

Topic 2 Simulation: Heteroskedasticity

We introduced heteroskedasticity in this weeks lecture so hopefully now you have a good idea of what it is and what impact it can have in regressions. These notes look to take you through a simulation of heteroskedasticity so that you can see in a control environment the impact it has on results. Hopefully this will help clarify the concept and will also allow us to verify that the tests we previously outlined pick up heteroskedasticity. With heteroskedasticity the variance of the errors has some form of functional relationship with observable characteristics. Heteroskedasticity does not cause bias (OLS results will still be unbiased) but it does cause inefficiency where OLS may not be the best model.

We will start by creating a dataset with 1000 observations, and two independent variables, x and x2. Hopefully you are familiar with all the STATA commands you see here as we used them last week.

```
clear
set seed 101
set obs 1000
gen x = rnormal()
gen x2= rnormal()*5
```

The next step we need to follow is to define an error term. If we are trying to create heteroskedasticity we want the variance of this error term to be correlated with one (or more) of our independent variables.

Below I define u to be a random draw from a normal distribution, with mean 0 and a standard deviation of $(x)^2$. As the standard deviation of the variable is a function of x then the variance (the standard deviation is the square root of the variance) must also now be related to x. I square the value of x as some of the observations of x are negative, and STATA will not allow me to have a negative standard deviation. The square of x will be positive and thus this is no longer an issue. We need to create our y variable so it is a function of this error which I have done below.

```
gen u = rnormal(0, (x^2))
gen y = 2 + 4*x + 8*x2 + u
reg y x x2
```

The results can be seen below in Figure 2. Our understanding suggests that heteroskedasticity should not bias the coefficients, but is likely to change the standard errors. As we can see below the model does quite well at finding the point estimates of x, x2 and the constant term. The problem is that we might not be able to trust our standard errors and thus we cannot trust our p-values or our significance testing.

Figure 1: OLS Output for $y=2+4x + 8x^2$ with heteroskedasticity

Source	SS	df	MS	Number of obs	=	1,000
Model	1536098.11	2	768049.056	F(2, 997)	>	99999.00
Residual	2579.30747	997	2.58706867	Prob > F	=	0.0000
				R-squared	=	0.9983
				Adj R-squared	=	0.9983
Total	1538677.42	999	1540.21764	Root MSE	=	1.6084

y	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
x	3.93367	.0525698	74.83	0.000	3.83051	4.036831
x2	7.988146	.0104713	762.86	0.000	7.967598	8.008695
_cons	2.004463	.0509179	39.37	0.000	1.904545	2.104381

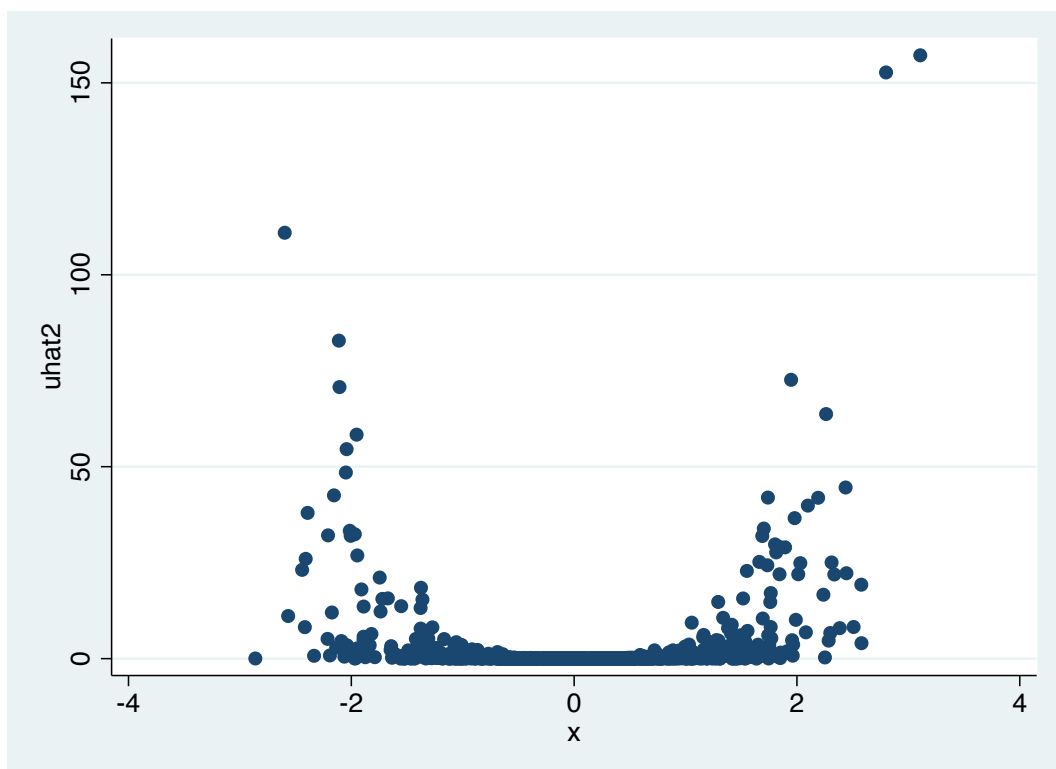
Are we sure there is heteroskedasticity? We should be able to run a quick test and find out. Below I predict the error term from the regression and square it to give the variance (uhat2). Plotting this against the independent variable that we believe it is related to gives us Figure 3, showing a clear non-linear relationship with x. Thus we know we did a good job creating heteroskedasticity. If you run the same scatter plot for uhat2 and x then you will not see such a clear pattern.

```

predict uhat, res
gen uhat2=uhat^2
scatter uhat2 x

```

Figure 2: Plot of uhat2 against x



So what is the final step of today? The final thing we should do, is try to see how much our robust standard errors correct for this heteroskedasticity. As we mentioned in lectures, running robust errors should allow us to correct for all but the most extreme cases of heteroskedasticity. Looking at the results in Figure 3 we can see that when running the regression with robust standard errors the errors are over twice as large for the independent variable x. While this has not made any difference to our p-value in this case because the coefficient on x is quite large (4 is quite far away from the null hypothesis of zero), but in other examples this could mean the difference between rejecting the null and finding a coefficient to be significant, and failing to reject the null and determining that it is not significant.

reg y x x2, robust

Figure 3: Regression Output for $y=2+4x + 8x^2$ with robust standard errors

Linear regression		Number of obs	=	1,000
		F(2, 997)	>	99999.00
		Prob > F	=	0.0000
		R-squared	=	0.9983
		Root MSE	=	1.6084

y	Coef.	Robust Std. Err.	t	P> t	[95% Conf. Interval]	
x	3.93367	.1129533	34.83	0.000	3.712017	4.155324
x2	7.988146	.0111912	713.79	0.000	7.966185	8.010108
_cons	2.004463	.0503285	39.83	0.000	1.905701	2.103225

Take away points, heteroskedasticity may not be that exciting, but it is very important when trying to correctly estimate the relationship between variables, and thus it is always worth checking for and trying to control for. In my own work I never run a regression which does not have the robust option on the end.....

That's it from me for today, next week I plan to use this section of the module to further demonstrate the issues with the LPM when the dependent variable is dichotomous. See you then!

David Morris