ECON-320-Lab-9

Sonan Memon

University of Oregon

Packages

```r
library(tinytex)

library(tidyverse)

library(dslabs)
library(dplyr)

library(ggplot2)

library(estimatr)

library(tibble)
library(broom)
library(modelsummary)
```

Introduction

- Heteroskedasticity and autocorrelation are two common violations of the standard assumptions of OLS model.
- Goldfeld-Quant Test is a test for heteroskedasticity: It assumes that $\sigma_{u,i}$, the standard deviation of the probability distribution of the disturbance term in observation i is proportional to size of X_i.
- White Test is the standard test for heteroskedasticity.
- Accounting for Clustering in Standard Errors.

Simulated Hetroskedastic Model

```r
n = 100
set.seed(1)

data = tibble(
i = c(1:n),
e1 = rnorm(n, 0, 1),
e2 = rnorm(n, 0, 3),
x = runif(n, 0, 10),
u = ifelse(x <= 5, e1, e2),
y = 1 + 3*x + u
)
```

Conducting Goldfeld-Quant Test

- The steps for the the test are:

1. Order your the observations by x
2. Split the data into two groups: first n' and last n'; the middle n' are excluded.
3. Run separate regressions of y on x for first n' and third n' sets.
4. Record RSS_1 and RSS_2 for the two subsets and calculate the test-statistic.
5. Compare the statistic with critical value from $F_{n'-k,n'-k}$ distribution.

Simulated Hetroskedastic Model

```
ranked_data = data %>% arrange(x)

n1 = ranked_data[1:33,]
n2 = ranked_data[67:100,]

lm1 = lm(data = n1, y~x)
lm2 = lm(data = n2, y~x)


 models <- modelsummary(
    list("Model 1" = lm1, "Model 2" = lm2),
    stars = TRUE,
    statistic = "std.error",
    output = "tinytable",
    title = "Results From Simulated Data",
    gof_omit = ".*"  # remove AIC, BIC, etc.
  )
```

Simulated Hetroskedastic Model

Table 1: Results From Simulated Data

	Model 1	Model 2
(Intercept)	0.940**	5.136
	(0.279)	(4.112)
x	3.114***	2.542***
	(0.149)	(0.478)

+ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Simulated Hetroskedastic Model

- We reject the null hypothesis below.

```
[1] 6.39848
```

White Test

- The White test does not rely on the assumption of a specific functional form of heteroskedasticity, making it a general test.
- $y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i$ and obtain the residuals $\hat{\varepsilon}_i$.
- Regress the squared residuals $\hat{\varepsilon}_i^2$ on the original regressors, their squares, and their cross-products:
  $\hat{\varepsilon}_i^2 = \alpha_0 + \alpha_1 x_{1i} + \alpha_2 x_{2i} + \alpha_3 x_{1i}^2 + \alpha_4 x_{2i}^2 + \alpha_5 x_{1i} x_{2i} + u_i$
- Compute the test statistic: $LM = n \cdot R^2$ where R^2 is from the auxiliary regression, and n is the sample size.

White Test

- Under the null hypothesis of homoskedasticity, the test statistic follows a chi-square distribution: $LM \sim \chi_q^2$ where q is the number of regressors (excluding the intercept) in the auxiliary regression.
- H_0: The error terms are homoskedastic; H_1: The error terms are heteroskedastic.
- Reject H_0 if the test statistic exceeds the critical value from the chi-square distribution at the chosen significance level.

Simulated Hetroskedastic Model

- We reject the null hypothesis below using white test as well; 5.991 is the critical value of the chi-squared distribution with 2 degrees of freedom and $\alpha = 0.05$.

```
lm = lm(data = data, y ~ x)

res = unname(resid(lm))

white_df = cbind(data, res)

white_df = white_df %>% mutate(res_sq = res^2)

lm_res = lm(data = white_df, res_sq ~ x + I(x^2))

rsq_w = summary(lm_res)$adj.r.squared

LM_w = rsq_w*n
LM_w - 5.991
```

```
[1] 3.653362
```

Clustering

- Clustering is a concern when standard errors are similar within sufficiently large groups or clusters but vary systematically across groups.
- For instance, if income inequality is higher in some US states than others or house price uncertainty is systematically different across states or income classes.

Clustering

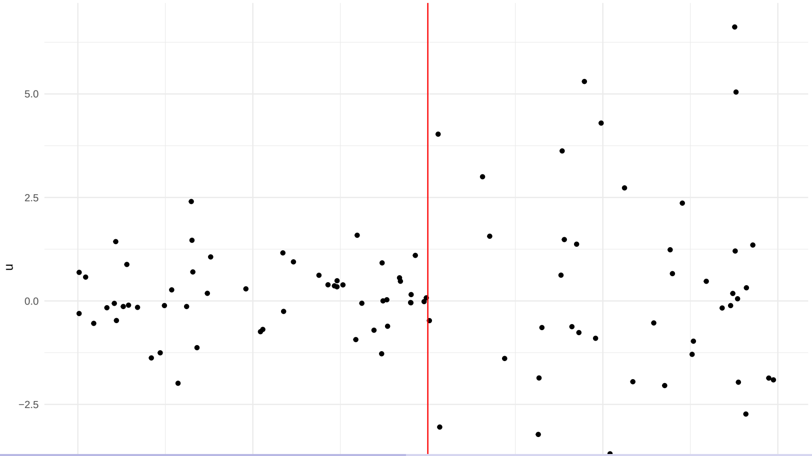
- In the following simulated data, the error terms vary systematically across states, indicating clustering.

```
n = 100
set.seed(1)
data2 = tibble(i = c(1:n),
e1 = rnorm(n, 0, 1),
e2 = rnorm(n, 2, 1),
v = rnorm(n, 0, 3),
state = ifelse(i <= 50, 'Oregon', 'California'),
u = ifelse(state == 'Oregon', e1, e2),
x = runif(n, 1, 10),
y = 1 + 2*x + u + v
)
```

Clustering

- As is evident below, the error term u varies systematically and is higher for California: $x > 50$ relative to Oregon $x <= 50$

```
ggplot(data = data, aes(x = x, y = u))+
geom_point() + theme_minimal() + geom_vline(xintercept = 5, colo
```



Cluster Robust Standard Errors

- In the regression below, we resolve the clustering by state via running a regression which produces cluster robust standard errors.
- By default, you should have cluster robust standard error in many applied settings for sharper inference.

```
lm = lm_robust(data = data2, y ~ x,
clusters = state)
```