March 29/30 Lab

Neil Montgomery

March 29, 2016

There is not much truly new to do for Lab work this week.

The second assignment will come out soon. It will involve fitting a variety of multiple regression models.

For this week let's make sure everyone is up to speed with fitting such models, using the data from exercise 12.45 and 12.65 from the book.

Dummy variables - 12.45

R can automatically generate the required dummy variables. Let's see how it works by trying it both ways.

```r
library(rio)
library(dplyr)
mpg <- import("Ex12.45.txt")
str(mpg)
```

```
## 'data.frame':    25 obs. of  4 variables:
##  $ MPG     : num  34.5 33.3 30.4 32.8 35 29 32.5 29.6 16.8 19.2 ...
##  $ Type    : chr  "sedan" "sedan" "sedan" "sedan" ...
##  $ Odometer: int  75000 60000 88000 15000 25000 35000 102000 98000 56000 72000 ...
##  $ Octane  : num  87.5 87.5 78 78 90 78 90 87.5 87.5 90 ...
```

The Type variable is a "character" variable according to R. We would call that a categortical or factor variable. Let's fit the model MPG ~ Type

```r
summary(lm(MPG ~ Type, data=mpg))
```

```
##
## Call:
## lm(formula = MPG ~ Type, data = mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2889 -2.5375  0.6111  2.3625  5.0111
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   32.137      1.032  31.148  < 2e-16 ***
## Typesuv      -12.325      1.459  -8.447 2.37e-08 ***
## Typevan      -12.049      1.418  -8.497 2.14e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.918 on 22 degrees of freedom
## Multiple R-squared:  0.8116, Adjusted R-squared:  0.7945
## F-statistic: 47.39 on 2 and 22 DF,  p-value: 1.06e-08
```

What has happened is that R has automagically converted `Type` to a `factor` variable, which we could do explicitly ourselves like this:

```
factor(mpg$Type)
```

```
##  [1] sedan sedan sedan sedan sedan sedan sedan sedan van   van   van  
## [12] van   van   van   van   van   van   suv   suv   suv   suv   suv  
## [23] suv   suv   suv  
## Levels: sedan suv van
```

The "levels" of the factor are: sedan, suv, van. R just takes all the unique values and makes the levels whatever they are in alphabetical order. You could define your own order of levels if you like, which we've done before mainly to make the order of boxplots be the way we wanted.

Look back at the regression output. What has happened is that R creates two dummy variables called `Typesuv` and `Typevan`. What happened to "sedan"? That's the (0,0) setting of the dummy variables, as requested in part (a) of the textbook question. "suv" is the (1,0) setting and "van" is the (0,1) setting.

We could make are own dummy variables and see if we get the same results. Let's call them t1 and t2.

```
mpg %>%
  mutate(t1 = Type=="suv", t2 = Type=="van") -> mpg2
```

If you look at `mpg2` you'll see two new columns with `TRUE` and `FALSE` in them, which are treated as 0 and 1 by R when suitable. So these are dummy variables. Let's fit the model with these two variables:

```
summary(lm(MPG ~ t1 + t2, mpg2))
```

```
## 
## Call:
## lm(formula = MPG ~ t1 + t2, data = mpg2)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.2889 -2.5375  0.6111  2.3625  5.0111 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   32.137      1.032  31.148  < 2e-16 ***
## t1TRUE       -12.325      1.459  -8.447 2.37e-08 ***
## t2TRUE       -12.049      1.418  -8.497 2.14e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.918 on 22 degrees of freedom
## Multiple R-squared:  0.8116, Adjusted R-squared:  0.7945 
## F-statistic: 47.39 on 2 and 22 DF,  p-value: 1.06e-08
```

Everything is identical. Try making two new dummy variables, this time encoding "van" as the (0,0) case. Run the regression model with these two new dummy variables. What changed? What stayed the same?

Finally, run the full model with `Type`, `Odometer` and `Octane` as inputs. Give a practical interpretation to the `Typevan` and `Typesuv` lines of the output, which answers 12.45(b).

12.65

Use the data to fit a variety of models including higher order terms (polynomial and interaction terms). Note that including an interaction term in the model automatically includes the individual terms as well. Note that there are two possible output variables. Just use `Y1` for now.