

What is heteroskedasticity and why is it a problem?

All econometrics textbooks have a good discussion of heteroskedasticity. These are the most important facts:

- Heteroskedasticity occurs when the variance of the disturbance is not constant.
- Heteroskedasticity is a problem often encountered in cross section data.
- Heteroskedasticity does not affect the parameter estimates: the coefficients should be unbiased.
- Heteroskedasticity does, however, bias the variance of the estimated parameters.
- As a result, the t-values for your estimated coefficients cannot be trusted.

There are a number of procedures which test for the presence of heteroskedasticity. The simplest (though least reliable) is simply to plot each of your independent variables against the square of the residual. If you can see a pronounced pattern, in which the squared residuals get larger or smaller as a particular independent variable gets larger or smaller, then you probably have a problem with heteroskedasticity. In R, the package *car* will do this kind of residual plot using the function *spreadLevelPlot*.

```
library(car)
zz<-lm(y~xx)
spreadLevelPlot(zz)
```

How do we test for heteroskedasticity?

The most-often used test for heteroskedasticity is called the **Breusch-Pagan test**, or the Lagrange Multiplier test for heteroskedasticity. Please look at the description of this test in your textbook.

The idea behind the Breusch-Pagan and other tests for heteroskedasticity is very simple: Is there some variation in the *squared* residuals which can be explained by variation in the independent variables? This suggests using some measure of explained variation similar to R-squared, or some test statistic similar to an F-test.

In R, the package *car* performs a version of the Breusch-Pagan test with the function *ncvTest*.

```
library(car)
zz<-lm(y~xx)
ncvTest(zz)
```

The null hypothesis is that the residuals are homoskedastic. The BP statistic is distributed Chi-Square, and a high value of the Chi-Square statistic (or a low p-value) allows you to reject the null hypothesis.

If heteroskedasticity exists, what is done about it?

Again, the problem with heteroskedasticity is that we cannot trust our t-statistics, because our estimates of the standard errors are biased. Various solutions are suggested for the problem. My favorite is very simple: use White's robust variance-covariance matrix to generate the standard errors for our t-statistics. A nice feature of White's correction is that the values will be correct whether or not you have heteroskedasticity; so one could even skip the tests above.

In R, the package *car* contains the function *hccm* that will calculate the robust variance-covariance matrix. The package *sandwich* contains the function *coefTest* that allows one to produce a table of the coefficients, t-stats, and p-values, based upon a corrected variance-covariance matrix. Both of these packages are loaded with package *AER*, so simply load this.

```
library(AER)
zz<-lm(y~xx)
hccm(zz)

coefTest(zz, vcov = hccm(zz))
```

More complex hypothesis testing can be done like this:

```
linearHypothesis(zz, "x1+x2==1"
, white.adjust=TRUE)
```

or

```
linearHypothesis(zz, "x1+x2==1"
, vcov=hccm(zz))
```

Homework (Short)

Redo the Cobb-Douglas production function for 1992, from homework two, this time taking heteroskedasticity into account. Test for heteroskedasticity. Then perform a test for returns to scale, with the corrected standard errors. Be sure to present your results in the appropriate format.

```

Monte Carlo for heteroskedasticity: S:\TEFF\662\R\r07mc.R
#--Monte Carlo: Heteroskedasticity--
rm(list=ls(all=TRUE))
#--Set path to your own directory--
setwd("S:/teff/662/R/")
options(echo=TRUE)
library(foreign)
library(AER)

#---We make up our data, 500 times---
#---Though our data are randomly generated, we KNOW our coefficients--
#---We estimate the coefficients and compare the estimates with true values--

estcoef<-NULL
#--empty object to store our results from the 500 simulations--
for (i in 1:500){
x<-matrix(rnorm(50),50,1)      #make 1 independent variable
err<-rnorm(50)                #homoskedastic error term
err<-(err-mean(err))/sd(err)  #standardizing err
herr<-x*err                   #heteroskedastic error term
herr<-(herr-mean(herr))/sd(herr) #standardizing herr
y1<-x*7+err
y2<-x*7+herr
z1<-summary(lm(y1~x))
cf1<-z1$coefficients[2,1]
se1<-z1$coefficients[2,2]
z2<-summary(lm(y2~x))
cf2<-z2$coefficients[2,1]
se2<-z2$coefficients[2,2]
estcoef<-rbind(estcoef,cbind(cf1,se1,cf2,se2))
}
estcoef<-data.frame(estcoef)
names(estcoef)<-c("homBeta","homSE","hetBeta","hetSE")

#--take mean and standard deviation of results--
#-True standard error is the standard deviation of the coefficient estimates--
#-Compare true standard error with the mean of the estimated standard errors--

apply(estcoef,2,mean)
apply(estcoef,2,sd)

#-----
#--plot histograms of results--
#-----

#--check if the coefficient is biased---
layout(matrix(1:2,1,2))
hist(estcoef$homBeta,breaks=30)
lines(density(estcoef$homBeta),col="blue")
abline(v=7,col="red",lty=2,lwd=2)
hist(estcoef$hetBeta,breaks=30)
lines(density(estcoef$hetBeta),col="blue")
abline(v=7,col="red",lty=2,lwd=2)
layout(1)

#--check if the standard error is biased--
layout(matrix(1:2,1,2))
hist(estcoef$homSE,breaks=30)
lines(density(estcoef$homBeta),col="blue")
abline(v=sd(estcoef$homBeta),col="red",lty=2,lwd=2)
hist(estcoef$hetSE,breaks=30)
lines(density(estcoef$hetBeta),col="blue")
abline(v=sd(estcoef$hetBeta),col="red",lty=2,lwd=2)
layout(1)

```