More Categorical Predictors

2017-03-10

Preparation

In this set of notes, you will continue learning about the inclusion of categorical predictors in regression models. To do so, we will examine the question of whether there are differences in reading scores between different ethnic groups. The data we will use in this set of notes is a subset of data from the Early Childhood Longitudinal Study (ECLS), a longitudinal study of child development and school readiness. The dataset, *ecls.csv*, includes the following variables for 200 4th graders:

- `id`: Student ID variable
- `read`: IRT scaled reading score (scale is from 0–96)
- `sex`: Student's sex/gender
- `race`: Student's race/ethnicity
- `ses`: Student's social economic status (as a z-score)
- `momEd`: Student's mother's level of education on an ordinal Likert scale

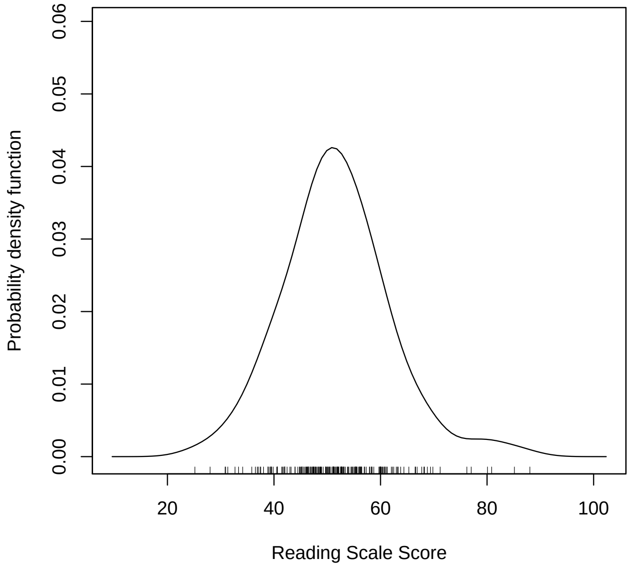
```
# Read in data
ecls = read.csv(file = "~/Google Drive/Documents/epsy-8251/data/ecls.csv")
head(ecls)
```

```
  id   read    sex     race   ses momEd
1  1 45.901   Male    White -0.29     3
2  2 52.913 Female    White -0.34     3
3  3 39.703   Male    White  0.13     4
4  4 49.583   Male    White  0.11     5
5  5 41.877 Female    White  0.96     6
6  6 37.746   Male Hispanic -1.27     1
```

```
# Load libraries
library(dplyr)
library(ggplot2)
library(sm)
```

Examine and Describe the Marginal Distribution of the Outcome

```r
sm.density(ecls$read, xlab = "Reading Scale Score")
```



```r
mean(ecls$read)
```

```
[1] 51.68699
```

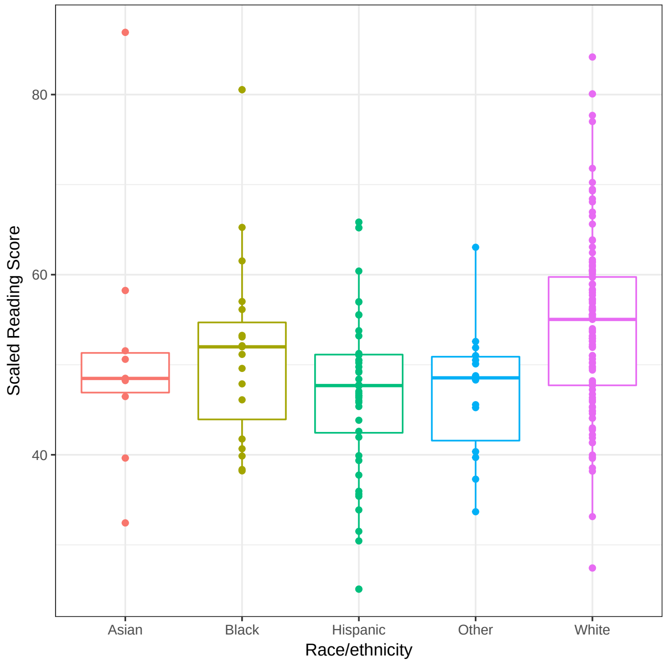
```r
sd(ecls$read)
```

```
[1] 10.07954
```

The reading scale scores seem approximately normally distributed ($M = 52$, $SD = 10$). This is not unexpected as the population was scaled to have a mean of 50 and a standard deviation of 10.

Examine and Describe the Distribution of the Outcome Conditioned on Race

```
# Plot the reading scores by ethnicity
ggplot(data = ecls, aes(x = race, y = read, color = race)) +
  geom_boxplot() +
  geom_point() +
    theme_bw() +
    xlab("Race/ethnicity") +
    ylab("Scaled Reading Score") +
  guides(color = FALSE)
```



```
ecls %>%
  group_by(race) %>%
  summarize( M = mean(read), SD = sd(read), N = n() )
```

Race	M	SD	N
Asian	51.10	14.36	10
Black	51.40	10.37	19
Hispanic	46.88	8.66	44
Other	47.00	7.49	14
White	54.24	9.64	113

The sample data suggests that there are potential racial differences in average reading score; the sample means are different.

Regression Analysis

Before fitting a regression model, we need to create a dummy variable for EACH of the categorical predictor's categories. For our analysis, we will need to create five dummy variables: `asian`, `black`, `hispanic`, `other`, and `white`. To do this we will use the `ifelse()` function. Remember, the `ifelse()` function evaluates a conditional statement (which produces elements that are either `TRUE` or `FALSE`) and outputs one thing IF the element is `TRUE` and outputs something ELSE if the element is `FALSE`. The function's useage looks like this:

$$\texttt{ifelse}(\text{conditional statement},~\text{output if TRUE},~\text{output if FALSE})$$

For example, to evaluate whether a student is Asian, we can use the conditional statement:

```
ecls$race == "Asian"
```

```
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[23]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[34] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[45] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[56] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[67] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
 [ reached getOption("max.print") -- omitted 125 entries ]
```

When we are creating the dummy variable `asian`, we will give this variable a value of `1` if the student is Asian (a `TRUE` element in our logical vector) and a `0` if the student is not Asian (a `FALSE` element in our logical vector). So the full `ifelse()` syntax is this:

```
# Create asian dummy variable
ecls$asian = ifelse(ecls$race == "Asian", 1, 0)

# Examine data
head(ecls)
```

```
  id   read    sex     race   ses momEd asian
1  1 45.901   Male    White -0.29     3     0
2  2 52.913 Female    White -0.34     3     0
3  3 39.703   Male    White  0.13     4     0
4  4 49.583   Male    White  0.11     5     0
5  5 41.877 Female    White  0.96     6     0
6  6 37.746   Male Hispanic -1.27     1     0
```

Since none of the first six students had a `race` value that was `Asian`, the dummy code for each of them is 0.

Now we will create the other four dummy variables.

```
ecls$black    = ifelse(ecls$race == "Black",    1, 0)
ecls$hispanic = ifelse(ecls$race == "Hispanic", 1, 0)
ecls$other    = ifelse(ecls$race == "Other",    1, 0)
ecls$white    = ifelse(ecls$race == "White",    1, 0)

# Examine data
head(ecls)
```

```
  id   read    sex     race   ses momEd asian black hispanic other white
1  1 45.901   Male    White -0.29     3     0     0        0     0     1
2  2 52.913 Female    White -0.34     3     0     0        0     0     1
3  3 39.703   Male    White  0.13     4     0     0        0     0     1
4  4 49.583   Male    White  0.11     5     0     0        0     0     1
5  5 41.877 Female    White  0.96     6     0     0        0     0     1
6  6 37.746   Male Hispanic -1.27     1     0     0        1     0     0
```

If you do not know the actual names of the categories (or you want to check capitalization, etc.) use the `levels()` function to obtain the category names. (Note this only works if the variable is a factor. It will not work if the variable is a character vector instead of a factor. You can check this with the `str()` function.)

```
# Check whether race is a factor
str(ecls)
```

```
'data.frame':   200 obs. of  11 variables:
 $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ read    : num  45.9 52.9 39.7 49.6 41.9 ...
 $ sex     : Factor w/ 2 levels "Female","Male": 2 1 2 2 1 2 2 2 2 1 ...
 $ race    : Factor w/ 5 levels "Asian","Black",..: 5 5 5 5 5 3 3 5 5 5 ...
 $ ses     : num  -0.29 -0.34 0.13 0.11 0.96 -1.27 0.84 0.4 0.55 1.08 ...
 $ momEd   : int  3 3 4 5 6 1 3 5 3 5 ...
 $ asian   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ black   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ hispanic: num  0 0 0 0 0 1 1 0 0 0 ...
 $ other   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ white   : num  1 1 1 1 1 0 0 1 1 1 ...
```

```
# Get the factor levels
levels(ecls$race)
```

```
[1] "Asian"    "Black"    "Hispanic" "Other"    "White"
```

Fit the Regression Model Using the Dummy Variables

Once the dummy variables have been created, fit the regression using all but one of the dummy variables you created. The dummy variable you leave out will correspond to the reference category. For example, in the model fitted below, we use the predictors `black`, `hispanic`, `other`, and `white`; we did not include the `asian` predictor. As such, Asian students are our reference group.

```r
# Asian is reference group
lm.asian = lm(read ~ 1 + black + hispanic + other + white, data = ecls)
summary(lm.asian)
```

```

Call:
lm(formula = read ~ 1 + black + hispanic + other + white, data = ecls)

Residuals:
    Min      1Q  Median      3Q     Max 
-26.809  -6.467   0.518   4.707  35.803 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  51.1040     3.0513  16.749   <2e-16 ***
black         0.2935     3.7696   0.078    0.938    
hispanic     -4.2212     3.3802  -1.249    0.213    
other        -4.1002     3.9950  -1.026    0.306    
white         3.1342     3.1834   0.985    0.326    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.649 on 195 degrees of freedom
Multiple R-squared:  0.102, Adjusted R-squared:  0.08362 
F-statistic:  5.54 on 4 and 195 DF,  p-value: 0.0003033
```

At the model-level, differences in race/ethnicity seem to explain a statistically significant amount of the variation in reading scores ($p = .0003$). In fact, racial differences explain 10.2% of the variation in reading scores.

At the coefficient-level, the intercept is the average Y value for the reference group. Each partial slope is the difference in average Y values between the reference group and the group represented by the dummy variable. In our example, the average reading score for Asians is 51.1. Black students have an average reading score that is 0.3 higher than the average Asian reading score. Hispanic students have an average reading score that is 4.2 lower than the average Asian reading score. It is important to note that none of the partial slopes are statistically significant. This implies that there is likely no difference in reading score between Asians and any other racial group, on average.

Omnibus Test vs. Coefficient Tests with Multiple Dummy Variables

Recall that one manner in which we could write the model-level test is:

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_k = 0$$

This is like asking whether ALL the partial slopes are equal to zero. The answer to this question at the model-level is that it is likely that at least one partial slope is not zero ($p = .0003$). But what about at the coefficient-level? We failed to reject all of the tests for the partial slopes included in the model. Isn't this inconsistent with the model-level test?

It turns out that when we use multiple dummy variables to represnt a single categorical predictor, the test at the model-level is about differences between ALL groups, not just the differences included in the model. In the model we fitted, the differences were only between Asian students and the other racial groups. For example, the difference between Hispanic and Black students, or the difference between White and Black students is not included in the model we fitted. The test at the model-level is considering all 10 differences simultaneously.

Because of this, the results are not inconsistent, but rather are tests of different questions. Based on the model fitted so far, we have considered four of the 10 possible differences.

Comparison	Mean Difference	p
Asian vs. Black	0.29	0.938
Asian vs. Hispanic	-4.22	0.213
Asian vs. Other	-4.10	0.306
Asian vs. White	3.13	0.326
Black vs. Hispanic	?	?
Black vs. Other	?	?
Black vs. White	?	?
Hispanic vs. Other	?	?
Hispanic vs. White	?	?
Other vs. White	?	?

In order to examine the remaining differences, we need to fit additional regression models that allow for those comparisons.

```r
# Black is reference group
lm.black = lm(read ~ 1 + asian + hispanic + other + white, data = ecls)
summary(lm.black)
```

```

Call:
lm(formula = read ~ 1 + asian + hispanic + other + white, data = ecls)

Residuals:
    Min      1Q  Median      3Q     Max 
-26.809  -6.467   0.518   4.707  35.803 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  51.3975     2.2136  23.219   <2e-16 ***
asian        -0.2935     3.7696  -0.078   0.9380    
hispanic     -4.5147     2.6488  -1.704   0.0899 .  
other        -4.3937     3.3986  -1.293   0.1976    
```

```
white          2.8407     2.3925   1.187   0.2365
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.649 on 195 degrees of freedom
Multiple R-squared:  0.102, Adjusted R-squared:  0.08362
F-statistic:  5.54 on 4 and 195 DF,  p-value: 0.0003033
```

```r
# Hispanic is reference group
lm.hispanic = lm(read ~ 1 + asian + black + other + white, data = ecls)
summary(lm.hispanic)
```

```

Call:
lm(formula = read ~ 1 + asian + black + other + white, data = ecls)

Residuals:
    Min      1Q  Median      3Q     Max
-26.809  -6.467   0.518   4.707  35.803

Coefficients:
            Estimate Std. Error t value  Pr(>|t|)
(Intercept)   46.883      1.455  32.230   < 2e-16 ***
asian          4.221      3.380   1.249    0.2132
black          4.515      2.649   1.704    0.0899 .
other          0.121      2.961   0.041    0.9674
white          7.355      1.715   4.290 0.0000281 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.649 on 195 degrees of freedom
Multiple R-squared:  0.102, Adjusted R-squared:  0.08362
F-statistic:  5.54 on 4 and 195 DF,  p-value: 0.0003033
```

```r
# Other is reference group
lm.other = lm(read ~ 1 + asian + black + hispanic + white, data = ecls)
summary(lm.other)
```

```

Call:
lm(formula = read ~ 1 + asian + black + hispanic + white, data = ecls)

Residuals:
    Min      1Q  Median      3Q     Max
-26.809  -6.467   0.518   4.707  35.803

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   47.004      2.579  18.227   <2e-16 ***
asian          4.100      3.995   1.026   0.3060
black          4.394      3.399   1.293   0.1976
hispanic      -0.121      2.961  -0.041   0.9674
white          7.234      2.734   2.646   0.0088 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 9.649 on 195 degrees of freedom
Multiple R-squared:  0.102, Adjusted R-squared:  0.08362
F-statistic:  5.54 on 4 and 195 DF,  p-value: 0.0003033
```

```
# White is reference group
lm.white = lm(read ~ 1 + asian + black + hispanic + other, data = ecls)
summary(lm.white)
```

```

Call:
lm(formula = read ~ 1 + asian + black + hispanic + other, data = ecls)

Residuals:
    Min      1Q  Median      3Q     Max 
-26.809  -6.467   0.518   4.707  35.803 

Coefficients:
            Estimate Std. Error t value  Pr(>|t|)    
(Intercept)  54.2382     0.9077  59.754   < 2e-16 ***
asian        -3.1342     3.1834  -0.985    0.3261    
black        -2.8407     2.3925  -1.187    0.2365    
hispanic     -7.3554     1.7146  -4.290 0.0000281 ***
other        -7.2344     2.7339  -2.646    0.0088 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.649 on 195 degrees of freedom
Multiple R-squared:  0.102, Adjusted R-squared:  0.08362 
F-statistic:  5.54 on 4 and 195 DF,  p-value: 0.0003033
```

Now we can fill in the remaining cells of the table.

Comparison	Mean Difference	p
Asian vs. Black	0.29	0.938
Asian vs. Hispanic	-4.22	0.213
Asian vs. Other	-4.10	0.306
Asian vs. White	3.13	0.326
Black vs. Hispanic	-4.51	0.090
Black vs. Other	-4.39	0.198
Black vs. White	2.84	0.236
Hispanic vs. Other	0.12	0.967
Hispanic vs. White	7.36	0.0000281
Other vs. White	7.23	0.0088

It looks as though the only statistically significant reading differences are between Hispanic and White students. Although, there is also some evidence of reading differences between Black and Hispanic students, and between White and Other students.

Multiple Comparisons

When we evaluated the *p*-values for each of these differences, we used an alpha value of 0.05 as the criterion for significance. This is consistent with how we have evaluated other predictors in regression models. The main difference is that when we evaluated other predictors, the effect was a single row in the regression model, For the effect of race/ethnicity, we really have 10 rows. To be "fair" with the same type of evaluation we used for the other predictors in the model, we should really split the 0.05 across the 10 rows.

There are many ways to do this. The easeist manner is to divide the 0.05 evenly across the 10 rows.

$$\frac{0.05}{10} = 0.005$$

In this manner, rather than rejecting the null hypothesis when the racial comparison has a *p*-value below 0.05, we will only reject if the comparison has a *p*-value below 0.005. This still suggests that there are reading differences between Hispanic and White students. The two other comparisons, however, look much less promising.

In practice, people are psychologically accustomed to comparing the *p*-value to 0.05, so changing the comparison can be a problem. Another way to achieve the same adjustment, but still allow people to compare to 0.05, is to change the *p*-value rather than change the alpha value. To do this, we multiply each *p*-value from the racial comparisons by the 10 (rather than dividing the alpha value by 10).

Comparison	Mean Difference	p	Adjusted p
Asian vs. Black	0.29	0.938	9.380
Asian vs. Hispanic	-4.22	0.213	2.130
Asian vs. Other	-4.10	0.306	3.060
Asian vs. White	3.13	0.326	3.260
Black vs. Hispanic	-4.51	0.090	0.900
Black vs. Other	-4.39	0.198	1.980
Black vs. White	2.84	0.236	2.360
Hispanic vs. Other	0.12	0.967	9.670
Hispanic vs. White	7.36	0.0000281	0.000281
Other vs. White	7.23	0.0088	0.880

When reporting these, be careful. Remember that *p*-values are always between 0 and 1. Anything above 1 needs to be reported as 1! This method of evenly splitting the alpha value or adjusting the *p*-value evenly is called the Bonferroni adjustment.

We can also use the `p.adjust()` function to compute the Bonferroni adjusted *p*-values. To use this, create a vector of the unadjusted *p*-values and then include this vector in the `p.adjust()` function along with the argument `method = "bonferroni"`.

```
p.values = c(
  0.938,     # asian vs. black
  0.213,     # asian vs. hispanic
  0.306,     # asian vs. other
  0.326,     # asian vs. white
  0.090,     # black vs. hispanic
  0.198,     # black vs. other
  0.236,     # black vs. white
  0.967,    # hispanic vs. other
  0.0000281, # hispanic vs. white
  0.0088     # other vs. white
)


# Bonferroni adjustment to the p-values
p.adjust(p.values, method = "bonferroni")
```

```
 [1] 1.000000 1.000000 1.000000 1.000000 0.900000 1.000000 1.000000
 [8] 1.000000 0.000281 0.088000
```

Comparison	Mean Difference	p	Adjusted p
Asian vs. Black	0.29	0.938	1.000
Asian vs. Hispanic	-4.22	0.213	1.000
Asian vs. Other	-4.10	0.306	1.000
Asian vs. White	3.13	0.326	1.000
Black vs. Hispanic	-4.51	0.090	0.900
Black vs. Other	-4.39	0.237	1.000
Black vs. White	2.84	0.198	1.000
Hispanic vs. Other	0.12	0.967	1.000
Hispanic vs. White	7.36	0.0000281	0.000281
Other vs. White	7.23	0.088	0.0880

Other p-Value Adjustment Methods

There is nothing that requires you to evenly adjust the p-value across the 10 comparisons. For example, some adjustment methods use different multipliers depending on the size of the initial unadjusted p-value. One of those methods is the Benjamani–Hochberg adjustment. This adjustment procedure ranks the unadjusted p-values from smallest to largest and then adjusts by the following computation:

$$p_{\text{adjusted}} = \frac{k \times p_{\text{unadjusted}}}{\text{Rank}}$$

The numerator is equivalent to making the Bonferroni adjustment. The size of the Bonferroni adjustment is then scaled back depending on the initial rank of the unadjusted p-value. The smallest initial p-value gets the complete Bonferroni adjustment, while the largest is scaled back the most.

Comparison	Mean Difference	Unadjusted p	Rank	Adjusted p
Hispanic vs. White	7.36	0.0000281	1	0.000281
Other vs. White	7.23	0.0088	2	0.044
Black vs. Hispanic	-4.51	0.090	3	0.300
Black vs. Other	-4.39	0.198	4	0.393
Asian vs. Hispanic	-4.22	0.213	5	0.393
Black vs. White	2.84	0.236	6	0.393
Asian vs. Other	-4.10	0.306	7	0.408
Asian vs. White	3.13	0.326	8	0.408
Asian vs. Black	0.29	0.938	9	0.967
Hispanic vs. Other	0.12	0.967	10	0.967

Using the Benjamani–Hochburg adjustment, we find reading differences between Hispanic and White students, and between White and Other students. We can use `method="BH"` in the `p.adjust()` function to obtain the Benjamani–Hochberg adjusted *p*-values directly.

```r
# Benjamani-Hochburg adjustment to the p-values
p.adjust(p.values, method = "BH")
```

```
 [1] 0.9670000 0.3933333 0.4075000 0.4075000 0.3000000 0.3933333 0.3933333
 [8] 0.9670000 0.0002810 0.0440000
```

There are many, many different adjustment methods you can choose. The `p.adjust()` function, for example, includes six adjustment options (the Holm method, the Hochberg method, the Hommel method, the Bonferroni method, the Bnjamani–Hochberg method, and the Benjamani–Yekutieli method). In addition, the `multcomp` package includes several other adjustment methods.

You should decide which adjustment method you will use before you do the analysis. In the social sciences, the Bonferroni method has been historically the most popular method (probably because it was easy to implement before computing). That being said, I would encourage you to use the Benjamani–Hochberg adjustment method. It is from a family of adjustment methods that a growing pool of research evidence points toward as the "best" solution to the problem of multiple comparisons (Williams, Jones, & Tukey, 1999). Because of its usefulness, the Institute of Education Sciences has recommended this procedure for use in its What Works Clearinghouse Handbook of Standards.

ANCOVA Model: Controlling for Differences in Covariates

We might want to control for differences in other covariates. For example, are there still racial differences in reading scores after we control for differences in socioeconomic status and mother's level of education? To do this we fit the same models as before, except now we include the two covariates in the model along with the dummy variables. Again, we will need to fit several models, each with a different reference group, to obtain all 10 comparisons.

```r
# Asian is reference group
lm.asian = lm(read ~ 1 + black + hispanic + other + white + ses + momEd, data = ecls)
summary(lm.asian)
```

```

Call:
lm(formula = read ~ 1 + black + hispanic + other + white + ses + 
    momEd, data = ecls)

Residuals:
    Min      1Q  Median      3Q     Max 
```

```
-25.074  -5.462   0.391   4.538  33.010

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  47.5916     3.5500  13.406  < 2e-16 ***
black         2.0285     3.6000   0.563  0.57376    
hispanic     -1.4462     3.2335  -0.447  0.65519    
other        -0.9880     3.7893  -0.261  0.79457    
white         2.8110     3.0217   0.930  0.35339    
ses           4.0097     1.4721   2.724  0.00705 ** 
momEd         0.5163     0.6049   0.854  0.39439    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.033 on 193 degrees of freedom
Multiple R-squared:  0.2211,    Adjusted R-squared:  0.1968 
F-statistic: 9.129 on 6 and 193 DF,  p-value: 0.00000000855
```

```r
# Black is reference group
lm.black = lm(read ~ 1 + asian + hispanic + other + white + ses + momEd, data = ecls)
summary(lm.black)
```

```
Call:
lm(formula = read ~ 1 + asian + hispanic + other + white + ses + 
    momEd, data = ecls)

Residuals:
    Min      1Q  Median      3Q     Max 
-25.074  -5.462   0.391   4.538  33.010 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  49.6202     3.3732  14.710  < 2e-16 ***
asian        -2.0285     3.6000  -0.563  0.57376    
hispanic     -3.4747     2.4960  -1.392  0.16549    
other        -3.0166     3.2206  -0.937  0.35011    
white         0.7824     2.2760   0.344  0.73138    
ses           4.0097     1.4721   2.724  0.00705 ** 
momEd         0.5163     0.6049   0.854  0.39439    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.033 on 193 degrees of freedom
Multiple R-squared:  0.2211,    Adjusted R-squared:  0.1968 
F-statistic: 9.129 on 6 and 193 DF,  p-value: 0.00000000855
```

```r
# Hispanic is reference group
lm.hispanic = lm(read ~ 1 + asian + black + other + white + ses + momEd, data = ecls)
summary(lm.hispanic)
```

```
Call:
lm(formula = read ~ 1 + asian + black + other + white + ses + 
    momEd, data = ecls)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-25.074  -5.462   0.391   4.538  33.010 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  46.1454     2.7894  16.543  < 2e-16 ***
asian         1.4462     3.2335   0.447  0.65519    
black         3.4747     2.4960   1.392  0.16549    
other         0.4582     2.7812   0.165  0.86932    
white         4.2572     1.7050   2.497  0.01337 *  
ses           4.0097     1.4721   2.724  0.00705 ** 
momEd         0.5163     0.6049   0.854  0.39439    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.033 on 193 degrees of freedom
Multiple R-squared:  0.2211,    Adjusted R-squared:  0.1968 
F-statistic: 9.129 on 6 and 193 DF,  p-value: 0.00000000855
```

```r
# Other is reference group
lm.other = lm(read ~ 1 + asian + black + hispanic + white + ses + momEd, data = ecls)
summary(lm.other)
```

```
Call:
lm(formula = read ~ 1 + asian + black + hispanic + white + ses + 
    momEd, data = ecls)

Residuals:
    Min      1Q  Median      3Q     Max 
-25.074  -5.462   0.391   4.538  33.010 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  46.6036     3.2748  14.231  < 2e-16 ***
asian         0.9880     3.7893   0.261  0.79457    
black         3.0166     3.2206   0.937  0.35011    
hispanic     -0.4582     2.7812  -0.165  0.86932    
white         3.7990     2.6523   1.432  0.15366    
ses           4.0097     1.4721   2.724  0.00705 ** 
momEd         0.5163     0.6049   0.854  0.39439    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.033 on 193 degrees of freedom
Multiple R-squared:  0.2211,    Adjusted R-squared:  0.1968 
F-statistic: 9.129 on 6 and 193 DF,  p-value: 0.00000000855
```

Comparison	Adjusted Mean Difference	p
Asian vs. Black	2.02	0.573
Asian vs. Hispanic	-1.45	0.655
Asian vs. Other	-1.45	0.794
Asian vs. White	2.81	0.353
Black vs. Hispanic	-3.47	0.165
Black vs. Other	-3.02	0.350
Black vs. White	0.78	0.731
Hispanic vs. Other	0.46	0.869
Hispanic vs. White	4.26	0.013
Other vs. White	3.80	0.154

Similar to our previous analysis, we need to djust the *p*-values for the racial comparisons.

```r
p.values = c(
  0.573, # asian vs. black
  0.655, # asian vs. hispanic
  0.794, # asian vs. other
  0.353, # asian vs. white
  0.165, # black vs. hispanic
  0.350, # black vs. other
  0.731, # black vs. white
  0.869, # hispanic vs. other
  0.013, # hispanic vs. white
  0.154  # other vs. white
)


# Benjamani-Hochberg adjustment to the p-values
p.adjust(p.values, method = "BH")
```

```
 [1] 0.869 0.869 0.869 0.706 0.550 0.706 0.869 0.869 0.130 0.550
```

Comparison	Adjusted Mean Difference	p	Adjusted p
Asian vs. Black	2.02	0.573	0.869
Asian vs. Hispanic	-1.45	0.655	0.869
Asian vs. Other	-1.45	0.794	0.869
Asian vs. White	2.81	0.353	0.706
Black vs. Hispanic	-3.47	0.165	0.550
Black vs. White	0.78	0.731	0.869
Black vs. Other	-3.02	0.350	0.706
Hispanic vs. Other	0.46	0.869	0.869
Hispanic vs. White	4.26	0.013	0.130
Other vs. White	3.80	0.154	0.550

After controlling for differences in SES and mother's level of education, none of the Benjamani-Hochberg adjusted *p*-values suggest racial difference in average reading scores.

Technical Reasons to Adjust for Multiple Comparisons

In the earlier sections, we presented the reason for adjusting the p-values for the racial comparisons as one of "fairness" with the other predictors in the model. This is true, but there are also technical reasons to make these adjustments. The main technical reason is related to the *Type I error rate.* Remember that a Type I error occurs when you falsely reject a true null hypothesis. In other words, we would say there is a difference between racial groups when there really isn't a difference.

When we use an alpha value of 0.05, we are saying we are willing to make a Type I error in 5% of the samples that could be randomly selected (we have no idea whether our sample is one of the 5% where we will make an error, or one of the 95% where we won't). For effects that only have one row in the model, there is only one test in which we can make a Type I error ($H_0 : \beta_j = 0$), so we are okay evaluating each at the alpha of 0.05.

However when we are using multiple rows to represent an effect, there are multiple tests we are evaluating for the same effect. In fact, there are 10 such tests:

$$
\begin{aligned}
H_0 &: \beta_{\text{Asian vs. Black}} = 0 \\
H_0 &: \beta_{\text{Asian vs. Hispanic}} = 0 \\
H_0 &: \beta_{\text{Asian vs. Other}} = 0 \\
H_0 &: \beta_{\text{Asian vs. White}} = 0 \\
H_0 &: \beta_{\text{Black vs. Hispanic}} = 0 \\
H_0 &: \beta_{\text{Black vs. Other}} = 0 \\
H_0 &: \beta_{\text{Black vs. White}} = 0 \\
H_0 &: \beta_{\text{Hispanic vs. Other}} = 0 \\
H_0 &: \beta_{\text{Hispanic vs. White}} = 0 \\
H_0 &: \beta_{\text{Other vs. White}} = 0
\end{aligned}
$$

Because of this, there are many ways to make a Type I error. For example, we could make a Type I error in any one of the 10 tests, or in two of the 10 tests, or in three of the 10 tests, etc. Therefore, the probability of making at least one Type I error is no longer 0.05, it is

$$1 - (1 - \alpha)^k$$

where α is the alpha level for each test, and k is the number of tests (comparisons) for the effect. In our example this is

$$P(\text{type I error}) = 1 - (1 - 0.05)^{10} = 0.401$$

The probability that we will make at least one Type I error in the 10 tests is 40%!!! This probability is called the family-wise Type I error rate. In the social sciences, the family-wise error rate needs to be at 0.05. What should α be if we want the family-wise error rate to be 0.05? Essentially we would need to solve this equation:

$$0.05 = 1 - (1 - \alpha)^{10}$$

Carlo Emilio Bonferroni solved this algebra problem for any value of k and found that the value for alpha that $\frac{\text{family-wise error rate}}{k}$ gives an upper-bound for the solution. Olive Jean Dunn then used Bonferroni's solution in practice. This is why dividing by the number of comparisons is referred to as the Bonferroni or the Dunn–Bonferroni method.

False Discovery Rate

The Benjamini–Hochberg procedure is an ensemble method based on *false discovery rate* (FDR). FDR is a relatively new approach to the multiple comparisons problem. Instead of making adjustments to control the probability of making at least one Type I error, FDR controls the *expected proportion of discoveries* (rejected null hypotheses) when the null hypothesis is true; in other words, it controls the expected proportion of Type I error. You can find out more from Wikipedia.

The FDR concept was formally described in a 1995 paper by Yoav Benjamini and Yosi Hochberg, and resulted in their proposal of the Benjamani–Hochbergm method (Benjamini & Hochberg, 1995). They argued that using FDR produces a less conservative and arguably more appropriate approach for identifying statistically significant comparisons.

In practice, using FDR rather than family-wise adjustment of error makes these methods less prone to over-adjustment of the *p*-values. However, the increased statistical power that comes with using the FDR methods is not without cost. They also have increased probabilities of Type I errors relative to the family-wise adjustment methods.

References

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B*, *57*(1), 289—300.

Williams, V., Jones, L., & Tukey, J. (1999). Controlling error in multiple comparisons, with examples from state-to-state differences in educational achievement. *Journal of Educational and Behavioral Statistics*, *24*(1), 42–69.