Lecture 10 - Categorical variables and interaction terms in linear regression, Introduction to plyr

94-842

February 16, 2017

Contents

Let's begin by loading the packages we'll need to get started

```
library(MASS)
library(plyr)
library(ggplot2)
library(knitr)

options(scipen=4)
```

Interpreting linear regression: proceed with caution

```
# Import data set
crime <- read.table("http://www.andrew.cmu.edu/user/achoulde/94842/data/crime_simple.txt", sep = "\t", 
```

The variable names that this data set comes with are very confusing, and even misleading.

R: Crime rate: # of offenses reported to police per million population

Age: The number of males of age 14-24 per 1000 population

S: Indicator variable for Southern states (0 = No, 1 = Yes)

Ed: Mean # of years of schooling x 10 for persons of age 25 or older

Ex0: 1960 per capita expenditure on police by state and local government

Ex1: 1959 per capita expenditure on police by state and local government

LF: Labor force participation rate per 1000 civilian urban males age 14-24

M: The number of males per 1000 females

N: State population size in hundred thousands

NW: The number of non-whites per 1000 population

U1: Unemployment rate of urban males per 1000 of age 14-24

U2: Unemployment rate of urban males per 1000 of age 35-39

W: Median value of transferable goods and assets or family income in tens of $

X: The number of families per 1000 earning below 1/2 the median income

```r
# Assign more meaningful variable names
colnames(crime) <- c("crime.per.million", "young.males", "is.south", "average.ed",
                     "exp.per.cap.1960", "exp.per.cap.1959", "labour.part",
                     "male.per.fem", "population", "nonwhite",
                     "unemp.youth", "unemp.adult", "median.assets", "num.low.salary")

# Convert is.south to a factor
# Divide average.ed by 10 so that the variable is actually average education
# Convert median assets to 1000's of dollars instead of 10's
crime <- transform(crime, is.south = as.factor(is.south),
                          average.ed = average.ed / 10,
                          median.assets = median.assets / 100)

# Fit model
crime.lm <- lm(crime.per.million ~ ., data = crime)

# Remove 1959 expenditure and youth unemployment
crime.lm2 <- update(crime.lm, . ~ . - exp.per.cap.1959 - unemp.youth)
```

Here's a comparison of the regression models (with and without the collinearity problem).

```r
kable(summary(crime.lm)$coef,
      digits = c(3, 3, 3, 4), format = 'markdown')
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-691.838	155.888	-4.438	0.0001
young.males	1.040	0.423	2.460	0.0193
is.south1	-8.308	14.912	-0.557	0.5812
average.ed	18.016	6.497	2.773	0.0091
exp.per.cap.1960	1.608	1.059	1.519	0.1384
exp.per.cap.1959	-0.667	1.149	-0.581	0.5653
labour.part	-0.041	0.153	-0.267	0.7909
male.per.fem	0.165	0.210	0.785	0.4381
population	-0.041	0.130	-0.319	0.7520
nonwhite	0.007	0.064	0.112	0.9112
unemp.youth	-0.602	0.437	-1.376	0.1780
unemp.adult	1.792	0.856	2.093	0.0441
median.assets	13.736	10.583	1.298	0.2033
num.low.salary	0.793	0.235	3.373	0.0019

```r
crime.lm.summary2 <- summary(crime.lm2)
kable(crime.lm.summary2$coef,
      digits = c(3, 3, 3, 4), format = 'markdown')
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-633.439	145.470	-4.354	0.0001
young.males	1.127	0.419	2.691	0.0109
is.south1	-0.557	13.883	-0.040	0.9682

	Estimate	Std. Error	t value	Pr(>|t|)
average.ed	15.328	6.203	2.471	0.0185
exp.per.cap.1960	1.138	0.227	5.015	0.0000
labour.part	0.069	0.134	0.515	0.6101
male.per.fem	0.003	0.173	0.017	0.9862
population	-0.064	0.128	-0.503	0.6184
nonwhite	-0.014	0.062	-0.223	0.8250
unemp.adult	0.931	0.542	1.719	0.0944
median.assets	15.159	10.524	1.440	0.1587
num.low.salary	0.826	0.234	3.527	0.0012

Observe that the coefficient of 1960 expenditure went from being non-signficant to significant (p-value is now very small).

Practical considerations in linear regression

After dealing with the colinearity issue by removing the 1959 expenditure variable, we see that `exp.per.cap.1960` is now highly significant.

```
crime.lm.summary2$coef["exp.per.cap.1960",]
```

```
##      Estimate    Std. Error       t value      Pr(>|t|)
## 1.13829907170 0.22697675756 5.01504684417 0.00001532994
```

This is interesting. It's essentially saying that, all else being equal, every dollar per capita increase in police expenditure is on average associated with an increase in crime of 1.13 per million population.

```
crime.lm.summary2$coef["average.ed",]
```

```
##    Estimate  Std. Error     t value    Pr(>|t|)
## 15.32802778  6.20251646  2.47125951  0.01847635
```

Also, for every unit increase in average education, we find that the number of reported crimes on average increases by about 15.3 per million.

One of the main reasons for selecting this data set is that it illustrates some of the more common pitfalls in interpreting regression models.

Just because a coefficient is significant, doesn't mean your covariate causes your response

- This is the old adage that correlation does not imply causation. In this example, we have strong evidence that higher police expenditures are positively associated with crime rates. This doesn't mean that decreasing police expenditure will lower crime rate. The relationship is not causal – at least not in that direction. A more reasonable explanation is that higher crime rates promt policy makers to increase police expenditure.

There's a difference between practical significance and statistical significance

- Both `average.ed` and `exp.per.cap.1960` are statistically significant. `exp.per.cap.1960` has a much more significant p-value, but also a much smaller coefficient. When looking at your regression model, you shouldn't just look at the p-value column. The really interesting covariates are the ones that are significant, but also have the largest effect.

Note also that the units of measurement should be taken into account when thinking about coefficient estimates and effect sizes. Suppose, for example, that we regressed income (measured in $) on height and got a coefficient estimate of 100, with a standard error of 20. Is 100 a large effect? *The answer depends on the units of measurement.* If height had been measured in metres, we would be saying that every 1m increase in height is associated on average with a $100 increase in income. That’s too small an effect for us to care about. Now what if height was measured in mm? Then we’d be saying that every 1mm increase in height is associated on average with a $100 increase in income. Since 1inch = 25.4mm, this means that every 1inch difference in height is on average associated with a $2540 difference in income. This would be a tremendously large effect. **Moral of the story**: Whether an effect is ‘practically significant’ depends a lot on the unit of measurement.

Assessing significance of factors and interactions in regression

Factors in linear regression

Interpreting coefficients of factor variables

In the case of quantitative predictors, we’re more or less comfortable with the interpretation of the linear model coefficient as a “slope” or a “unit increase in outcome per unit increase in the covariate”. This isn’t the right interpretation for factor variables. In particular, the notion of a slope or unit change no longer makes sense when talking about a categorical variable. E.g., what does it even mean to say “unit increase in major” when studying the effect of college major on future earnings?

To understand what the coefficients really mean, let’s go back to the birthwt data and try regressing birthweight on mother’s race and mother’s age.

```
# Fit regression model
birthwt.lm <- lm(birthwt.grams ~ race + mother.age, data = birthwt)

# Regression model summary
summary(birthwt.lm)
```

```
##
## Call:
## lm(formula = birthwt.grams ~ race + mother.age, data = birthwt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2131.57  -488.02    -1.16   521.87  1757.07
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2584.264    258.393  10.001   <2e-16 ***
## raceother     80.249    165.582   0.485    0.628
## racewhite    365.715    160.636   2.277    0.024 *
## mother.age     6.288     10.073   0.624    0.533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 715.7 on 185 degrees of freedom
## Multiple R-squared:  0.05217,    Adjusted R-squared:  0.0368
## F-statistic: 3.394 on 3 and 185 DF,  p-value: 0.01909
```

Note that there are two coefficients estimated for the race variable (`raceother` and `racewhite`). What’s happening here?

When you put a factor variable into a regression, you’re allowing a **different intercept at every level of the factor**. In the present example, you’re saying that you want to model `birthwt.grams` as

Baby’s birthweight = Intercept(based on mother’s race) + β * mother’s age

We can rewrite this more succinctly as:

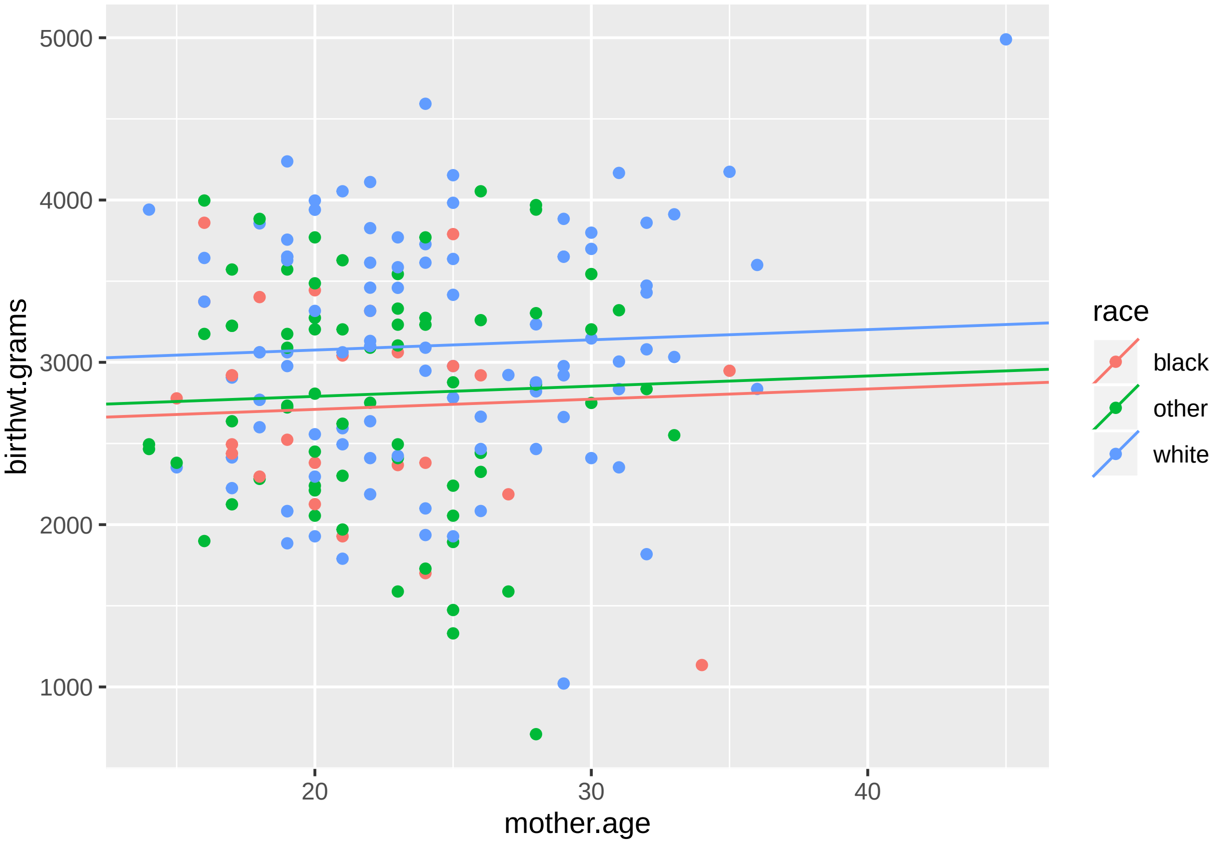
$$y = \text{Intercept}_{race} + \beta \times \text{age}$$

Essentially you're saying that your data is broken down into 3 racial groups, and you want to model your data as having the same slope governing how birthweight changes with mother's age, but potentially different intercepts. Here's a picture of what's happening.

```
# Calculate race-specific intercepts
intercepts <- c(coef(birthwt.lm)["(Intercept)"],
                coef(birthwt.lm)["(Intercept)"] + coef(birthwt.lm)["raceother"],
                coef(birthwt.lm)["(Intercept)"] + coef(birthwt.lm)["racewhite"])

lines.df <- data.frame(intercepts = intercepts,
                       slopes = rep(coef(birthwt.lm)["mother.age"], 3),
                       race = levels(birthwt$race))

qplot(x = mother.age, y = birthwt.grams, color = race, data = birthwt) +
  geom_abline(aes(intercept = intercepts,
                  slope = slopes,
                  color = race), data = lines.df)
```



How do we interpret the 2 race coefficients? For categorical variables, the interpretation is relative to the given baseline. The baseline is just whatever level comes first (here, "black"). E.g., the estimate of `raceother` means that the estimated intercept is 80.2492209 higher among "other" race mothers compared to black mothers. Similarly, the estimated intercept is 365.7150005 higher for white mothers than black mothers.

> Another way of putting it: Among mothers of the same age, babies of white mothers are born on average weighing 365.7g more than babies of black mothers.

Why is one of the levels missing in the regression?

As you've already noticed, there is no coefficient called "raceblack" in the estimated model. This is because this coefficient gets absorbed into the overall (Intercept) term.

Let's peek under the hood. Using the `model.matrix()` function on our linear model object, we can get the data matrix that underlies our regression. Here are the first 20 rows.

```r
head(model.matrix(birthwt.lm), 20)
```

```
##     (Intercept) raceother racewhite mother.age
## 85            1         0         0         19
## 86            1         1         0         33
## 87            1         0         1         20
## 88            1         0         1         21
## 89            1         0         1         18
## 91            1         1         0         21
## 92            1         0         1         22
## 93            1         1         0         17
## 94            1         0         1         29
## 95            1         0         1         26
## 96            1         1         0         19
## 97            1         1         0         19
## 98            1         1         0         22
## 99            1         1         0         30
## 100           1         0         1         18
## 101           1         0         1         18
## 102           1         0         0         15
## 103           1         0         1         25
## 104           1         1         0         20
## 105           1         0         1         28
```

Even though we think of the regression `birthwt.grams ~ race + mother.age` as being a regression on two variables (and an intercept), it's actually a regression on 3 variables (and an intercept). This is because the `race` variable gets represented as two dummy variables: one for `race == other` and the other for `race == white`.

Why isn't there a column for representing the indicator of `race == black`? This gets back to our colinearity issue. By definition, we have that

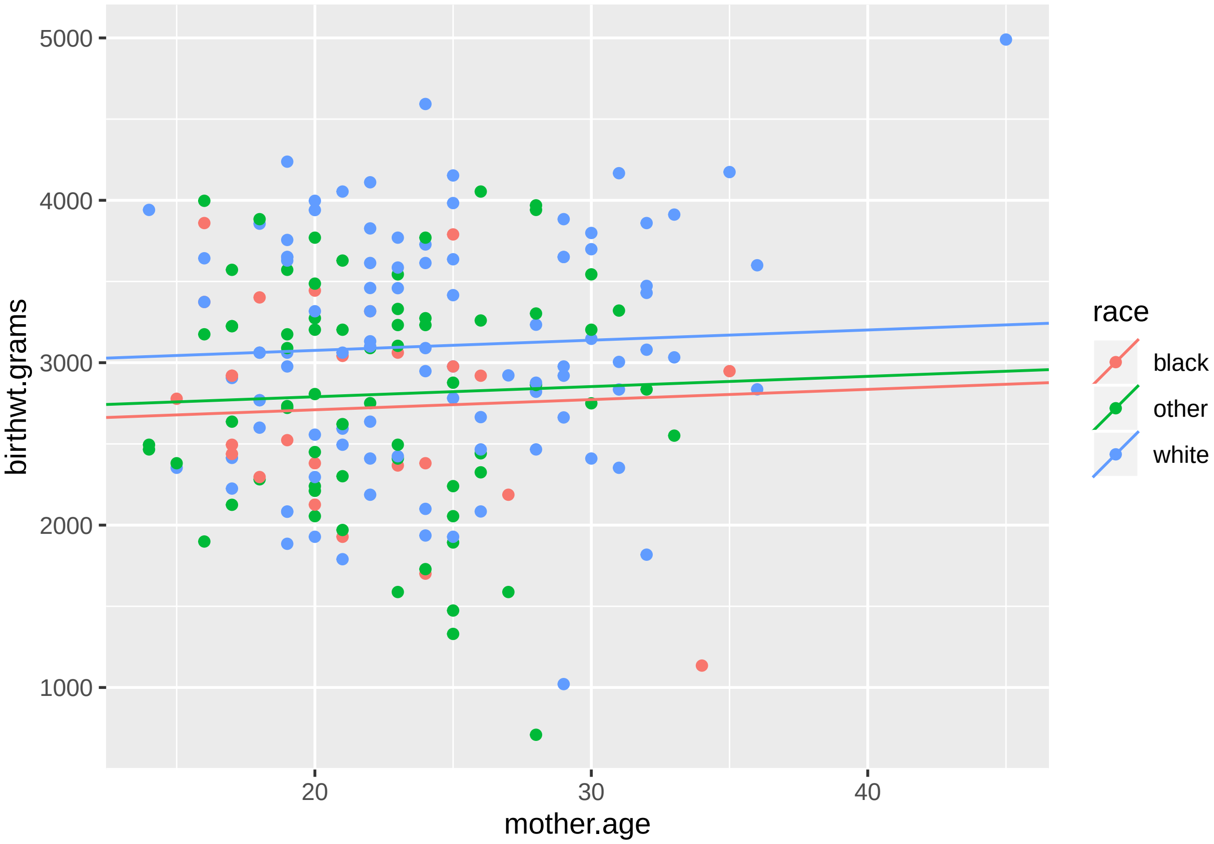
raceblack + raceother + racewhite = 1 = (Intercept)

This is because for every observation, one and only one of the race dummy variables will equal 1. Thus the group of 4 variables {raceblack, raceother, racewhite, (Intercept)} is perfectly colinear, and we can't include all 4 of them in the model. The default behavior in R is to remove the dummy corresponding to the first level of the factor (here, raceblack), and to keep the rest.

Interaction terms

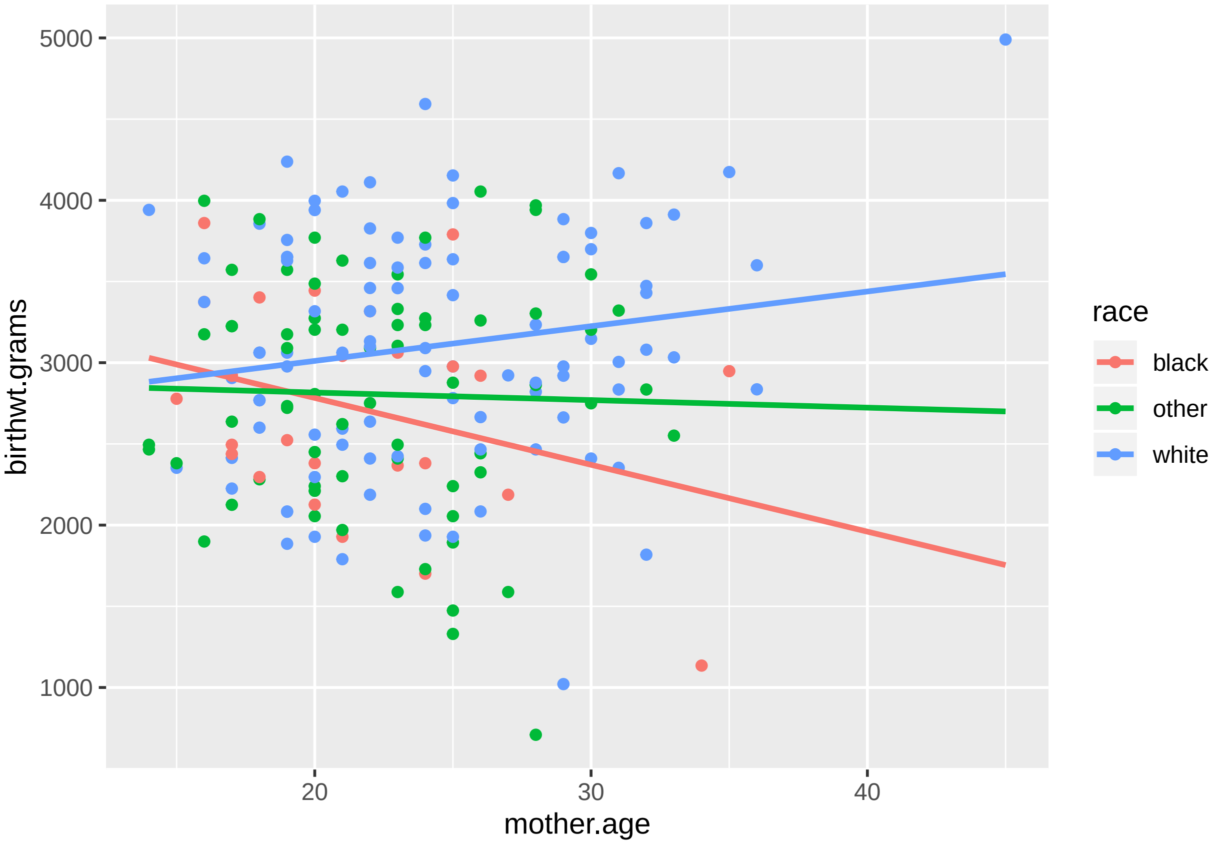
Let's go back to the regression line plot we generated above.

```r
qplot(x = mother.age, y = birthwt.grams, color = race, data = birthwt) +
  geom_abline(aes(intercept = intercepts,
                  slope = slopes,
                  color = race), data = lines.df)
```



We have seen similar plots before by using the `geom_smooth` or `stat_smooth` commands in `ggplot`. Compare the plot above to the following.

```
qplot(x = mother.age, y = birthwt.grams, color = race, data = birthwt) +
  stat_smooth(method = "lm", se = FALSE, fullrange = TRUE)
```



In this case we have not only race-specific intercepts, but also **race-specific slopes**. The plot above corresponds to the model:

Baby's birthweight = Intercept(based on mother's race) + β(based on mother's race) * mother's age

We can rewrite this more succinctly as:

$$y = \text{Intercept}_{race} + \beta_{race} \times \text{age}$$

To specify this interaction model in R, we use the following syntax

```
birthwt.lm.interact <- lm(birthwt.grams ~ race * mother.age, data = birthwt)

summary(birthwt.lm.interact)
```

```
##
## Call:
## lm(formula = birthwt.grams ~ race * mother.age, data = birthwt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2182.35  -474.23    13.48   523.86  1496.51
##
## Coefficients:
##                        Estimate Std. Error t value    Pr(>|t|)
## (Intercept)             3606.33     615.26   5.861 0.000000021 ***
## raceother               -696.74     756.65  -0.921      0.3584
## racewhite              -1022.79     694.21  -1.473      0.1424
## mother.age              -41.17      27.82  -1.480      0.1407
## raceother:mother.age     36.51      33.85   1.078      0.2823
## racewhite:mother.age     62.54      30.67   2.039      0.0429 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 710.7 on 183 degrees of freedom
## Multiple R-squared:  0.07541,    Adjusted R-squared:  0.05015
## F-statistic: 2.985 on 5 and 183 DF,  p-value: 0.01291
```

We now have new terms appearing. Terms like `racewhite:mother.age` are deviations from the baseline slope (the coefficient of `mother.age` in the model) in the same way that terms like `racewhite` are deviations from the baseline intercept. This models says that:

> On average among black mothers, every additional year of age is associated with a 41.2g decrease in the birthweight of the baby.

To get the slope for white mothers, we need to add the interaction term to the baseline.

$$\begin{aligned}\beta_{racewhite} &= \beta_{raceblack} + \beta_{racewhite:mother.age} \\ &= \text{mother.age} + \text{racewhite:mother.age} \\ &= -41.2 + 62.5 \\ &= 21.4\end{aligned}$$

This slope estimate is positive, which agrees with the regression plot above.

Is a categorical variable in a regression statistically significant?

Last class we considered modelling birthweight as a linear function of mother's age, allowing for a race-specific intercept for each of the three race categories. This model is fit again below.

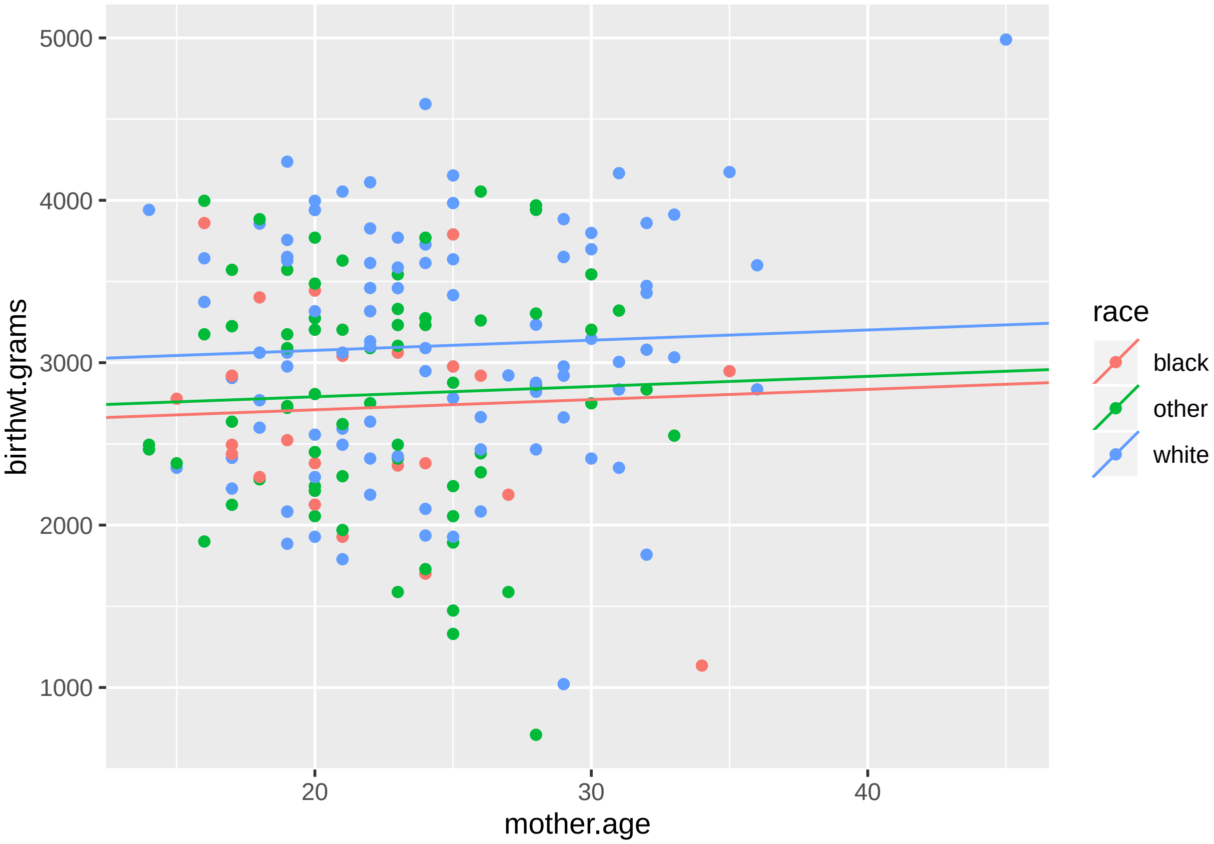
```
birthwt.lm <- lm(birthwt.grams ~ race + mother.age, data = birthwt)
```

Here's a visualization of the model fit that we wound up with. Note that while there are 3 lines shown, this is a visualization of just one model: `birthwt.grams ~ race + mother.age`. This model produces 3 lines because the coefficients of the `race` variable result in different intercepts.

```
# Calculate race-specific intercepts
intercepts <- c(coef(birthwt.lm)["(Intercept)"],
                coef(birthwt.lm)["(Intercept)"] + coef(birthwt.lm)["raceother"],
                coef(birthwt.lm)["(Intercept)"] + coef(birthwt.lm)["racewhite"])

lines.df <- data.frame(intercepts = intercepts,
                       slopes = rep(coef(birthwt.lm)["mother.age"], 3),
                       race = levels(birthwt$race))

qplot(x = mother.age, y = birthwt.grams, color = race, data = birthwt) +
  geom_abline(aes(intercept = intercepts,
                  slope = slopes,
                  color = race), data = lines.df)
```



At this stage we may be interested in assessing whether the `race` variable is statistically significant. i.e., Does including the `race` variable significantly improve the fit of our model, or is the simpler model `birthwt.grams ~ mother.age` just as good?

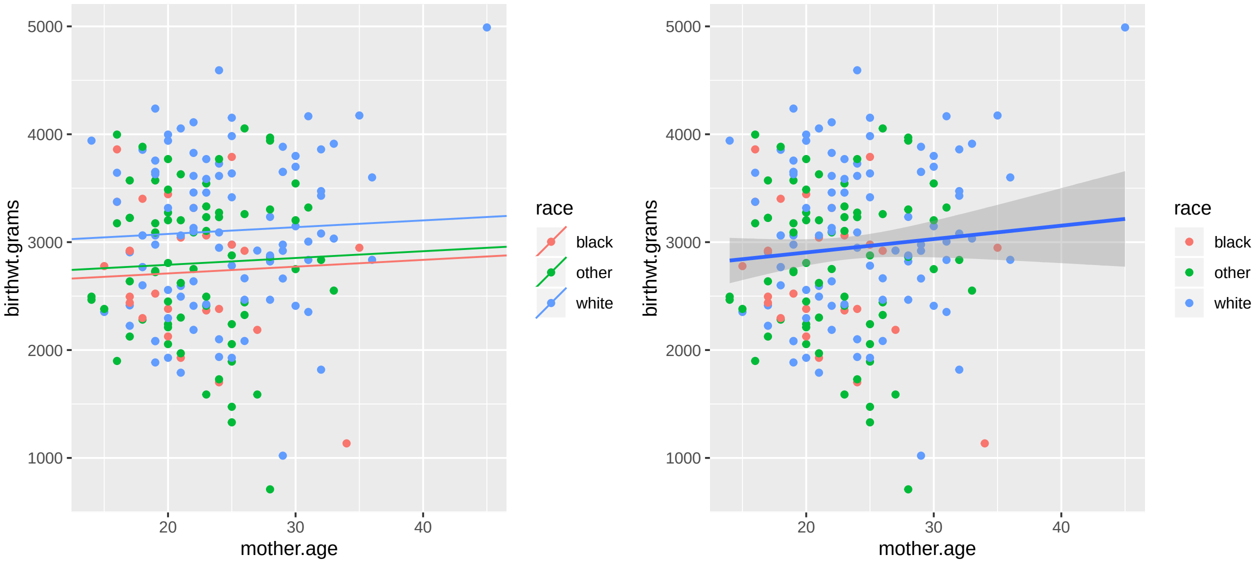
Essentially, we want to know if the race-specific intercepts capture significantly more variation in the outcome (birthweight) than the single intercept model, or if allowing for different intercepts isn't doing much more than capturing random fluctuations in the data.

Here's a picture of the two models we're comparing:

```r
library(gridExtra)
plot.complex <- qplot(x = mother.age, y = birthwt.grams,
                      color = race, data = birthwt) +
  geom_abline(aes(intercept = intercepts,
                  slope = slopes,
                  color = race), data = lines.df)

# Single intercept model (birthwt.grams ~ mother.age)
p <- ggplot(birthwt, aes(x = mother.age, y = birthwt.grams))
plot.simple <- p + geom_point(aes(colour = race)) + stat_smooth(method = "lm")

grid.arrange(plot.complex, plot.simple, ncol = 2)
```



To test this hypothesis, we use the `anova` function (not to be confused with the `aov` function). This function compares two **nested** models, accounting for their residual sums of squares (how well they fit the data) and their complexity (how many more variables are in the larger model) to assess statistical significance.

```
# Fit the simpler model with mother.age as the only predictor
birthwt.lm.simple <- lm(birthwt.grams ~ mother.age, data = birthwt)

# Compare to more complex model
anova(birthwt.lm.simple, birthwt.lm)
```

```
## Analysis of Variance Table
##
## Model 1: birthwt.grams ~ mother.age
## Model 2: birthwt.grams ~ race + mother.age
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1    187 99154173
## 2    185 94754346  2   4399826 4.2951 0.01502 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This output tells us that the `race` variable is statistically significant: It is unlikely that the improvement in fit when the add the `race` variable is simply due to random fluctuations in the data. Thus it is important to consider race when modeling how birthweight depends on the mother's age.

Is an interaction term significant?

Assessing significance of interaction terms operates on the same principle. We once again ask whether the improvement in model fit is worth the increased complexity of our model. For instance, consider the example we saw last class, where we allowed for a race-specific slope in addition to the race-specific intercept from before.

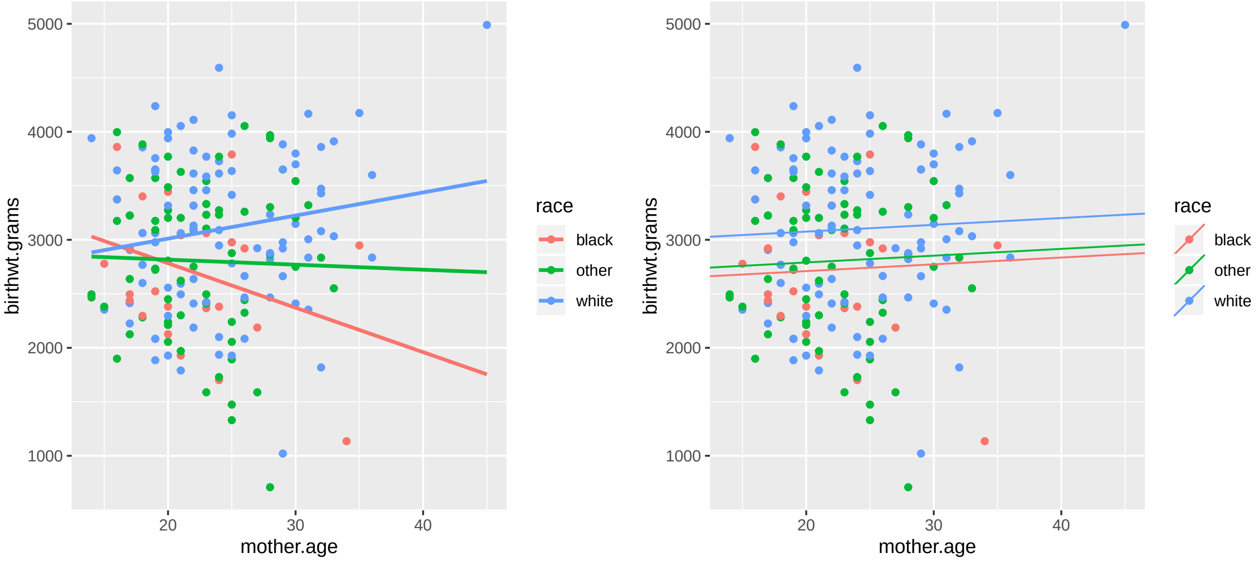
```
birthwt.lm.interact <- lm(birthwt.grams ~ race * mother.age, data = birthwt)
summary(birthwt.lm.interact)
```

```
##
## Call:
## lm(formula = birthwt.grams ~ race * mother.age, data = birthwt)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2182.35  -474.23    13.48   523.86  1496.51 
## 
## Coefficients:
##                       Estimate Std. Error t value    Pr(>|t|)    
## (Intercept)            3606.33     615.26   5.861 0.000000021 ***
## raceother              -696.74     756.65  -0.921      0.3584    
## racewhite             -1022.79     694.21  -1.473      0.1424    
## mother.age              -41.17      27.82  -1.480      0.1407    
## raceother:mother.age     36.51      33.85   1.078      0.2823    
## racewhite:mother.age     62.54      30.67   2.039      0.0429 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 710.7 on 183 degrees of freedom
## Multiple R-squared:  0.07541,    Adjusted R-squared:  0.05015 
## F-statistic: 2.985 on 5 and 183 DF,  p-value: 0.01291
```

Here's a side-by-side visual comparison of the `race + mother.age` model and the `race + mother.age + race*mother.age` interaction model.

```r
plot.interact <- qplot(x = mother.age, y = birthwt.grams, color = race, data = birthwt) + stat_smooth(me
grid.arrange(plot.interact, plot.complex, ncol = 2)
```



So, do the lines with different slopes fit the data significantly better than the common slope model? Let's compare the two with the `anova()` function.

```r
anova(birthwt.lm, birthwt.lm.interact)
```

```
## Analysis of Variance Table
## 
## Model 1: birthwt.grams ~ race + mother.age
## Model 2: birthwt.grams ~ race * mother.age
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    185 94754346                           
## 2    183 92431148  2   2323199 2.2998 0.1032
```

This p-value turns out to not be statistically significant. So even though the estimated slopes in the interaction model look very different, our estimates are quite variable, so we don't have enough evidence to conclude that the interaction term (different slopes) is providing significant additional explanatory power over the simpler `race + mother.age` model.

Is my complex model signficantly better than a simpler one?

The testing strategy above applies to any two nested models. Here's an example where we add in a few more variables and see how it compares to the `race + mother.age` model from earlier.

```r
birthwt.lm.complex <- lm(birthwt.grams ~ mother.smokes + physician.visits + race + mother.age, data = bi

summary(birthwt.lm.complex)
```

```
##
## Call:
## lm(formula = birthwt.grams ~ mother.smokes + physician.visits +
##     race + mother.age, data = birthwt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2335.06  -455.16    31.74   499.29  1623.57
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2838.0676   258.2194  10.991  < 2e-16 ***
## mother.smokesyes -424.6512   110.3715  -3.847 0.000165 ***
## physician.visits   14.3913    48.9525   0.294 0.769102
## raceother          -0.8211   161.8314  -0.005 0.995957
## racewhite         444.3396   156.5860   2.838 0.005057 **
## mother.age          1.5474     9.9965   0.155 0.877155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 691.7 on 183 degrees of freedom
## Multiple R-squared:  0.1241, Adjusted R-squared:  0.1001
## F-statistic: 5.184 on 5 and 183 DF,  p-value: 0.000179
```

Let's compare to our earlier model:

```r
anova(birthwt.lm, birthwt.lm.complex)
```

```
## Analysis of Variance Table
##
## Model 1: birthwt.grams ~ race + mother.age
## Model 2: birthwt.grams ~ mother.smokes + physician.visits + race + mother.age
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1    185 94754346
## 2    183 87567280  2   7187067 7.5098 0.0007336 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Highly significant! This is probably due to the fact that mother's smoking status has a tremendously high association with birthweight.