in the model.

Hypothesis Test for Significance of Regression

Another hypothesis test we can do an ANOVA F-test for the significance of regression. Testing to see if there is a significant linear relationship between at least one of the predictors and the response, i.e. if all the coefficients are simultaneously zero;

$$H_o: \ \beta_1 = \beta_2 = \ldots = \beta_k = 0$$

vs.

$$H_A: \ \beta_j \neq 0 \text{for at least one } j.$$

The test statistic for this hypothesis is given by:

$$F_0 = \frac{MS_R}{MS_{Res}}$$

Like with all hypothesis tests once the test statistic is computed, we find the p-value and compare it to α. Again, this information is all contained in the model summary object. However, although the p-value is shown in the output it is not readily accessible.

```
(fstat <- mdl_sum$fstatistic)
```

```
##    value    numdf    dendf
## 467.9102   2.0000 389.0000
```

If we want to access the p-value we'll have to compute it ourselves.

```
pf(fstat[1], fstat[2], fstat[3], lower.tail = FALSE)
```

```
##         value
## 3.059606e-104
```

So, our p-value for the test of significance of regression is extremely small. We therefore reject the null hypothesis and conclude that at least one of the coefficients is not equal to zero.

Another way of interpreting the significance of regression hypothesis is that we are comparing the null model (the model with no regressors included) to the full model (the model with all regressors included). Performing an analysis of variance to test to see if there is a significant difference between the two models. To do this We can fit the null model using `lm()` and compare it to the full model using the `anova()` function.

```r
null_mdl <- lm(mpg ~ 1, data = auto_mpg)
anova(null_mdl, mdl)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ 1
## Model 2: mpg ~ wt + hp
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    391 23819.0
## 2    389  6993.8  2     16825 467.91 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output here more closely resembles the ANONA table you're familiar with. And, we see that this way of performing the significance of regression test comes to the same conclusion as before.

The useful thing about performing the test this way is that we are not constricted to just comparing the null and the full models. We can compare other subsets of the full model (aka nested models). For example, we can test to see if there is a significant difference between the model with just wt as a predictor and the full model with both wt and hp included. The hypothesis becomes:

$$H_0: \ \beta_2 = 0$$

$$\text{vs.}$$

$$H_A: \ \beta_2 \neq 0$$

The F-statistic for this test is similar in form to the one above, only the calculation of MS_R will change to be sum of squared the differences between the predictions for the two model. I now perform the test, first fitting the SLR with just wt as a predictor, then performing the ANOVA with the full model.

```r
wt_mdl <- lm(mpg ~ wt, data = auto_mpg)
anova(wt_mdl, mdl)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt
## Model 2: mpg ~ wt + hp
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
```

```
## 1    390 7321.2
## 2    389 6993.8  1    327.39 18.209 2.488e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here we again reject the null hypothesis, concluding that the coefficient for hp is not zero.

Confidence Intervals

Confiden Intervals for β_j

In the last tutorial I showed how to create confidence interval directly, using g quantities found in the model summary object. This week we'll be using the `confint()` function. The equation follows the familiar form confidence intervals - the estimate plus/minus the margin of error. Where the margin of error is the critical value of the t-distribution on $n - p$ degrees of freedom times the standard error of the coefficient estimate.

Below I calculate a 90% confidence interval for the the coefficients in our full model.

```
confint(mdl, level = 0.9)
```

```
##                      5 %         95 %
## (Intercept) 44.332405277 46.948016403
## wt          -0.006622384 -0.004965931
## hp          -0.065579734 -0.029025992
```

Prediction

Preforming prediction and computing intervals for predictions are are again essentially the same as for SLR. Remember to avoid extrapolation - this a little more tricky when dealing with more than one predictor.

Below I compute predictions along with a 90% prediction interval for a single new data point.

```
x_new <- data.frame(wt = 2500, hp = 150)

predict(mdl, newdata = x_new, interval = 'prediction', level = 0.9)
```

```
##        fit     lwr      upr
## 1 24.05939 16.9588 31.15998
```

Coefficient of Determination

R^2 for multiple linear regression has the same interpretation as in SLR. However, now the adjusted R^2 is relevant, whose value is given by:

$$R^2_{adj} = 1 - (1 - R^2)\frac{n-1}{n-p-1}.$$

Here we see that there is a penalization for including more terms in the model that do not add much to the model in terms of explaining the variance found in the response.

```
c(mdl_sum$r.squared, mdl_sum$adj.r.squared)
```

```
## [1] 0.7063753 0.7048656
```

Here we see that the value does not change too much since, as we found in previous sections, both terms is our model are significant predictors of the response.