Variance Issues

Dr. Michael Fix
mfix@gsu.edu

Georgia State University

13 March 2025

What is Heteroskedasticity?

- One of the Gauss-Markov assumptions requires that we have homoskedasticity, or consistant variance in the error term
- Heteroskedasticity is when there is unequal error variance over $\mathbf{X}$
- Heteroskedasticity is common in observational social science data

Causes of Heteroskedasticity

- Two common causes of heteroskedasticity common in observational social science data are:
  1. Aggregation across subunits of differing size
  2. Pooled data across units

Problems Caused by Heteroskedasticity

- When heteroskedasticity is present, OLS estimates are still unbiased
- However, standard erros are no longer unbiased estimates
- Thus, OLS is no longer BLUE as other linear models may be more efficient
- Further, if our SEs are biased, our t-statistic, p-values, confidence intervals, etc will all be unreliable

Testing for Heteroskedasticity

- As heteroskedasticity is very common in observation social science data, it is important to test for it even if we have no theoretical reason to believe it likely (although we usually do)
- There are several tests for detecting heteroskedasticity,
- Two of the most common are to visually examine a plot of residuals vs fitted values and the Breusch Pagan Test

Toy Model

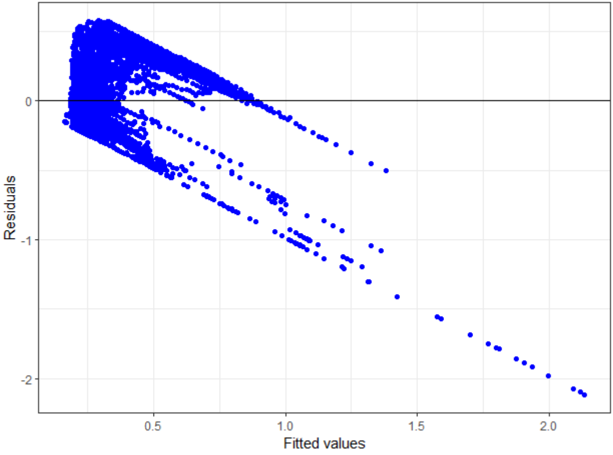
- Check Ozlem's R script for details
- Sample: all countries, 1900-2021

Predictors of democracy in the world

	Dependent variable:
	Electoral Democracy Index
GDP per capita	0.018 (0.0002) t = 72.287 p = 0.000
Urbanization	-0.027 (0.009) t = -3.032 p = 0.003
Constant	0.186 (0.003) t = 65.317 p = 0.000
Observations	15,125
R2	0.279
Adjusted R2	0.279
Residual Std. Error	0.214 (df = 15122)
F Statistic	2,931.659* (df = 2; 15122)
Notes	Standard errors are in parentheses.

Residual vs Fitted Plot

```
# Looking for heteroskedasticity - plot residuals ~ fitted.values
my_model |>
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point(col = 'blue') +
  geom_abline(slope = 0) +
  labs(x = "Fitted values", y = "Residuals") +
  theme_bw()
```

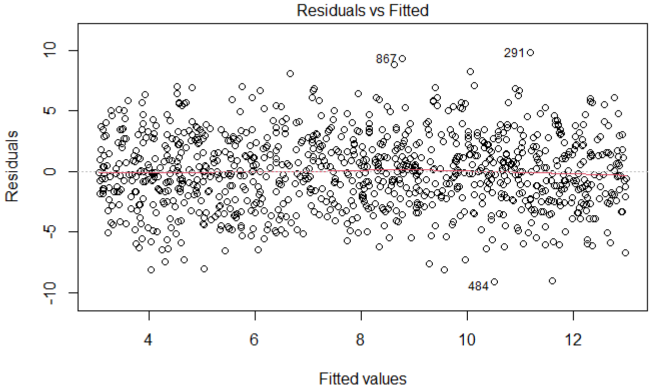


How does homoskedasticity look like?

```r
# How homoskedasticity looks like ----
n <- 1000       # sample size
x <- runif(n,
           min = 0,
           max = 100)

y.good <- 3 + 0.1 * x + rnorm(n, sd = 3)

# Residual vs fitted for ideal OLS setting
lm.good <- lm(y.good ~ x)
plot(lm.good, which = 1)
```



Breusch Pagan Test

```
# Breusch-Pagan test ----
# From lmtest() package

# H0: Homoscedasticity is present (the residuals are # distributed with equal variance)
# HA: Heteroscedasticity is present (the residuals are not distributed with equal variance)
bptest(my_model)

# p is smaller than 0.05 - so we reject the null, find support for heteroscedasticity
```

```
> bptest(my_model)

	studentized Breusch-Pagan test

data:  my_model
BP = 5389.2, df = 2, p-value < 2.2e-16
```

- R output is not intuitive
- However, useful when sample size small
- Always use both residual vs. fitted values plot and Breusch Pagan test together

Solutions for Modeling Heteroskedastic Data

- We will discuss three solutions for dealing with heteroskedastic data
  1. Weighted Least Squares (WLS)
  2. "Robust" Standard Erros
  3. Clustering

What is WLS?

- Where the OLS estimator assumes consistent error variance, weighted least square offers a relaxation of that assumption
- It does this by weighting each observation in a way that is inversely proportional to the error variance
- This requires that we know these weights!

WLS in Practice

Let's start with a linear regression with a relaxed variance assumption:

$$Y_i = \mathbf{X}_i\boldsymbol{\beta} + u_i$$

with:

$$\mathrm{Var}(u_i) = \sigma^2/w_i$$

where w_i is <u>known</u>.

WLS in Practice

WLS now minimizes:

$$\text{RSS} = \sum_{i=1}^{N} w_i (Y_i - \mathbf{X}_i \beta).$$

which gives:

$$\begin{aligned} \hat{\beta}_{WLS} &= [\mathbf{X}'(\sigma^2 \mathbf{\Omega})^{-1}\mathbf{X}]^{-1}\mathbf{X}'(\sigma^2 \mathbf{\Omega})^{-1}\mathbf{Y} \\ &= [\mathbf{X}'\mathbf{W}^{-1}\mathbf{X}]^{-1}\mathbf{X}'\mathbf{W}^{-1}\mathbf{Y} \end{aligned}$$

WLS in Practice

where:

$$\mathbf{W} = \begin{bmatrix} \frac{\sigma^2}{w_1} & 0 & \cdots & 0 \\ 0 & \frac{\sigma^2}{w_2} & \cdots & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \frac{\sigma^2}{w_N} \end{bmatrix}$$

With the variance-covariance matrix:

$$\begin{aligned} \text{Var}(\hat{\beta}_{WLS}) &= \sigma^2(\mathbf{X}'\boldsymbol{\Omega}^{-1}\mathbf{X})^{-1} \\ &\equiv (\mathbf{X}'\mathbf{W}^{-1}\mathbf{X})^{-1} \end{aligned}$$

WLS in Practice

A common case is:

$$\text{Var}(u_i) = \sigma^2 \frac{1}{N_i}$$

where N_i is the number of observations upon which (aggregate) observation i is based.

Estimating WLS in R

- The atheoretical approach is to estimate error variance by regressing the squared (or absolute value of) residuals of our base model on predicted values, then using the inverse of the predictions from this model as weights.

```r
{r}
# Estimate base model
base_model <- lm(democracy ~ gdp_per_capita + urbanization,
                 data = filtered_data)
# Generate weights vector
weights_vec <- 1 / lm(abs(base_model$residuals) ~ base_model$fitted.values)$fitted
.values^2
# Estimate WLS
wls_model <- lm(democracy ~ gdp_per_capita + urbanization,
                data = filtered_data,
                weights = weights_vec)
# Summarize
summary(wls_model)
```

```
Call:
lm(formula = democracy ~ gdp_per_capita + urbanization, data = filtered_data,
    weights = weights_vec)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-2.7077 -0.8674  0.1721  0.8647  2.0852

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.383553   0.033615  11.410  < 2e-16 ***
gdp_per_capita  0.011328   0.001832   6.185 4.99e-09 ***
urbanization   -0.065995   0.089888  -0.734    0.464
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.129 on 160 degrees of freedom
Multiple R-squared:  0.1964,    Adjusted R-squared:  0.1863
F-statistic: 19.55 on 2 and 160 DF,  p-value: 2.535e-08
```

WLS vs Robust SEs

- WLS is ideal when you have heteroskedasticity present
- However, it requires us to have a lot of knowledge about our error variances and we often lack this knowledge
- Robust standard errors offer an attractive alternative as they offer consistant standard error estimates in the presence of heteroskedasticity when we have no knowledge about the form of the heteroskedasticity

WLS vs Robust SEs

However, nothing comes without a cost.

- Robust SEs are consistant, meaning t-statistic estimates (and F tests) are only *asymptotically* valid. They are potentially biased in small samples
- They are less efficient than OLS estimates if errors are actually homoskedasticity (i.e. when $\text{Var}(u) = \sigma^2\mathbf{I}$)

Nonetheless, Robust SEs are "better" than OLS estimates anytime heteroskedasticity is present, just be careful with small sample sizes as their accuracy improves as N increases

Estimating "Robust" SEs

The formula for the variance-covariance of the parameters under heteroskedasticity:

$$\begin{aligned}\text{Var}(\beta_{\text{Het.}}) &= (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathbf{W}^{-1}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1} \\ &= (\mathbf{X}'\mathbf{X})^{-1}\,\mathbf{Q}\,(\mathbf{X}'\mathbf{X})^{-1}\end{aligned}$$

where $\mathbf{Q} = (\mathbf{X}'\mathbf{W}^{-1}\mathbf{X})$ and $\mathbf{W} = \sigma^2\mathbf{\Omega}$.

We can rewrite $\mathbf{Q}$ as

$$\begin{aligned}\mathbf{Q} &= \sigma^2(\mathbf{X}'\mathbf{\Omega}^{-1}\mathbf{X}) \\ &= \sum_{i=1}^{N}\sigma_i^2\mathbf{X}_i\mathbf{X}_i'\end{aligned}$$

Estimating this would require us to know $\mathbf{\Omega}$ (and $\mathbf{W}$).

Estimating "Robust" SEs

Huber and White's solution was to estimate $\hat{\mathbf{Q}}$ as:

$$\widehat{\mathbf{Q}} = \sum_{i=1}^{N} \hat{u}_i^2 \mathbf{X}_i \mathbf{X}_i'$$

Yields:

$$\begin{aligned}
\widehat{\text{Var}(\beta)}_{\text{Robust}} &= (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\widehat{\mathbf{Q}}^{-1}\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1} \\
&= (\mathbf{X}'\mathbf{X})^{-1}\left[\mathbf{X}'\left(\sum_{i=1}^{N} \hat{u}_i^2 \mathbf{X}_i \mathbf{X}_i'\right)^{-1}\mathbf{X}\right](\mathbf{X}'\mathbf{X})^{-1}
\end{aligned}$$

Estimating Robust SEs in R

```
# Robust standard errors ----

# We are going to use sandwich package and vcov related functions
# I recommed reading the package carefully for your own projects
# https://cran.r-project.org/web/packages/sandwich/sandwich.pdf

# This is an econometric package that computes robust standard errors in a
# regression model. These robust estimates are also called sandwich estimators
# because the formula looks like a sandwich. But, you only know that if you
# studied a bit of econometric theory.

# sandwich package has 7 different ways to estimate standard errors
# vcovBS, vcovCL, vcovHAC, vcovHC, vcovOPG, vcovPC, vcovPL
# We use vcovHC (heteroscedasticity- consistent) for Huber-White correction

# Get robust standard errors
coeftest(my_model, vcov. = vcovHC(my_model, type = "HC0"))

# type can be:
# type = c("HC3", "const", "HC", "HC0", "HC1", "HC2", "HC4", "HC4m", "HC5")

# This gives us robust standard errors but in order to get this output
# We need to generate these robust se's differently
```

```
> coeftest(my_model, vcov. = vcovHC(my_model, type = "HC0"))

t test of coefficients:

                Estimate Std. Error t value  Pr(>|t|)
(Intercept)   0.26372283 0.00866490 30.4357 < 2.2e-16 ***
gdp_per_capita 0.01348383 0.00052046 25.9077 < 2.2e-16 ***
urbanization  0.25342419 0.06114728  4.1445 5.045e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Estimating Robust SEs in R

```r
# This gives us robust standard errors but in order to get this output
# We need to generate these robust se's differently

# Generate robust standard errors to use them in stargazer
cov_m1 <- vcovHC(my_model, method = "HC3")
rob_m1 <- sqrt(diag(cov_m1))

# Use these robust standard errors in stargazer function
stargazer(my_model,
          se = (list(rob_m1)),
          type = "text",
          title = "Predictors of democracy in the US",
          covariate.labels = c("GDP per capita", "Urbanization"),
          dep.var.labels = c("Electoral Democracy Index"),
          report = "vcstp",
          ci.level = 0.95,
          star.cutoffs = c(0.05),
          notes.align = "l",
          notes.append = FALSE,
          notes.label = "Notes",
          notes = "Standard errors are in parentheses.")
```

Clustering SEs

- So far we have seen an approach for dealing with heteroskedasticity when we have *a lot* of information about the nature of the heteroskedasticity (WLS)
- ...and one for when we have *no* information about the nature of the heteroskedasticity (robust SEs)
- In practice we are often somewhere in the middle

Nested Data

- Often we have data were observations are nested into groups
  - E.g. individuals within countries/states
- If we assume that the error variance *within* each group is the same but the error variance *between* each group is different, then we can account for this with a modified version of Huber-White Robust SEs

Clustering SEs

A common case:

$$Y_{ij} = \mathbf{X}_{ij}\boldsymbol{\beta} + u_{ij}$$

with

$$\sigma_{ij}^2 = \sigma_{ik}^2.$$

"Robust, clustered" estimator:

$$\widehat{\text{Var}(\beta)}_{\text{Clustered}} = (\mathbf{X}'\mathbf{X})^{-1}\left\{\mathbf{X}'\left[\sum_{i=1}^{N}\left(\sum_{j=1}^{n_j}\hat{u}_{ij}^2\mathbf{X}_{ij}\mathbf{X}_{ij}'\right)\right]^{-1}\mathbf{X}\right\}(\mathbf{X}'\mathbf{X})^{-1}$$

Estimating Clustered SEs in R

```r
# Clustered standard errors ----

# Get clustered standard errors
coeftest(my_model, vcov. = vcovCL(my_model, cluster = ~ country_name))

# Generate clustered standard errors to use them in stargazer
cov_m2 <- vcovCL(my_model, cluster = ~ country_name)
rob_m2 <- sqrt(diag(cov_m2))

# Use these robust standard errors in stargazer function
stargazer(my_model,
          se = (list(rob_m2)),
          type = "text",
          title = "Predictors of democracy in the world",
          covariate.labels = c("GDP per capita", "Urbanization"),
          dep.var.labels = c("Electoral Democracy Index"),
          report = "vcstp",
          ci.level = 0.95,
          star.cutoffs = c(0.05),
          notes.align = "l",
          notes.append = FALSE,
          notes.label = "Notes",
          notes = "Standard errors are in parentheses.")
```