R Applications for Heteroskedasticity

Introductory Remarks

This file builds on the previous R Activities. If you have not yet gone through those files, please make sure to do so before tackling this R Activity.

Wage Equation Set-Up

We will work with a set-up similar to the one in the lecture. Specifically, we aim to look at the relationship between log wages and education and experience. Mathematically,

$$log(wage_i) = \alpha_0 + \alpha_1 educ_i + \alpha_2 exper_i + \epsilon_i$$

To do this we will use the "WAGE2" dataset, available in the "wooldridge" package in R. After installing and loading the package, type

```
data("wage2") # load data
```

As you can see, the data consist of 935 observations and 17 variables, of which we're only interested in *lwage*, *educ*, and *exper*.

OLS with Heteroskedasticity Robust Standard Errors

From what we've learned so far, we already know how to estimate the above equation using OLS, i.e. we type

```
wage2_ols1 <- lm(lwage ~ educ + exper, data=wage2) # run regression
summary(wage2_ols1) # homosk. SEs
```

Suppose we are interested in the effect of education on log wages. We know from previous lectures and R applications how to interpret the coefficient we find (.078): a one year increase in education is associated with a 7.8% increase in wages, holding experience constant. As always we want to be careful not to attribute a causal interpretation to this coefficient. When doing hypothesis testing, we have relied on the reported standard error (0.0066 in the case of education) to calculate, for example, t-values or confidence intervals. The standard errors R reports using the "summary" command are calculated under the assumption of homoskedasticity.

In the lecture, we saw that the assumption of homoskedasiticty is often not satisfied. Indeed, we might want to report heteroskedasiticty robust standard errors. It turns out, that doing this in R is very easy. The regression we run using the "lm" command stays the same. Then, when displaying the results, instead of using the "summary" command, we use the "coeftest" command as follows (make sure you have installed the packages "lmtest" and "sandwich" for this).

```
coeftest(wage2_ols1, vcov = vcovHC(wage2_ols1, type = "HC1")) # heterosk. SEs
```

The first input into the "coeftest" function is the object we have estimated above. The second one is a bit more complicated to when not using matrix notation. Technically, we want to specify the covariance matrix of the estimated coefficients. We do this by using the "vcovHC" function. This function consistently estimates the covariance matrix of the coefficients of the "wage2_ols1" regression. The "type" arguments specified the "type of heteroskedasticity robust standard errors" we want. For the purposes of this course, this last part is not of great importance. While the inputs into the "coeftest" function are a bit more complicated, the above code can just be copied to any other regression you ran to then calculate the heteroskedasticity robust

standard errors. Furthermore, as you can see from the output, the interpretation and layout of the output is the same as what we're already used to.

Notice that as always, the F-test is not provided in this output. Luckily, we can still use the familiar "linearHypothesis" command to report a F-test that is robust to heteroskedasticity. Just for illustration purposes, suppose we add a few additional controls into our wage regression and want to test whether all the controls on the right-hand side are jointly zero (assuming heteroskedasticity). In R, we could type

```
wage2_ols2<- lm(lwage ~ educ + exper + age + married + black +
                  south, data=wage2) # run regression
linearHypothesis(wage2_ols2, c("exper", "age", "married", "black", "south"),
                 c(0, 0, 0, 0, 0), vcov=hccm(wage2_ols2, type="hc1")) # F-test
```

The regression with added control is standard. Then, to conduct the F-test, we use the by now familiar "linearHypothesis" command. Everything is as we're used to, except that the last argument into the function is once again asking us to input the covariance matrix of the coefficients. This time, we get this covariance matrix using the "hccm" function. This is just an alternative function to the "vocHC" function we used above. You can convince yourself of this by typing

```
linearHypothesis(wage2_ols2, c("exper", "age", "married", "black", "south"),
                 c(0, 0, 0, 0, 0), vcov=vcovHC(wage2_ols2, type="HC1")) # F-test
```

As you can see, the output is exactly the same.

Exploring Heteroskedasticity

Given that we now know how to report heteroskedasticity robust standard errors in our regressions, let's focus on whether we might actually want to do so. We will consider three ways of exploring heteroskedasticity in more detail. The first is to plot our data. The second and third are to test for heteroskedasticity using the Breusch-Pagan and White tests, respectively, that you covered in class.

Visual Inspections

To visually inspect whether we should be concerned about heteroskedasticity, we learned in class that we could plot the residuals squared against either the predicted values of the outcome or against any of our regressors of interest. We already know how to do this in R, i.e.

```
wage2$ols1resid <- resid(wage2_ols1) # get residuals
wage2$ols1residsq <- wage2$ols1resid^2 # square the residuals
wage2$fittedlwage <- fitted.values(wage2_ols1) # get fitted values
plot(wage2$fittedlwage, wage2$ols1resid, xlab="Fitted Log Wage", ylab="Residuals")
plot(wage2$educ, wage2$ols1resid, xlab="Education", ylab="Residuals")
```

The first three lines of code calculate the residuals, residuals squared, and fitted values from our original OLS regression (of log wages on education and experience). The last two lines plot the fitted log wages and education, respectively, against the residuals using the "plot" command. You can yourself try to plot the residuals squared against these variables. The interpretation of these plots is analogous to what you learned in the lecture. For example, the plot of the fitted wages against the residuals does not neccesarily suggest that heteroskedasticity is a huge issue.

More important for our purposes here is the fact that there are other ways to create plots in R. One popular one is to use the "ggplot" function. Note that to load this function you need to load the "ggplot2" package. While a full introduction into ggplot is beyond the scope of this R application, note that many great introductory tutorials into ggplot are available online (e.g. just type "ggplot in R" into Google). For our purposes, we will try to reproduce the plot displaying fitted log wages against residuals from above. To start of with, let's type

```
ggplot(wage2, aes(x=fittedlwage, y=ols1resid))
```

We're calling the command "ggplot" and are telling it in the first argument that we want to use the "wage2" data. Next, we tell it that from that dataset, we want the x-axis of the plot to be the fitted log wages and the y-axis to be the residuals of our original regression. You should as always see the output of the graph in the lower right quadrant of RStudio. Notice, however, that the plot that ggplot displays is empty, even though it lists the correct variables on the x- and y-axes.

This is done on purpose. Ggplot can be thought of as a function that creates the plot you want in layers. As such, if we wanted to now add a scatter of our data into this empty plot we could do this as follows

```
ggplot(wage2, aes(x=fittedlwage, y=ols1resid)) + geom_point()
g <- ggplot(wage2, aes(x=fittedlwage, y=ols1resid)) + geom_point()
plot(g)
```

The first line and the second and third lines do the exact same thing, i.e. these are two ways to display the same plot. Indeed, as we can now see the resulting figure is equivalent to the one we found using the "plot" command above. What is the advantage of using the second approach to plotting this figure? Recall that above we've said that ggplot makes your plot in layers. Hence, if you have defined an object "g" which includes this simple scatter, it is now easier to add additional layers to this graph. For example, if we wanted to add a title and name the x- and y-axes, we could type

```
g + ggtitle("Residuals vs. Fitted Wage", subtitle="Exploring Heteroskedasticity") +
  xlab("Fitted Log Wage") + ylab("Residuals")
```

In other words, we don't have to rewrite the whole formula for the full figure, but we can build layer upon layer.

While this application of using ggplot is relatively simple, it is hopefully clear that this idea of building a plot from scratch and layer by layer allows for much more flexibility and, in the end, the creation of nicer and better figures. While for the purposes of this class the "plot" command is usually enough, we highly recommend that you familiarize yourself with the ggplot command. For example, you could try to find out how to change the color of the dots or how to change the scales of the x- and y-axes in the existing plot. If you're feeling more adventurous, you can try to redraw a scatter plot of wages and education (instead of residuals and fitted wages) and find a way to also plot the OLS regression line "across" the scatter plot. Lastly, you could try to plot a histogram of the education variable. The following code provides possible solutions to these three ideas.

```
# color the scatter plot according to different levels of education
p1 <- ggplot(wage2, aes(x=fittedlwage, y=ols1resid)) +
  geom_point(aes(col=educ), size=1.5) +
  ggtitle("Residuals vs. Fitted Wage", subtitle="Exploring Heteroskedasticity") +
  xlab("Fitted Log Wage") + ylab("Residuals")  # add xlim and ylim to change scales
plot(p1)
# scatter of log wages and education including best fit
p2 <- ggplot(wage2, aes(x=educ, y=lwage)) +
  geom_point() +
  geom_smooth(method="lm") +
  ggtitle("Log Wages and Education") +
  xlab("Education") + ylab("Log Wages")
plot(p2)
# bar plot of education
p3 <- ggplot(wage2, aes(x=educ)) +
  geom_bar() +
  ggtitle("Histogram of Education Variable") +
  xlab("Education") + ylab("Frequency")
plot(p3)
```

Breusch-Pagan Test

Let's return to the issue of heteroskedasticity. After having visually explored whether we are concerned about heteroskedasticity in our regressions (which we said was unlikely to be the case), we may want to test this more formally. The Breusch-Pagan test provides one way to do this. From the lecture, we know that the implementation of this test consists of three steps. The first step is to run the regression of interest and to then compute the residuals squared of said regression. Notice that we have already done this above. The residuals squared in our case are called $ols1residsq$. The second step is to regress these squared residuals on all explanatory variables (education and experience in our case) and to compute the usual F-statistic. The third step is to then look at the output of this test and to conclude whether we have a heteroskedasticity issue or not. The null hypothesis is that our assumption of homoskedasticity holds (i.e. that the coefficients on education and experience are jointly zero). In R, implementing step 2 is simple, i.e.

```
BP_step2 <- lm(ols1residsq~educ + exper, data=wage2)
summary(BP_step2)
```

As you can see, the resulting F-statistic is 2.042 with a p-value of 0.1303, implying that we fail to reject the null hypothesis of homoskedasticity. Hence, the Breusch-Pagan test confirms what we suspected when looking at the data visually: we do not seem to have a heteroskedasticity problem.

White Test

In class, you were told about two cases of the White test. The first and general case regresses the residuals squared on education, experience, as well as the square of each term plus their interaction. The second more special case of the White test regresses the residuals squared on the fitted values of log wages as well as the square of these fitted values. Otherwise, the test procedure is analogous to the Breusch-Pagan test discussed above. In R, this is all very straightforward, i.e.

```
# define squares and interactions for white test
wage2$educsq<- wage2$educ^2 # education squared
wage2$expersq <- wage2$exper^2 # experience squared
wage2$educexper <- wage2$educ*wage2$exper # education * experience

# define fittes values and squares for the special white test
wage2$olsfitted <- fitted.values(wage2_ols1)
wage2$olsfittedsq <- wage2$olsfitted^2

# Usual White test
White_1 <- lm(ols1residsq~educ+exper+educsq + expersq+educexper, data=wage2)
summary(White_1)

# Special White test
White_2 <- lm(ols1residsq~olsfitted + olsfittedsq, data=wage2)
summary(White_2)
```

The resulting F-statistics are 0.8436 and .02164 with p-values of 0.5189 and 0.8055, respectively for the usual and special White test. Clearly, we again fail to reject the null hypothesis of homoskedasticity. Thus, the White test confirms that heteroskedasticity does not seem to pose an issue in our data.