In-class Lab 3

ECON 4223 (Prof. Tyler Ransom, U of Oklahoma)

January 27, 2022

The purpose of this in-class lab is to practice running regressions, computing regression formulas, visualizing the Sample Regression Function, using non-linear transformations, and interpreting coefficients. You may complete this lab as a group, but please turn in separate copies for each group member. To get credit, upload your .R script to the appropriate place on Canvas.

For starters

Open up a new R script (named `ICL3_XYZ.R`, where `XYZ` are your initials) and add the usual "preamble" to the top:

```
library(tidyverse)
library(modelsummary)
library(broom)
library(wooldridge)
```

For this lab, let's use data on school expenditures and math test pass rates from Michigan. This is located in the `meap93` data set in the `wooldridge` package. Each observation is a school district in Michigan.

```
df <- as_tibble(meap93)
```

The Relationship between Expenditures and Math Test Pass Rates

Estimate the following regression model:

$$math10 = \beta_0 + \beta_1 expend + u$$

The code to do so is:

```
est <- lm(math10 ~ expend, data=df)
tidy(est)
glance(est)
```

You should get a coefficient of `0.00246` on `expend`. Interpret this coefficient. (You can type the interpretation as a comment in your .R script.) Is this number small, given the units that `math10` and `expend` are in?

Regression Coefficients "By Hand"

Verify that the regression coefficients in `est` are the same as the formulas from the book:

$$\hat{\beta}_0 = \overline{math10} - \hat{\beta}_1 \overline{expend}, \hat{\beta}_1 = \frac{\widehat{cov}(math10, expend)}{\widehat{var}(expend)}$$

You can do this by typing:

```
beta1 <- cov(df$math10,df$expend)/var(df$expend)
beta0 <- mean(df$math10)-beta1*mean(df$expend)
```

Visualizing Regression Estimates

Often, it's helpful to visualize the estimated regression model. Wooldridge (2015) calls this the "Sample Regression Function." We can do this with the following code:

```
ggplot(df,aes(expend,math10)) +
    geom_point() +
    geom_smooth(method='lm')
```

```
## `geom_smooth()` using formula 'y ~ x'
```

Nonlinear transformations

Let's consider a modified version of our model, where now we use *log* expenditures instead of expenditures. Why might we want to use log expenditures? Likely because we think that each additional dollar spent *doesn't* have an equal effect on pass rates. That is, additional dollars spent likely have diminishing effects on pass rates. (See also: the Law of Diminishing Marginal Returns)

Create the log expenditures variable using `mutate()`:

```
df <- df %>% mutate(logexpend = log(expend))
```

Now estimate your model again and re-do the visualization (showing both functional forms together):

```
est <- lm(math10 ~ logexpend, data=df)
tidy(est)
glance(est)
modelsummary(est)

ggplot(df, aes(expend,math10)) +
  geom_point() +
  stat_smooth(method = "lm", col = "red" ,se=F,formula = y~log(x)) +
  stat_smooth(method = "lm", col = "blue",se=F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

What is the interpretation of β_1 in this new model? (Add it as a comment in your R script)

Standard Errors and Regression Output

Finally, we can look at the standard error, t-statistic, and p-values associated with our regression parameters β_0 and β_1. The `p.value` reported in `tidy(est)` tests the following hypothesis:

$$H_0 : \beta_1 = 0, H_a : \beta_1 \neq 0$$

Does increased school spending significantly increase the math test pass rate?

Computing standard errors by hand

If you have extra time, try computing the standard error formulas by hand, according to the formulas in the text book. To do so, we need to compute the following formulas: `sig` (the standard deviation of u), `n` (our regression's sample size), `SSTx` ($N - 1$ times the variance of `logexpend`), and the sum of the squares of `logexpend`:

```
n <- dim(df)[1]
sig <- sqrt( sum (est$residuals^2) /(n-2) ) # or, more simply, glance(est)$sigma
SSTx <- (n-1)*var(df$logexpend)
sumx2 <- sum(df$logexpend^2)
```

The standard error of the intercept is computed with the following formula:

```
sqrt((sig^2*(1/n)*sumx2)/SSTx)
```

And the standard error of the slope coefficient (`logexpend` in this case) is:

```
sqrt(sig^2/SSTx)
```

References

Wooldridge, Jeffrey M. 2015. *Introductory Econometrics: A Modern Approach*. 6th ed. Cengage Learning.