Week 7 Plugging Values into a Model

Transformed Predictor Variables

For this sample report, I'll use a portion of R's built-in data set `mtcars`. I created a data file with five of the variables from that set for the purposes of this sample report.

We load the data set.

```
library(tidyverse)
```

```
myData <- read_csv("carmpg.csv")
```

```
Parsed with column specification:
cols(
  mpg = col_double(),
  disp = col_double(),
  hp = col_double(),
  wt = col_double(),
  gear = col_double()
)
```

```
attach(myData)
```

```
The following object is masked from package:ggplot2:

    mpg
```

```
head(myData)
```

```
# A tibble: 6 x 5
    mpg  disp    hp    wt  gear
  <dbl> <dbl> <dbl> <dbl> <dbl>
1  21     160   110  2.62     4
2  21     160   110  2.88     4
3  22.8   108    93  2.32     4
4  21.4   258   110  3.22     3
5  18.7   360   175  3.44     3
6  18.1   225   105  3.46     3
```

The variable `gear` only has three values which seems to suggest that we should treat it as a categorical variable, not numerical. We can tell R to do this by specifying that `gear` is a factor. This will tell R to create dummy variables when making the model.

```
nominal_gear <- factor(gear)
```

When checking correlations, I found displacement and horsepower both show some non-linearity. We will transform both and then attempt to build the model.

```
ln_disp <- log(disp)
ln_hp <- log(hp)
```

You can verify that the new scatterplots show a much more linear relationship and the correlation coefficients for `mpg` with both `ln_disp` and `ln_hp` have increased.

Building the Model

Let's construct a model and attempt to make a prediction from it. For the purposes of this exercise, I'm not going to worry about which variables are significant.

```
mpgModel <- lm( mpg ~ ln_disp + ln_hp + wt + nominal_gear)
mpgModel
```

```

Call:
lm(formula = mpg ~ ln_disp + ln_hp + wt + nominal_gear)

Coefficients:
  (Intercept)        ln_disp          ln_hp             wt  nominal_gear4
       70.080         -3.137         -5.703         -1.720         -0.715
nominal_gear5
        1.491
```

Now let's predict the mpg for a car with displacement 200, 150 horsepower, weight 3.5, and 4 gears. Be careful when plugging in since some of these are transformed. In particular, note the quotes around the nominal variable.

```
values <- data.frame( ln_disp = log(200), ln_hp = log(150), wt=3.5, nominal_gear = "4")
predict(mpgModel, values, interval="predict")
```

```
       fit      lwr      upr
1 18.14809 12.99643 23.29974
```

Other transformations

What if we've created a model where we transformed the response variable? With the Mammal Gestation data in Example 9-2 of Chapter 9 in the Penn State e-book on regression, the authors conclude the only problem with the data is unequal variances in the errors. They decide to transform the response variable to address this problem.

Let's load the data, apply a `log` transformation to the response variable, then generate the new model.

```
gest <- read_csv("mammgest.csv")
attach(gest)
```

```
ln_Gestation <- log(Gestation)
gest_NEWmodel <- lm( ln_Gestation ~ Birthwgt)
gest_NEWmodel
```

```

Call:
lm(formula = ln_Gestation ~ Birthwgt)

Coefficients:
(Intercept)     Birthwgt
    5.27882      0.01041
```

Let's make a prediction for a birthweight of 45.

```
predict( gest_NEWmodel, data.frame( Birthwgt = 45) )
```

```
       1
5.747278
```

Now is this the prediction for gestation length? No, its the prediction for `ln_Gestation`

```
exp(5.747278)
```

```
[1] 313.3366
```

How about a prediction interval?

```
predict( gest_NEWmodel, data.frame( Birthwgt = 45), interval = "predict" )
```

```
       fit      lwr      upr
1 5.747278 5.234582 6.259974
```

```
exp(5.234582)
```

```
[1] 187.6507
```

```
exp(6.259974)
```

```
[1] 523.2053
```

So our prediction interval is between 187.6507 and 523.2053.