Section materials: More Regression with Maternal Smoking and Infant Birthweight (solutions)

scott cunningham

Background

This week we deepen our understanding of the mechanics of Ordinary Least Squares (OLS). We'll continue using the **Cattaneo (2010)** dataset that examines the effect of maternal smoking (`mbsmoke`) on infant birthweight (`bweight`), but we will also look at other variables so that students have practice understanding the mechanics of OLS in a context that they are by now familiar with. Our goal is to **see how OLS solves for coefficients** without matrix algebra—first by hand with simple formulas, then by confirming that lm() produces identical results.

Reviewing with bivariate regression

Question 1 (4 points)

First, we will begin by importing the dataset from Cunningham's github repo with the `read_dta` command from the `haven` package. Then we will regress `bweight` onto `mbsmoke`. They did this last week, but I think they need to keep doing it, keep practicing the interpretation.

```
data <- read_dta("https://raw.github.com/scunning1975/mixtape/master/cattaneo2.dta")

fit1 <- lm(bweight ~ mbsmoke, data = data)
tidy(fit1)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    3413.      9.26     369.  0
## 2 mbsmoke        -275.     21.5      -12.8 4.68e-37
```

Notice that we immediately estimated the effect of maternal smoking on birthweight. Students are encouraged at this point to correctly interpret the coefficient. I would like for you to help them understand that the regression coefficient when the treatment is a dummy is interpreted as **percentage point differences**. It's important that they learn to use the correct units – that these are **percentage points**, not **percentage changes**.

1. Interpret the slope and the intercept.

As with last week, the intercept measures the average outcome for the units whose `mbsmoke=0`, which in this case is the mothers who did not smoke during pregnancy. The slope is the coefficient on `mbsmoke` in the regression and measures the difference in the average outcomes between those who smoked (`mbsmoke=1`) and those did not (`mbsmoke=0`). And since the slope measures the difference in two means, it is `-275` fewer grams at birth.

2. Use the intercept and the slope coefficient to calculate the treatment group outcome mean.

To calculate the treatment group outcome mean, you add the intercept (`3413`) to the slope coefficient (`-275`) to get `3138`. That is, the average birth weight for smokers is `3,138` grams.

Illustration of covariance and variance.

Next we will use R as a calculator as we help students understand *how* OLS calculates the slope and intercept coefficients. Remember in a simple regression model:

$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$

We have to two unknown coefficients – the β_0 and β_1. Recall that we call β_0 the intercept which is the mean outcome for the comparison group (i.e., when $X = 0$). And we call ε the error term which is all other causes of the outcome that is not X.

We discussed in class that the slope coefficient formula that OLS uses is a "scaled covariance":

$\widehat{\beta_1} = \frac{Cov(X,Y)}{Var(X)}$

But the intercept also has a formula and it is:

$\widehat{\beta_0} = \bar{Y} - \widehat{\beta_1}\bar{X}$

So our goal here is to *manually* using R as a calculator calculate those two terms. So that you have these formula here with you, the equations for calculating covariance and variance are:

$Cov(X,Y) = E[XY] - E[X]E[Y]$

$Var(X) = E[X^2] - E[X]^2$

We will do this in steps. First, let's calculate the covariance and variance terms. First, we will just rename `bweight` and `mbsmoke` as `Y` and `X` so that you can trace it back to those formula more easily.

```
Y <- data$bweight
X <- data$mbsmoke
```

Next we will calculate the expectations.

```
EX   <- mean(X)            # E[X]
EY   <- mean(Y)            # E[Y]
EXY  <- mean(X * Y)        # E[XY]
EX2  <- mean(X * X)        # E[X^2]
```

Then we will build those covariances and variances.

```
Cov_XY <- EXY - EX * EY
Var_X  <- EX2 - EX^2
```

And finally, we calculate those OLS coefficients. We will now rebuild these quantities using R as a calculator (R does a lot of things) so you can see exactly how `lm()` arrived at those numbers.

```
b1_hat <- Cov_XY / Var_X
b0_hat <- EY - b1_hat * EX
c(b0_hat = b0_hat, b1_hat = b1_hat)
```

```
##    b0_hat    b1_hat
## 3412.9116 -275.2519
```

Now let's compare this with what we did earlier with the `lm()` command.

```
fit1          # OLS results from lm()
```

```
##
## Call:
## lm(formula = bweight ~ mbsmoke, data = data)
##
## Coefficients:
## (Intercept)      mbsmoke
```

```
##      3412.9       -275.3
```

```
b1_hat        # our manual slope (Cov/Var)
```

```
## [1] -275.2519
```

```
b0_hat        # our manual intercept
```

```
## [1] 3412.912
```

Question 2 (10 points)

Now it's your turn. Redo what we did, but use instead the following regressions.

1. Regress `bweight` onto `mage`, calculate the coefficients manually, and compare it with the `lm()` output.

```
fit2 <- lm(bweight ~ mage, data = data)
tidy(fit2) # regression coefficients
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   3074.      40.7      75.4  0
## 2 mage            10.9      1.50      7.22 6.22e-13
```

```
# manual calculation
Y <- data$bweight
X <- data$mage

EX   <- mean(X)            # E[X]
EY   <- mean(Y)            # E[Y]
EXY  <- mean(X * Y)        # E[XY]
EX2  <- mean(X * X)        # E[X^2]

Cov_XY <- EXY - EX * EY
Var_X  <- EX2 - EX^2

b1_hat <- Cov_XY / Var_X
b0_hat <- EY - b1_hat * EX
c(b0_hat = b0_hat, b1_hat = b1_hat)
```

```
##     b0_hat     b1_hat
## 3074.05657   10.85186
```

```
fit2          # OLS results from lm()
```

```
##
## Call:
## lm(formula = bweight ~ mage, data = data)
##
## Coefficients:
## (Intercept)         mage
##     3074.06        10.85
```

```
b1_hat        # our manual slope (Cov/Var)
```

```
## [1] 10.85186
```

```
b0_hat        # our manual intercept
```

```
## [1] 3074.057
```

1. Regress `bweight` onto `fedu`, calculate the coefficients manually, and compare it with the `lm()` output.

```
fit3 <- lm(bweight ~ fedu, data = data)
tidy(fit3) # regression coefficients
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   3129.      29.4     106.   0
## 2 fedu            18.9      2.29      8.25 2.05e-16
```

```
# manual calculation
Y <- data$bweight
X <- data$fedu

EX   <- mean(X)            # E[X]
EY   <- mean(Y)            # E[Y]
EXY  <- mean(X * Y)        # E[XY]
EX2  <- mean(X * X)        # E[X^2]

Cov_XY <- EXY - EX * EY
Var_X  <- EX2 - EX^2

b1_hat <- Cov_XY / Var_X
b0_hat <- EY - b1_hat * EX
c(b0_hat = b0_hat, b1_hat = b1_hat)
```

```
##     b0_hat     b1_hat
## 3129.19338   18.89029
```

```
fit3          # OLS results from lm()
```

```
##
## Call:
## lm(formula = bweight ~ fedu, data = data)
##
## Coefficients:
## (Intercept)         fedu
##     3129.19        18.89
```

```
b1_hat        # our manual slope (Cov/Var)
```

```
## [1] 18.89029
```

```
b0_hat        # our manual intercept
```

```
## [1] 3129.193
```

1. If OLS is just a scaled covariance (scaled by the variance), then what does the sign and size of `Cov(X,Y)` tell us about the relationship between smoking and birthweight?

First, the variance is always positive. So the sign of the OLS coefficient always comes from the numerator, `Cov(X,Y)`. If the OLS coefficient is negative, it is because `Cov(X,Y)<0` which means that as X is above average, Y is *below* average.

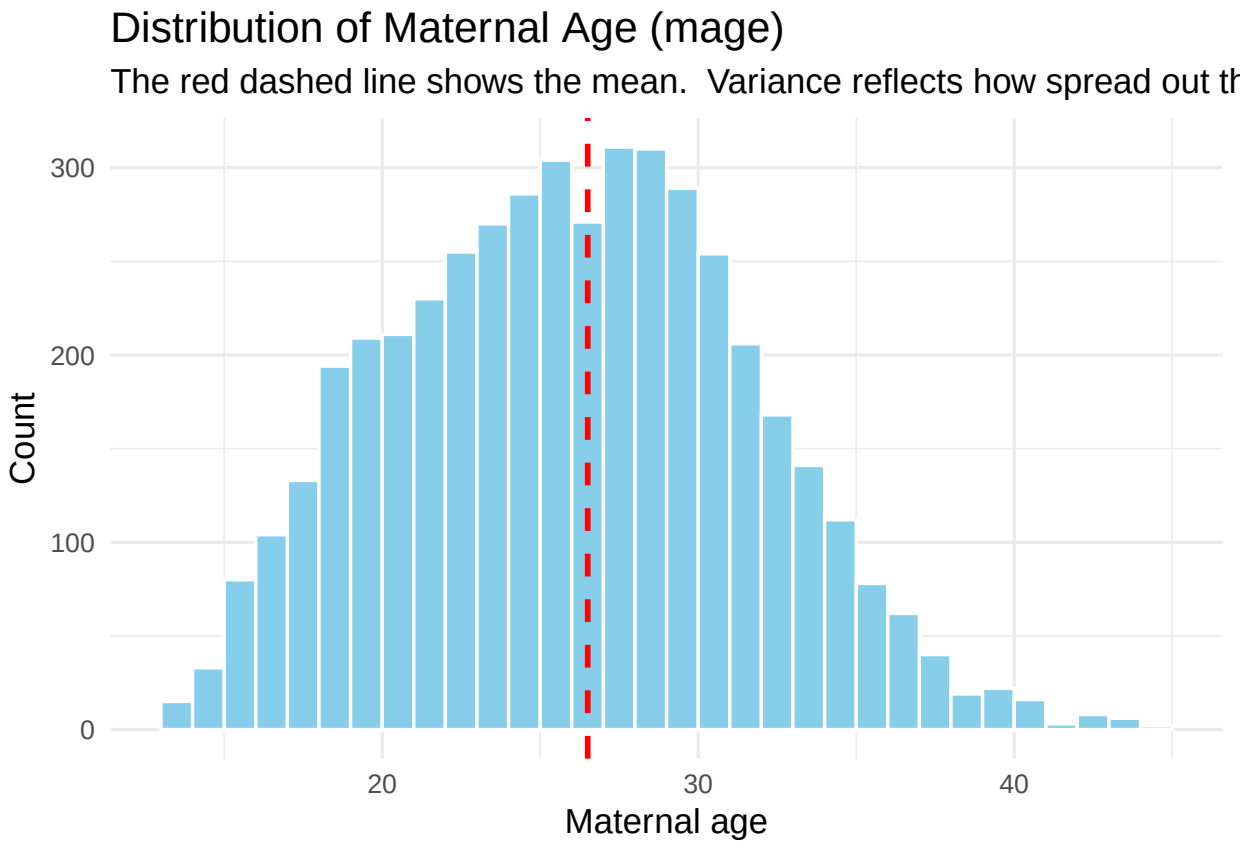
Second, the magnitude of `Cov(X,Y)` reflects how strongly the two variables move together. A larger absolute covariance reflects a steeper relationship, but once we divide by `Var(X)` we scale the strength found in the

covariance relative to how much actual variation there is in smoking behavior, maternal age, father's education or what have you.

Question 3 (8 points): Visualizing Variance and Mean with `ggplot`

Next, let's visualize this. Variance measures how far data points are spread around their mean, and since both variance and mean were in our regression coefficients, this is a good time to introduce them both visually. Let's use `mage` (maternal age) to visualize what this means since you just calculated a regression equation with it as your right-hand-side variable.

```
ggplot(data, aes(x = mage)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "white", boundary = 0) +
  geom_vline(aes(xintercept = mean(mage, na.rm = TRUE)),
             color = "red", linetype = "dashed", linewidth = 1) +
  labs(
    title = "Distribution of Maternal Age (mage)",
    subtitle = "The red dashed line shows the mean.  Variance reflects how spread out the ages are arou
    x = "Maternal age",
    y = "Count"
  ) +
  theme_minimal(base_size = 13)
```



1. What does the height and width of each bar represent?

The width of the bar is simply a part of the number line measuring `mag3`. The height of the bars represents the number of units with those values of `mage`.

1. Where is most of the data concentrated?

Most of the data is is in the center.

1. If the bars were more tightly bunched around the red line, how would that change the variance?

The variance would decline.

1. How might "high" or "low" variance in `mage` affect our regression coefficient estimates?

When there is higher variance in `mage`, it means OLS has more information to see how much `bweight` changes with `mage`. The more variance, the more you can see how `bweight` changes across mothers' age. But when nearly all the mothers have the same age, OLS cannot easily tell if birthweight depends on maternal age or not.

```
mean(data$mage, na.rm = TRUE)
```

```
## [1] 26.50452
```

```
var(data$mage, na.rm = TRUE)
```

```
## [1] 31.57346
```

Question 4 (9 points): Visualizing Variance and Standard Deviation with `ggplot`

Now that students have seen the regression coefficient as a covariance scaled by the variance, we will introduce them to the concept of standard deviation via visualization.

The variance is the *average squared distance from the mean.* And the standard deviation is just its square root, which puts it back in the same units as the mean itself. Our hope is that they will be able to understand conceptually variance, mean and standard deviation better if they can see it visualized, and since we just covered the regression coefficient, our hope is that this is laddering off the previous idea.

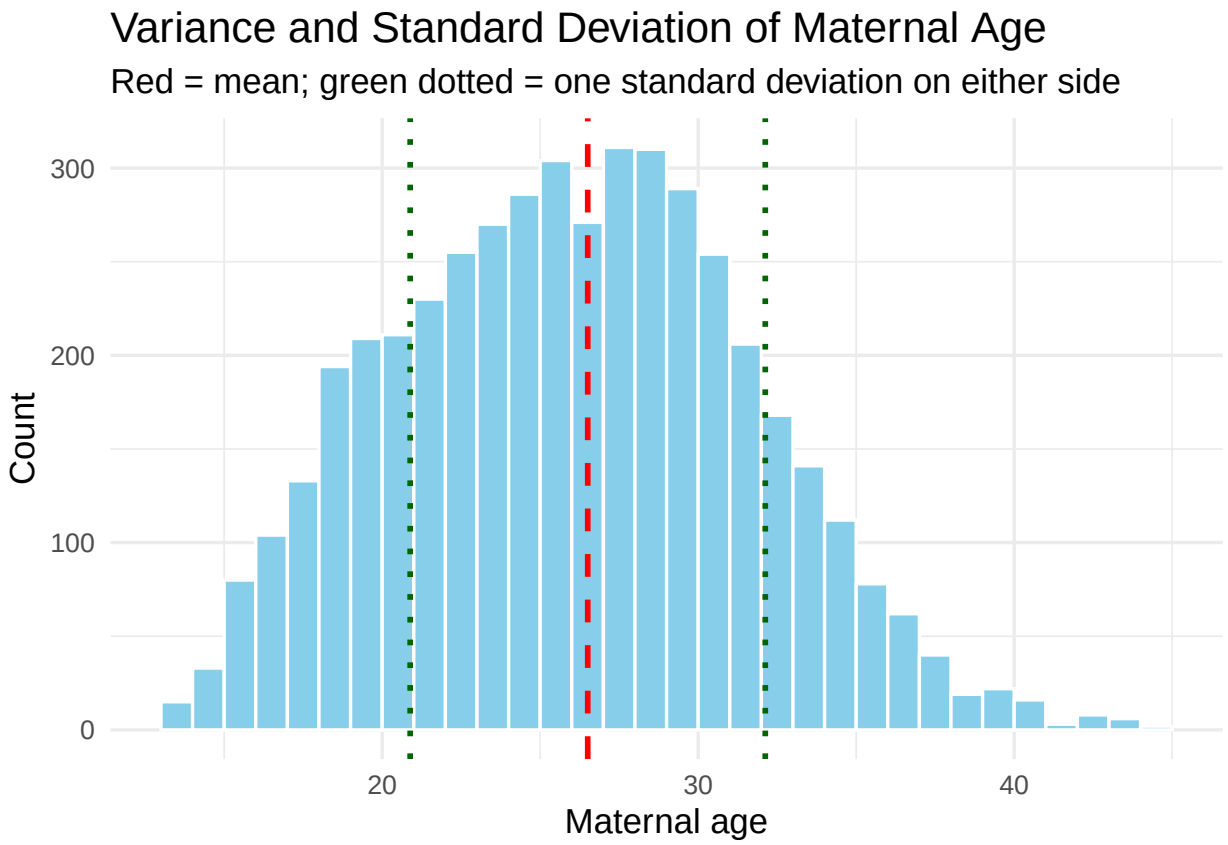
We write down the standard deviation as follows:

$$s_X = \sqrt{Var(X)} = \sqrt{\frac{\sum (X_i - \bar{X})^2}{n-1}}$$

And then we recreate the previous figure, but now we lay on top of the variance graph a vertical line for the mean, and two lines for the standard deviation.

```
sd_mage <- sd(data$mage, na.rm = TRUE)
mean_mage <- mean(data$mage, na.rm = TRUE)

ggplot(data, aes(x = mage)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "white", boundary = 0) +
  geom_vline(aes(xintercept = mean_mage),
             color = "red", linetype = "dashed", linewidth = 1) +
  geom_vline(aes(xintercept = mean_mage + sd_mage),
             color = "darkgreen", linetype = "dotted", linewidth = 1) +
  geom_vline(aes(xintercept = mean_mage - sd_mage),
             color = "darkgreen", linetype = "dotted", linewidth = 1) +
  labs(
    title = "Variance and Standard Deviation of Maternal Age",
    subtitle = "Red = mean; green dotted = one standard deviation on either side",
    x = "Maternal age",
    y = "Count"
  ) +
  theme_minimal(base_size = 13)
```



1. How much of the data is within one standard deviation of the mean?

To calculate how much of the data is within one standard deviation of the mean, we have to first calculate the standard deviation and the mean. We then add and subtract from the mean plus and minus one standard deviation. We then count the number of units within that range. Here is it around `0.68`.

```
# Using mage
X <- data$mage

mean_X <- mean(X, na.rm = TRUE)
sd_X   <- sd(X, na.rm = TRUE)

lower <- mean_X - sd_X
upper <- mean_X + sd_X

# Calculate proportion of observations within 1 SD of mean
within_1sd <- mean(X >= lower & X <= upper, na.rm = TRUE)
within_1sd
```

```
## [1] 0.6887118
```

1. Why is the standard deviation easier to interpret than the variance?

Variance squares the units. So for instance if the variable is `bweight`, which is measured in grams, then its variance is measured in squared grams, and squared grams doesn't have a natural interpretation. But if you take the square root of the variance, then the units are back in the original measurement (i.e., grams), making it a bit easier to interpret. You can see that here:

```r
sd_mage <- sd(data$mage, na.rm = TRUE)
var_mage <- var(data$mage, na.rm = TRUE)
cat("Variance:", var_mage, "\nStandard deviation:", sd_mage)
```

```
## Variance: 31.57346
## Standard deviation: 5.619026
```

Notice that if you square the standard deviation – roughly 5.62 – you get the variance (31.6).

Conclusion

That concludes the basics of regression mechanics. In the next Section, we will review two more things: 1) multivariate regression *calculations* without matrix algebra and 2) *sampling*. We will also start working towards hypothesis testing. But for now, this week's Section has been focused on helping students calculate regression coefficients by hand, using R as a calculator, visualizing mean, variance and standard deviations.