In-Class Lab 9

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

September 28, 2020

The purpose of this in-class lab is to use R to practice correcting for the presence of heteroskedasticity in regression models. The lab should be completed in your group. To get credit, upload your .R script to the appropriate place on Canvas.

For starters

First, install the `lmtest` and `estimatr` packages.

Open up a new R script (named `ICL9_XYZ.R`, where `XYZ` are your initials) and add the usual "preamble" to the top:

```r
# Add names of group members HERE
library(tidyverse)
library(broom)
library(wooldridge)
library(car)
library(lmtest)
library(estimatr)
library(magrittr)
```

Load the data

We'll use a data set on college GPA, called `gpa3`. The data set contains a sample of 732 students.

```r
df <- as_tibble(gpa3)
```

Check out what's in the data by typing

```r
glimpse(df)
```

The main variables we're interested in are: SAT, high school percentile, credit hours, gender and race. We also only want to look at observations that are in the Spring semester.

Restrict to observations in spring semester

Use a `filter()` statement to drop observations not in the Spring semester. (I won't show you the code; refer to a previous lab if you can't remember how to do it.)

Get rid of variables you won't use

Use a `select()` statement to keep only the variables that will be used:

```
df %<>% select(cumgpa,sat,hsperc,tothrs,female,black,white)
```

Look at the data to make sure the code worked as expected. You should now have 366 observations and 7 variables.

Inference with Heteroskedasticity-Robust Standard Errors

Let's obtain standard errors from the above regression that are robust to heteroskedasticity. To do so, we use the `lm_robust()` function from the `estimatr` package. This function works like regular `lm()` but instead reports a refined version of White's robust standard errors.

```
est.rob <- lm_robust(cumgpa ~ sat + hsperc + tothrs + female + black + white, data=df)
summary(est.rob)
```

1. Compare your new estimates with the original ones (i.e with just using `lm()`). Are any of the default hypothesis test results overturned?

Now look at the robust version of the overall F-test. Is its conclusion changed relative to the default with just `lm()`?

```
glance(est)
linearHypothesis(est.rob, c('sat','hsperc','tothrs','female','black','white'))
```

The LM test

The LM test is an alternative to the overall F test that is reported in `glance(est)`. To perform the LM test, we need to do the following:

- Estimate the restricted model (in the current case, this is an intercept-only model) and then obtain the residuals from that.
- Regress the residuals from (a) on the regressors in the full model.
- the LM statistic is equal to $N * R^2$ from the regression in the second bullet.

2. Conduct an LM test following the steps above (also on p.173 of Wooldridge (2018))

```
# Restricted model
restr <- lm(cumgpa ~ 1, data=df)
LMreg <- lm(resid(restr) ~ sat + hsperc + tothrs + female + black + white, data=df)
LM    <- nobs(LMreg)*glance(LMreg)$r.squared
pval  <- 1-pchisq(LM,6)
```

3. Compare the p-value from the LM test with the p-value for the overall F test (with and without heteroskedasticity-robust correction).

Inference with Cluster-Robust Standard Errors

Now let's obtain standard errors from a different data set and regression model that are robust to heteroskedasticity. We can use `lm_robust()` for this as well.

Load new data

First load the data, which is a CSV file from my website:

```
df.auto <- read_csv('https://tyleransom.github.io/teaching/MetricsLabs/auto.csv')
```

The data set contains specifications for 74 different makes of automobiles. Estimate the following regression model:

$$\log(price) = \beta_0 + \beta_1 weight + \beta_2 foreign + u$$

```
df.auto %<>% mutate(log.price = log(price), foreign = as.factor(foreign))
est.auto <- lm(log.price ~ weight + foreign, data=df.auto)
```

Regular standard errors:

```
summary(est.auto)
```

Now use the heteroskedasticty-robust SEs from last lab:

```
est.rob.auto <- lm_robust(log.price ~ weight + foreign, data=df.auto)
summary(est.rob.auto)
```

Now use the cluster-robust SEs:

```
est.clust.auto <- lm_robust(log.price ~ weight + foreign, data=df.auto,
                            clusters=df.auto$manufacturer)
summary(est.clust.auto)
```

Notice that the SEs on each of the coefficients get bigger with each additional robustness option. The reason for this is that price is correlated within auto manufacturer (due to branding effects).

Finally, you can do an F-test as follows:

```
linearHypothesis(est.clust.auto,c("weight=0","foreignForeign=0"))
```

References

Wooldridge, Jeffrey M. 2018. *Introductory Econometrics: A Modern Approach*. 7th ed. Cengage Learning.