Empirical Example: Child Test Scores

Econ 103 – Lecture 23

Introduction

This is our final lecture! Today we're going to bring together all the skills you've learned over the course of the semester to work through a real-world example in which we try to predict which children are most at-risk of low test scores based on information about their mothers. Although this is a very simple example, regressions like these are frequently used to inform policy. For example, if we want to design an intervention to target disadvantaged children, it would be very valuable to be able to predict which children are most in need *before* we see their test scores at age three since the interventions that make the greatest impact are those adminstered in early childhood.

Getting Started

To make R's regression output easier to read, we'll the `display` function that I discussed in our previous lecture:

```r
source("http://ditraglia.com/econ103/display.R")
```

You'll need to re-run this if you restart R and want to continue using `display`.
To start, let's load the student test score data from my website.

```r
data_url <- "http://ditraglia.com/econ103/child_test_data.csv"
child <- read.csv(data_url)
```

Now let's take a quick look at the dataset

```r
head(child)

##   kid.score mom.hs    mom.iq mom.age
## 1        65      1 121.11753      27
## 2        98      1  89.36188      25
## 3        85      1 115.44316      27
## 4        83      1  99.44964      25
## 5       115      1  92.74571      27
## 6        98      0 107.90184      18
```

Here's a description of all of the columns in this dataframe: As you can see, we have

Variable Name	Description
`kid.score`	Child's Test Score at Age 3
`mom.age`	Age of Mother at Birth of Child
`mom.hs`	Mother Completed High School? (1 = Yes)
`mom.iq`	Mother's IQ Score

a lot of information here! For today, we'll only use the columns `kid.score`, `mom.hs` and `mom.iq`. In the extension problem for this lecture you'll look at `mom.age`,

First Regression: `mom.hs`

This regression compares average test scores of children whose mother completed high school to those whose mother didn't. Here, `mom.hs` is a *dummy variable*: it takes on the value 1 if that child's mother completed high school, 0 otherwise.

```
reg1 <- lm(kid.score ~ mom.hs, data = child)
display(reg1)

## lm(formula = kid.score ~ mom.hs, data = child)
##             coef.est coef.se
## (Intercept) 77.55     2.06
## mom.hs      11.77     2.32
## ---
## n = 434, k = 2
## residual sd = 19.85, R-Squared = 0.06
```

Rounding, we can summarize these regression results as follows:

$$\texttt{kid.score} = 78 + 12 \cdot \texttt{mom.hs} + \text{error}$$

Since `mom.hs` is a dummy variable, taking on the value 1 if a child's mother completed high school and 0 otherwise, this regression is the *same thing* as comparing the mean test scores of two groups: those whose mother completed high school and those whose mother didn't!

$$\begin{aligned}(\texttt{mom.hs} = 1) \Rightarrow \texttt{kid.score} &= 78 + 12 \cdot 1 + \text{error}\\ &= 90 + \text{error}\end{aligned}$$

$$\begin{aligned}(\texttt{mom.hs} = 0) \Rightarrow \texttt{kid.score} &= 78 + 12 \cdot 0 + \text{error}\\ &= 78 + \text{error}\end{aligned}$$

The difference of means simply equals the coefficient on `mom.hs`, namely 12. Creating a confidence interval for this difference of means is easy, since R has already calculated the required standard error for us. Rounding, this value is approximately 2.3, so our approximate 95% confidence interval for the difference of means (the coefficient on `mom.hs`) is 12 ± 4.6, in other words $(7.4, 16.6)$. Since this interval doesn't include zero, we would reject the null that children whose mothers completed high school have the same test scores on average as those whose mothers didn't against the two-sided alternative at the 5% significance level. It looks like children whose mothers completed high school do better on this test.

Second Regression: `mom.iq`

Now let's try something different. We'll use mother IQ to predict child test scores.

```r
reg2 <- lm(kid.score ~ mom.iq, data = child)
display(reg2)

## lm(formula = kid.score ~ mom.iq, data = child)
##             coef.est coef.se
## (Intercept) 25.80     5.92
## mom.iq       0.61     0.06
## ---
## n = 434, k = 2
## residual sd = 18.27, R-Squared = 0.20
```

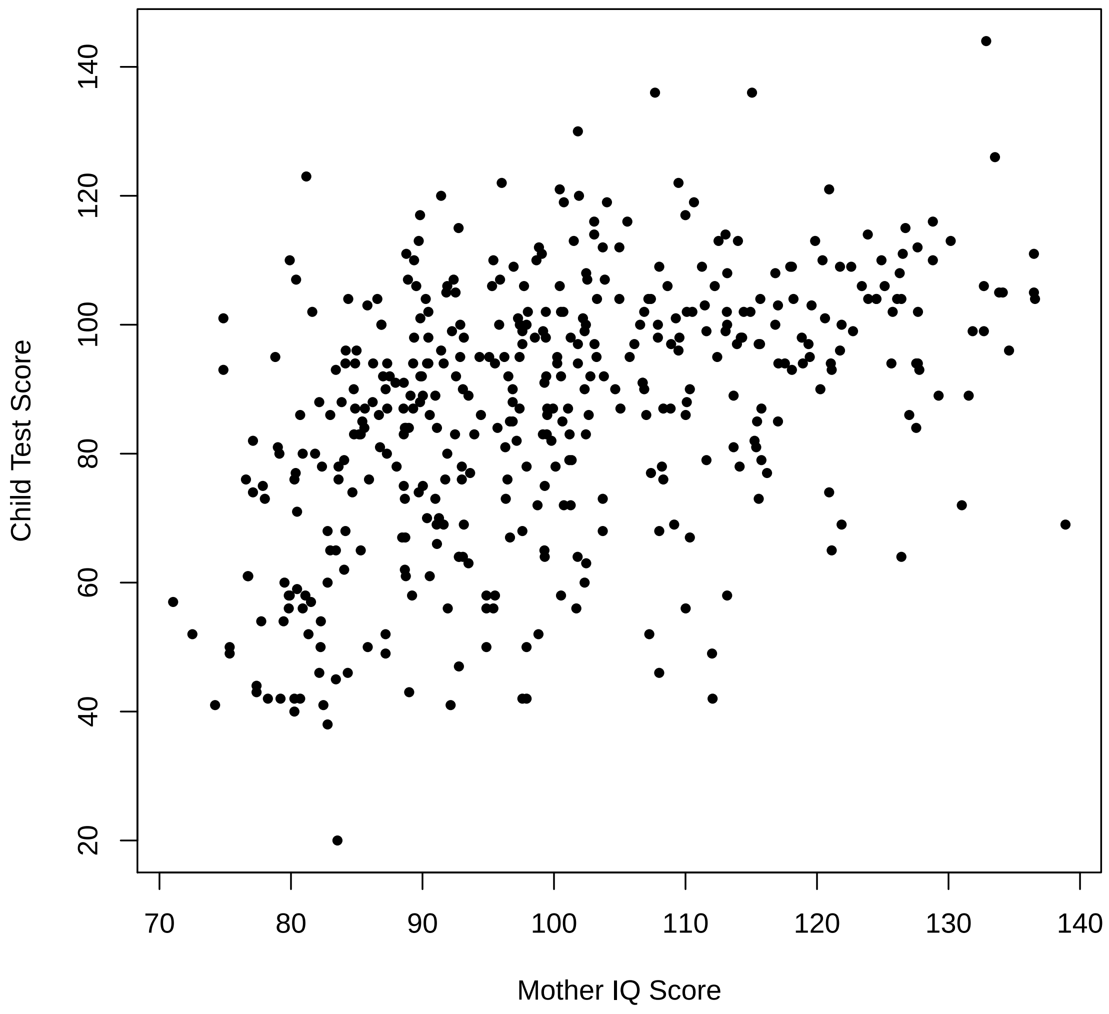
Rounding, we can summarize the results as follows:

$$\texttt{kid.score} = 26 + 0.6 \cdot \texttt{mom.iq} + \text{error}$$

The intercept in this model is not interpretable: it is the predicted test score for a child whose mother has an IQ of zero! The coefficient on `mom.iq` is meaningful, however. If we compare two groups of students who differ by one point in mothers' IQ, we would predict that the group with higher mother IQ will score 0.6 points higher, on average.

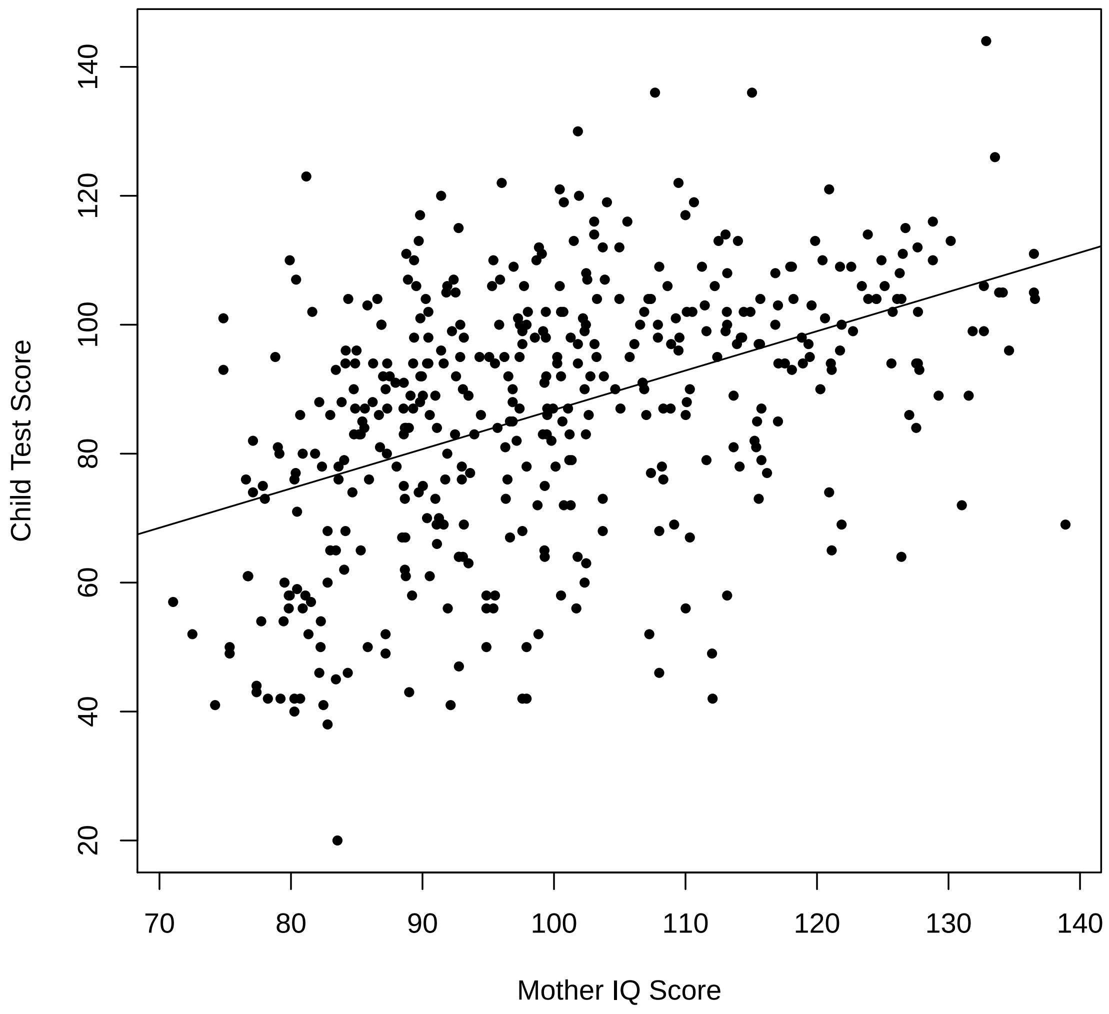
We can plot the data as follows

```r
plot(child$mom.iq, child$kid.score, pch = 20, xlab = 'Mother IQ Score',
     ylab = 'Child Test Score')
```



To add the regression line, we need to extract the slope and intercept from the fitted regression:

```
plot(child$mom.iq, child$kid.score, pch = 20, xlab = 'Mother IQ Score',
     ylab = 'Child Test Score')
abline(coef(reg2))
```



Third Regression: `mom.hs` and `mom.iq`

Now we'll fit a regression with both `mom.hs` and `mom.iq`. It turns out that this allows the regression line to have a different *intercept* depending on whether a child's mother completed high school.

```
reg3 <- lm(kid.score ~ mom.hs + mom.iq, data = child)
display(reg3)

## lm(formula = kid.score ~ mom.hs + mom.iq, data = child)
##             coef.est coef.se
## (Intercept) 25.73     5.88
## mom.hs       5.95     2.21
## mom.iq       0.56     0.06
## ---
## n = 434, k = 3
## residual sd = 18.14, R-Squared = 0.21
```

Rounding, we can summarize the fitted model as follows:

$$\texttt{kid.score} = 26 + 6 \cdot \texttt{mom.hs} + 0.6 \cdot \texttt{mom.iq} + \text{error}$$

Since `mom.hs` is binary, this is equivalent to letting each group have a regression line with a *different intercept* but the same slope:

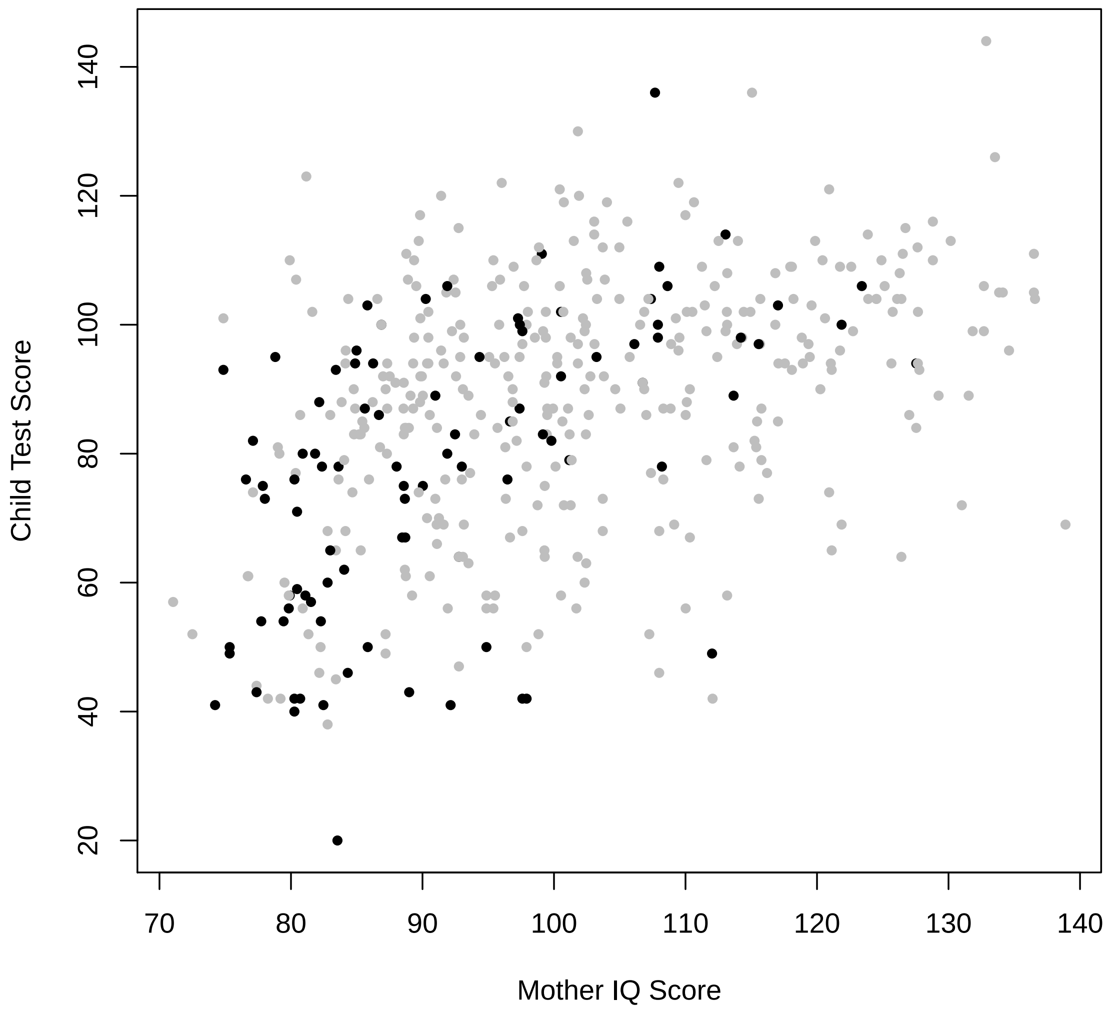
$$\begin{aligned}(\texttt{mom.hs = 1}) \Rightarrow \texttt{kid.score} &= 26 + 6 \cdot 1 + 0.6 \cdot \texttt{mom.iq} + \text{error}\\ &= 32 + 0.6 \cdot \texttt{mom.iq} + \text{error}\end{aligned}$$

$$\begin{aligned}(\texttt{mom.hs = 0}) \Rightarrow \texttt{kid.score} &= 26 + 6 \cdot 0 + 0.6 \cdot \texttt{mom.iq} + \text{error}\\ &= 26 + 0.6 \cdot \texttt{mom.iq} + \text{error}\end{aligned}$$

In this case the intercept is not interpretable: it corresponds to the average test score for children whose mother did not complete high school and have a zero IQ! The other two coefficients, however, are meaningful. The coefficient on `mom.hs` compares test scores for children whose mothers have the same IQ but differ in whether or not they completed high school. The coefficient on `mom.iq` compares children whose mothers have the same value of `mom.iq` but differ in IQ by one point.

We can plot the data and fitted models as follows. We'll plot the children whose mothers went to high school in *gray* and those whose mothers didn't in *black*.

```
colors <- ifelse(child$mom.hs == 0, "black", "gray")
plot(child$mom.iq, child$kid.score, pch = 20, xlab = 'Mother IQ Score',
     ylab = 'Child Test Score', col = colors)
```



To add the fitted regression lines we need to extract the common slope as well as the intercept for *each group*

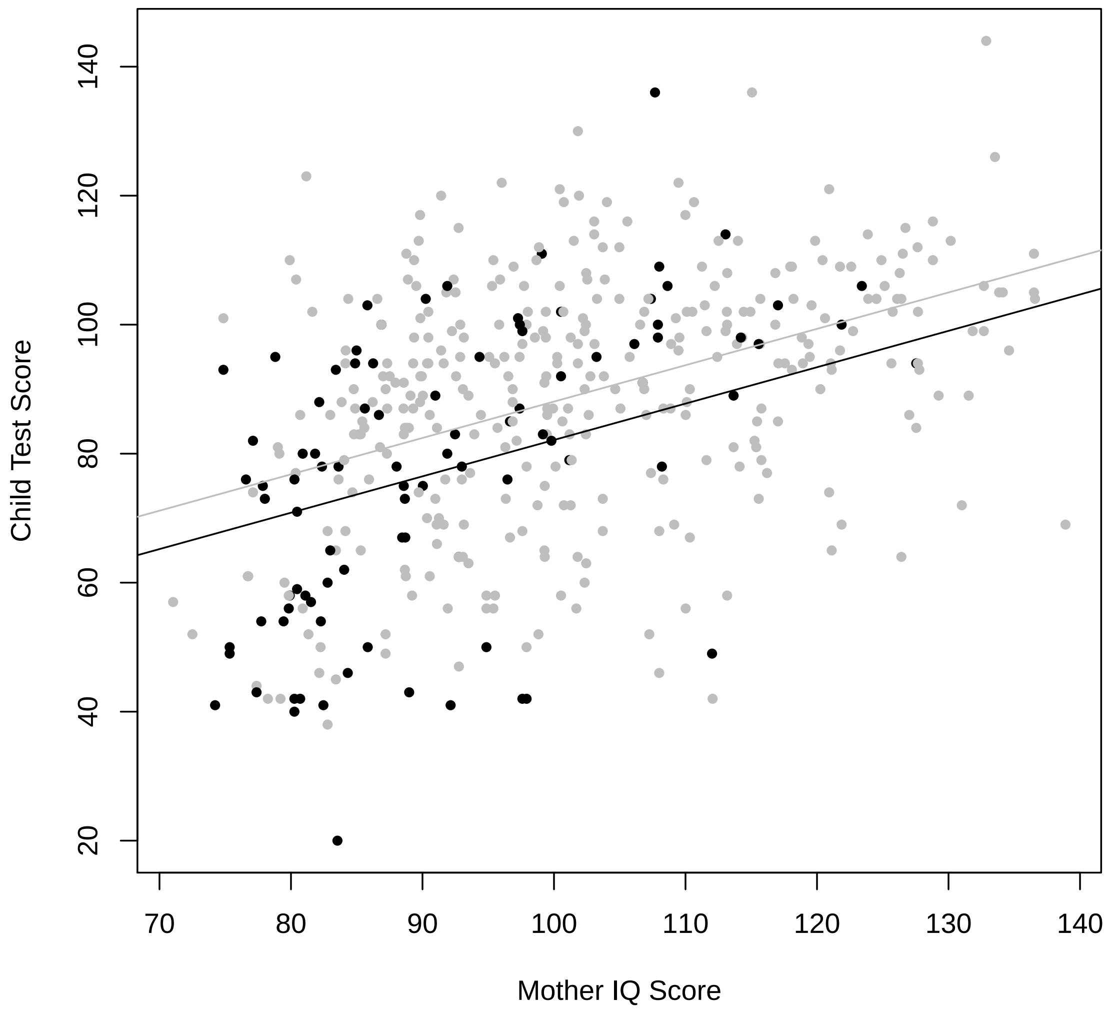
```
coef(reg3)

## (Intercept)      mom.hs      mom.iq
##   25.731538    5.950117    0.563906

b_both <- coef(reg3)[3]
a_HS <- coef(reg3)[1] + coef(reg3)[2]
a_no_HS <- coef(reg3)[1]
```

Now we plot the results alongside the two regression lines:

```r
colors <- ifelse(child$mom.hs == 0, "black", "gray")
plot(child$mom.iq, child$kid.score, pch = 20, xlab = 'Mother IQ Score',
     ylab = 'Child Test Score', col = colors)
abline(a = a_HS, b = b_both, col = 'gray')
abline(a = a_no_HS, b = b_both, col = 'black')
```



Fourth Regression: `mom.hs`, `mom.iq` and `mom.hs:mom.iq`

Now we'll add an interaction between `mom.hs` and `mom.iq`: that is, we'll include a predictor whose value equals the *product* of these two variables. The way to express this in R is `mom.hs:mom.iq`. Consider a child whose mother completed high school (`mom.hs`=1). For this child, `mom.hs:mom.iq` $= 1 \cdot$ `mom.iq` = `mom.iq`. For a child whose mother did *not* complete high school, `mom.hs:mom.iq` $= 0 \cdot$ `mom.iq` $= 0$. It turns out that adding this interaction allows the two groups (those whose mother completed high school and those whose mother did not) to have regression lines with different slopes. Since `mom.iq` is included as in its own right, we allow different intercepts as well.

First we'll run the regression:

```
reg4 <- lm(kid.score ~ mom.hs + mom.iq + mom.hs:mom.iq, data = child)
display(reg4)

## lm(formula = kid.score ~ mom.hs + mom.iq + mom.hs:mom.iq, data = child)
##                coef.est coef.se
## (Intercept)   -11.48    13.76
## mom.hs         51.27    15.34
## mom.iq          0.97     0.15
## mom.hs:mom.iq  -0.48     0.16
## ---
## n = 434, k = 4
## residual sd = 17.97, R-Squared = 0.23
```

Rounding, we can summarize the results as follows:

$$\begin{aligned}
(\texttt{mom.hs = 1}) \Rightarrow \texttt{kid.score} &= -11 + 51 \cdot 1 + 1 \cdot \texttt{mom.iq} - 0.5 \cdot 1 \cdot \texttt{mom.iq} + \text{error} \\
&= 40 + 0.5 \cdot \texttt{mom.iq} + \text{error}
\end{aligned}$$

$$\begin{aligned}
(\texttt{mom.hs = 0}) \Rightarrow \texttt{kid.score} &= -11 + 51 \cdot 0 + 1 \cdot \texttt{mom.iq} - 0.5 \cdot 0 \cdot \texttt{mom.iq} + \text{error} \\
&= -11 + 1 \cdot \texttt{mom.iq} + \text{error}
\end{aligned}$$

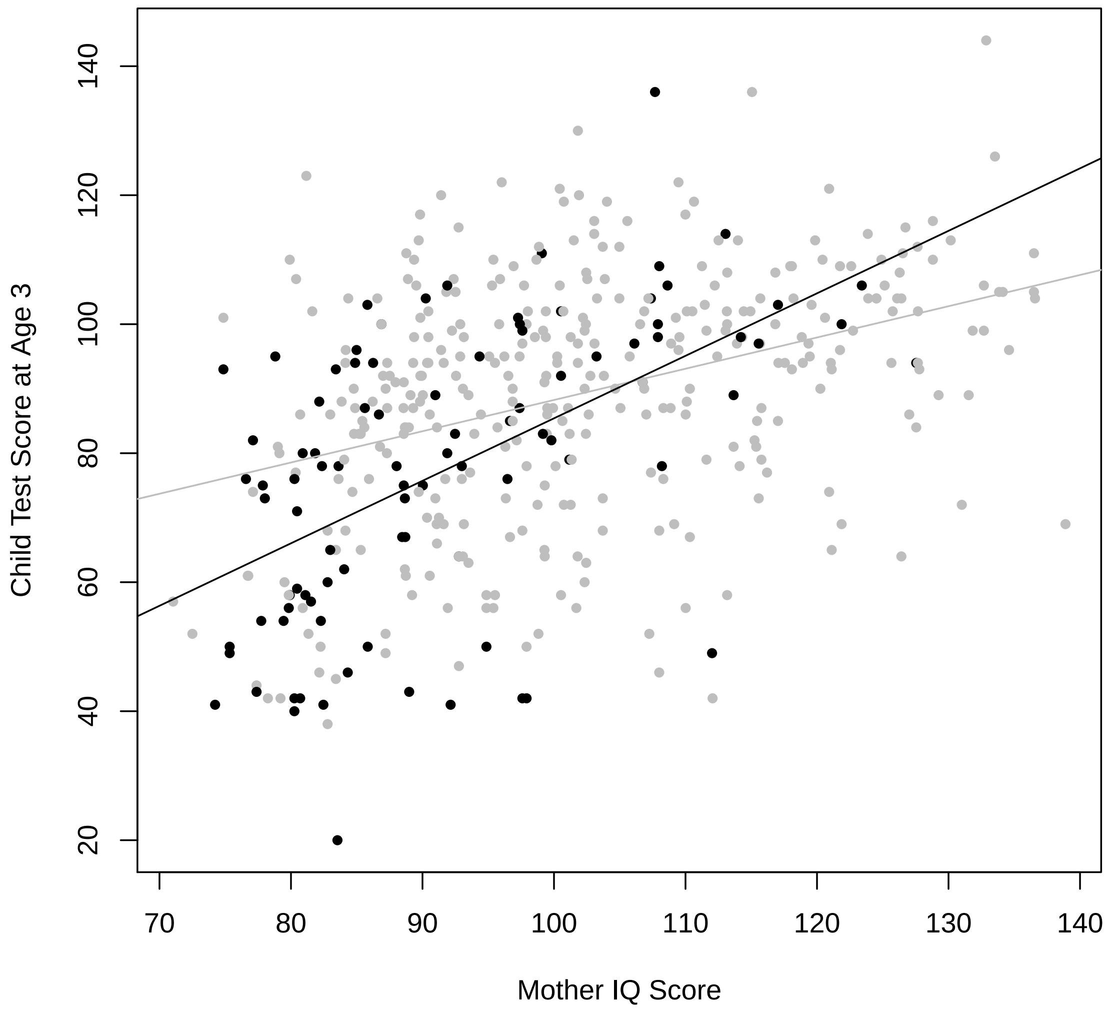
We can plot the two regression lines as follows. This time we need to allow different slopes *as well as intercepts.*

```
coef(reg4)

##   (Intercept)        mom.hs        mom.iq mom.hs:mom.iq
##   -11.4820211    51.2682234     0.9688892    -0.4842747

b_HS <- coef(reg4)[3] + coef(reg4)[4]
b_no_HS <- coef(reg4)[3]
```

```r
a_HS <- coef(reg4)[1] + coef(reg4)[2]
a_no_HS <- coef(reg4)[1]
colors <- ifelse(child$mom.hs == 0, "black", "gray")
plot(child$mom.iq, child$kid.score, pch = 20, xlab = 'Mother IQ Score',
     ylab = 'Child Test Score at Age 3', col = colors)
abline(a = a_HS, b = b_HS, col = 'gray')
abline(a = a_no_HS, b = b_no_HS, col = 'black')
```


Child Test Score at Age 3
Mother IQ Score