Lecture 6.1: OLS and the “Sandwich” Standard Error

A Robust Haiku

T-stat looks too good.

Use robust standard errors.

Significance gone.

Quiet Questions and Section Questions

People are getting quieter…

too quiet.

too, too quiet.

Final Projects: Save the date

December 12th

Block 1: 9:30 - 12pm
Block 2: 1pm - 3:30pm
Block 3: 4pm - 6:30pm

Two components:

- ★ Final presentation (Exact time TBD, but ideally 15 min for 2 person group)
- ★ Final report (details forthcoming and in dropbox, think “methods and results section of paper”)

Goals of this lecture

Today is about regression, not multilevel modeling.

Think about what the residuals of a regression equation means

Understand that the standard error and standard error estimate is a function of the randomness in the residuals only.

Present and unpack the

Heteroskedastic Robust Standard Errors (the Sandwich Estimator)

- ★ Classic OLS can be improved by allowing for heteroskedastic residuals.
- ★ Core idea: use the residuals to get a rough (bad) estimate of the variability of individual observations, and average to get overall good performance.

Thinking about classic regression and the uncertainty of the fixed effects

Canonical Regression Formula for OLS

An n-vector of outcomes

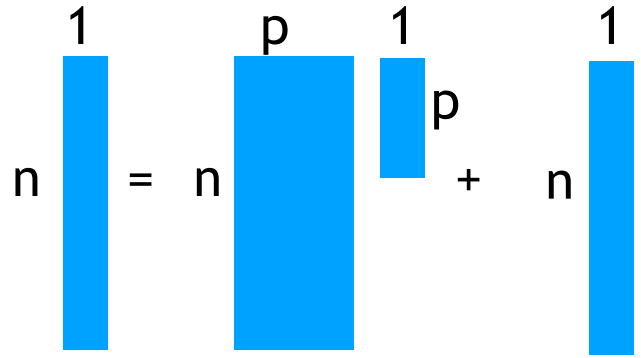
$$Y = X\beta + \epsilon$$

An n-vector of residuals

An n X p vector of covariates

A p-vector of coefficients to be estimated

X will have a column of 1s as the first "covariate" beta will have an initial coefficient for the intercept corresponding to this column of 1s



$$Y = X\beta + \epsilon$$

We can think of a regression (with fixed X) as a single matrix multiply.

This is useful when thinking about how the residual ***vector***, $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$, is random.

The OLS Estimator

The ordinary least squares estimator is

$$\hat{\beta} = (X'X)^{-1}X'Y$$

If our model is true: $Y = X\beta + \epsilon$

Then we have

$$\begin{aligned}\hat{\beta} &= (X'X)^{-1}X'(X\beta + \epsilon) \\ &= (X'X)^{-1}X'X\beta + (X'X)^{-1}X'\epsilon \\ &= \beta + (X'X)^{-1}X'\epsilon\end{aligned}$$

Matrix math cheat sheet

$$
\begin{aligned}
X &\text{ An } n \text{ by } p \text{ matrix } (n \text{ rows, } p \text{ columns}) \\
X' &\text{ “X transpose”} \\
X'X &\text{ “X transpose times X”} \\
(X'X)^{-1} &\text{ “The inverse of X transpose times X”}
\end{aligned}
$$

“Transpose” means flip on its side, so rows are columns and columns are rows.

“Inverse” is kind of like division. Core feature

$$A^{-1}A = I_p$$

The inverse times the original matrix is the “identity matrix” (A matrix with 1s on the diagonal and 0s elsewhere).

Think of matrix math as "batch processing": We can do all our predictions at once

Here we have 5 outcomes, and they are 5 predictions plus 5 residuals:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix} = \begin{pmatrix} X_1'\beta \\ X_2'\beta \\ X_3'\beta \\ X_4'\beta \\ X_5'\beta \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{pmatrix}$$

Each **prediction** is just OLS

$$\mu_i = X_i'\beta$$

Under this view, we can say different things about the residuals.

Our residuals have a **joint random distribution**

This is how we can think of 5 i.i.d. residuals, all normal and independent:

The nxn covariance matrix for the residuals' joint distribution

$$\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2 & & & & \\ 0 & \sigma^2 & & & \\ 0 & 0 & \sigma^2 & & \\ 0 & 0 & 0 & \sigma^2 & \\ 0 & 0 & 0 & 0 & \sigma^2 \end{pmatrix} \right]$$

Covariance matrices are always *positive-(semi)definite*, which means they are also always *symmetric* (you can flip the rows and columns and get the same thing).

For the above we can write the matrix $\Sigma = \sigma^2 I_n$

Note that Σ is capital sigma, and σ is lowercase sigma!

If we just have independence, but no heteroskedasticity, we have this

All the off-diagonals are still zero, but each residual has its own variance.

$$\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & & & & \\ 0 & \sigma_2^2 & & & \\ 0 & 0 & \sigma_3^2 & & \\ 0 & 0 & 0 & \sigma_4^2 & \\ 0 & 0 & 0 & 0 & \sigma_5^2 \end{pmatrix} \right]$$

But we can put anything we want on this matrix, if we want to get fancy.

This, for example, is an **autoregressive error structure**

The residuals that are closer together are **more correlated**

Our matrix has 2 parameters defining our overall structure

$$\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2 & & & & \\ \rho\sigma^2 & \sigma^2 & & & \\ \rho^2\sigma^2 & \rho\sigma & \sigma^2 & & \\ \rho^3\sigma^2 & \rho^2\sigma^2 & \rho\sigma^2 & \sigma^2 & \\ \rho^4\sigma^2 & \rho^3\sigma^2 & \rho^2\sigma^2 & \rho\sigma^2 & \sigma^2 \end{pmatrix}\right]$$

We can express the covariance between any two residuals as

$$cov(\epsilon_j, \epsilon_k) = \sigma^2 \rho^{|j-k|}$$

Core concept: What we want

Consider our plain ol' regression of

$$Y_i = \beta_0 + \beta_1 X_{1i} + ... + \beta_p X_{pi} + \epsilon_i$$

We *estimate* our **fixed effects** (the betas) with $\hat{\beta}_k$

$\hat{\beta} = (\hat{\beta}_0, \ldots, \hat{\beta}_k)$ is a *vector*, one element for each β_k

We then want the uncertainty of our estimates.

The ***variance-covariance matrix*** for $\hat{\beta}$ tell us how all our estimates vary and co-vary around the true β.

Variance-covariance what now?

Our model says our fixed portion (the $X'\beta$) is all *fixed* (i.e., it just is what it is, no randomess).

The randomness is all the residuals.

Different residuals give different estimates:

$$\hat{\beta} = (X'X)^{-1}X'Y = \beta + (X'X)^{-1}X'\epsilon$$

Classic expression for how to get OLS estimates.

Fixed, our true beta

Random due to residuals

What is this variance-covariance matrix for $\hat{\beta}$?

We generally assume

Note: this is NOT the covariance of random effects or residuals!!

$$\widehat{\beta} \sim N(\beta, \Sigma) = N\left[\begin{pmatrix}\beta_0 \\ \dots \\ \beta_p\end{pmatrix}, \begin{pmatrix}\sigma_{00} & \dots & \sigma_{p0} \\ \dots & \dots & \dots \\ \sigma_{p0} & \dots & \sigma_{pp}\end{pmatrix}\right]$$

This is the true sampling distribution of $\hat{\beta}$: it describes how our estimate can vary

The true betas

Whole matrix is Sigma

The standard errors are (the square root of) the **diagonal** of Σ.

If we then have an estimate of Σ, we can test and generate confidence intervals.

Everything depends on estimating Σ!

Looking at the variance-covariance matrix

Simple OLS with one covariate and one intercept (two fixed effects)

```
> m2 <- lm(growth ~ 1 + time, dat)
> arm::display( m2 )
lm(formula = growth ~ 1 + time, data = dat)
            coef.est coef.se
(Intercept) -3.07     0.62
time         1.37     0.05
---
n = 4023, k = 2
residual sd = 16.50, R-Squared = 0.15
> VC = vcov( m2 )
> VC
             (Intercept)          time
(Intercept)  0.38320854 -0.029390280
time        -0.02939028  0.002737223
> diag( VC )
(Intercept)        time
0.383208543 0.002737223
> sqrt( diag( VC ) )
(Intercept)        time
 0.61903840  0.05231848
```

This is the variance-covariance matrix. The covariance says if we randomly are over for our slope, we are likely under for our intercept.

Look, our standard errors for our two fixed effects!

The key concern

OLS will give us an estimated variance-covariance matrix

We then take the diagonal to get our standard errors (well, R does this for us automatically).

But this **relies on homoskedasticity**

What happens if we don't have that? How can we estimate Sigma some other way?

Doing this in a **heteroskedastic-robust** fashion is what the Sandwich estimator does.

National Youth Survey Example

Cross-Sectional data

This data comes from a survey in which the same students were asked yearly about their acceptance of 9 "deviant" behaviors (such as smoking marijuana, stealing, etc.).

The study began in 1976

Key variables:

- ★ "Attitude towards deviation" (our outcome)
- ★ Gender, minority status, family income

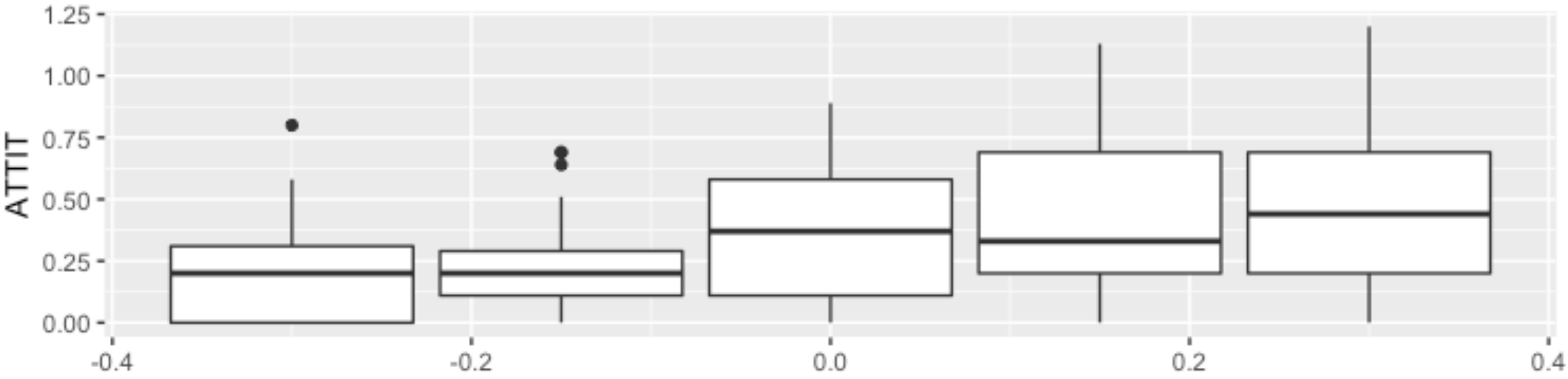
In our cross-sectional subset of this data we have 239 kids, with one observation per kid.

(I made this dataset by subsetting for illustration.)

What heteroskedasticity looks like

```
> nys1.solo %>% group_by( AGE ) %>%
+   summarise( mean.ATTIT = mean( ATTIT ),
+                 sd.ATTIT = sd( ATTIT ) )
# A tibble: 5 x 3
    AGE mean.ATTIT sd.ATTIT
  <int>      <dbl>    <dbl>
1    11      0.196    0.190
2    12      0.210    0.191
3    13      0.359    0.276
4    14      0.417    0.296
5    15      0.453    0.281
```

Note SDs are going up



Simple OLS and SEs: but no trust

```
> M0 = lm( ATTIT ~ 1 + AGE + FEMALE, data=nys1.solo )
> arm::display( M0 )
(Intercept) -0.52      0.15
AGE          0.07      0.01
FEMALE      -0.10      0.03
---
n = 239, k = 3
residual sd = 0.24, R-Squared = 0.18
>
> VC = vcov( M0 )
> VC
                (Intercept)           AGE        FEMALE
(Intercept)  0.0216521052 -1.631360e-03 -8.906064e-04
AGE         -0.0016313602  1.257271e-04  3.129524e-05
FEMALE      -0.0008906064  3.129524e-05  9.975721e-04
> diag( VC )
  (Intercept)           AGE        FEMALE
0.0216521052 0.0001257271 0.0009975721
>
> SE.homo = sqrt( diag( VC ) )
> SE.homo
(Intercept)         AGE      FEMALE
 0.14714654  0.01121281  0.03158437
```

Fixed effects and estimated standard errors

Variance-covariance of our fixed effects

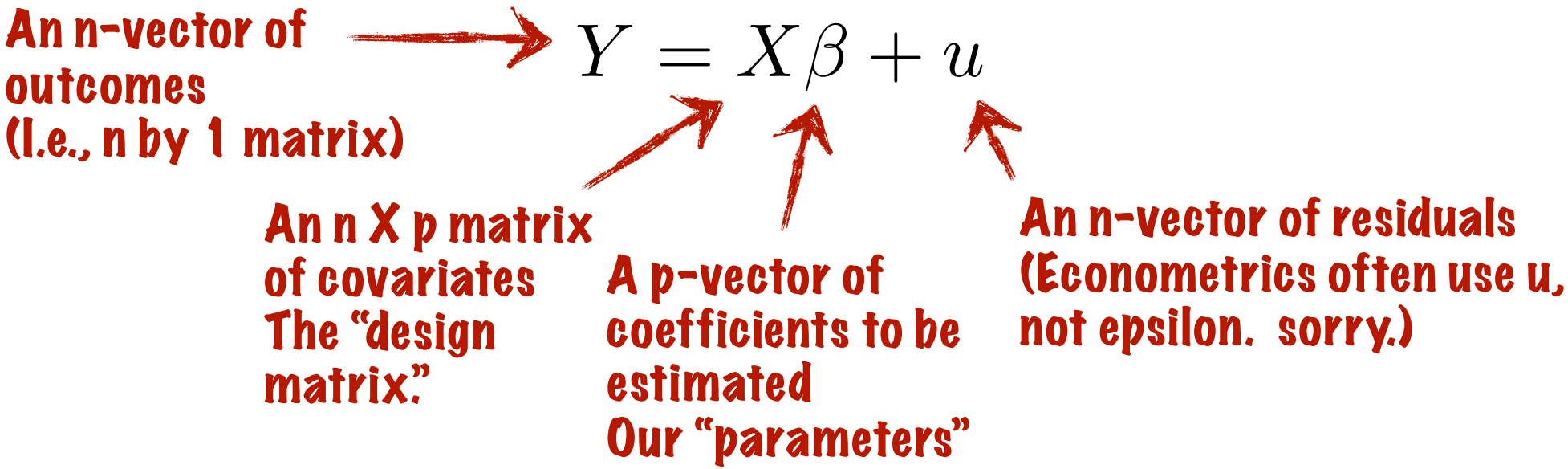
Our standard errors (under homoskedasticity)

QUESTION: How do we get SEs that account for heteroskedasticity?

The Sandwich Estimator

Which sandwich is your favorite?

The Canonical Regression Formula



We can think of a regression (with fixed X) as a single matrix multiply.

This is useful when thinking about how the residual vector, *u*, is random.

The covariance of the residuals

$$E[\epsilon\epsilon'] = \begin{bmatrix} E[\epsilon_1\epsilon_1] & & & & \\ E[\epsilon_1\epsilon_1] & E[\epsilon_2\epsilon_2] & & & \\ E[\epsilon_1\epsilon_1] & E[\epsilon_2\epsilon_3] & E[\epsilon_3\epsilon_3] & & \\ E[\epsilon_1\epsilon_1] & E[\epsilon_2\epsilon_4] & E[\epsilon_3\epsilon_4] & E[\epsilon_4\epsilon_4] & \\ E[\epsilon_1\epsilon_1] & E[\epsilon_2\epsilon_5] & E[\epsilon_3\epsilon_5] & E[\epsilon_4\epsilon_5] & E[\epsilon_5\epsilon_5] \end{bmatrix} = \Sigma$$

The $\epsilon\epsilon'$ is the *outer product*; it makes an nxn matrix.

The E[] is the *expectation. What we expect the random numbers to be. I.e., the average of the random numbers over many, many replications.*

The covariance of the residuals

$$E[\epsilon\epsilon'] = \begin{bmatrix} E[\epsilon_1\epsilon_1] & & & & \\ E[\epsilon_1\epsilon_1] & E[\epsilon_2\epsilon_2] & & & \\ E[\epsilon_1\epsilon_1] & E[\epsilon_2\epsilon_3] & E[\epsilon_3\epsilon_3] & & \\ E[\epsilon_1\epsilon_1] & E[\epsilon_2\epsilon_4] & E[\epsilon_3\epsilon_4] & E[\epsilon_4\epsilon_4] & \\ E[\epsilon_1\epsilon_1] & E[\epsilon_2\epsilon_5] & E[\epsilon_3\epsilon_5] & E[\epsilon_4\epsilon_5] & E[\epsilon_5\epsilon_5] \end{bmatrix} = \Sigma$$

The $\epsilon\epsilon'$ is the *outer product*; it makes an nxn matrix.

The E[] is the *expectation*. What we expect the random numbers to be. I.e., the average of the random numbers over many, many replications.

Important to note: $Var(\epsilon_i) = E[\epsilon_i^2] - E[\epsilon_i]^2 = E[\epsilon_i^2]$

If we assume independence

$$E[\epsilon\epsilon'] = \begin{bmatrix} E[\epsilon_1\epsilon_1] & & & & \\ 0 & E[\epsilon_2\epsilon_2] & & & \\ 0 & 0 & E[\epsilon_3\epsilon_3] & & \\ 0 & 0 & 0 & E[\epsilon_4\epsilon_4] & \\ 0 & 0 & 0 & 0 & E[\epsilon_5\epsilon_5] \end{bmatrix} = \Sigma$$

If the residuals are independent, then the correlation/covariance of any two residuals is 0

Elements of our diagonal is then $E[\epsilon_k\epsilon_k] = \sigma_k^2$

The true variance of our estimator

$$Var(\hat{\beta}) = Var\left[\beta + (X'X)^{-1}X'\epsilon\right]$$

This is matrix math.

$$= E\left[(X'X)^{-1}X'\epsilon\epsilon'X(X'X)^{-1}\right]$$

$$= (X'X)^{-1}X'E\left[\epsilon\epsilon'\right]X(X'X)^{-1}$$

This is what you should focus on

$$= (X'X)^{-1}X'\Sigma X(X'X)^{-1}$$

This is our variance-covariance matrix for our parameters.

This expression is true, no matter what are residuals are up to.

Core concern is then:

How do you get an estimate of this?

Dissecting our variance formula

$$Var(\hat{\beta}) = (X'X)^{-1} \cdot X\Sigma X \cdot (X'X)^{-1}$$

Our p-vector of estimates

X is a nXp design matrix So (X'X) and $(X'X)^{-1}$ are pXp matrices

This is our covariance matrix of the full residual vector (n x n)

These $(X'X)^{-1}$ terms are often called "the bread"

To estimate our variance-covariance matrix, we need to estimate this.

What elements do we know?

What do we need to estimate?

Note: If you assume homoskedasticity, this simplifies to our old friend $Var(\hat{\beta}) = \sigma^2 (X'X)^{-1}$

To estimate, we pull a rabbit out of a hat

Robust Standard Errors: Estimating our residual covariance with a "plug in"

$$X'\hat{\Sigma}X = X'\begin{bmatrix} \hat{u}_1^2 & 0 & 0 \\ 0 & \hat{u}_2^2 & 0 \\ & & \\ 0 & 0 & \hat{u}_n^2 \end{bmatrix} X$$

We use our residuals for each observation to estimate the variance of each observation.

For each individual observation, this estimate is *terrible*.

Happily, when we average over them all (which the matrix multiply does) our average estimate is good!

Behold!
The Huber-White Sandwich Estimator

$$Var(\hat{\beta}) = (X'X)^{-1} X'\hat{\Sigma}X (X'X)^{-1}$$

Bread — Meat — More Bread

The square root of the diagonal of this matrix gives

Heteroskedastic-Robust Standard Errors

Extension are possible. For example, we can put structure on (parameterize) Sigma to get improved estimation, based on further assumptions.

The Sandwich as a Sum

$$X'\hat{\Sigma}X = \sum_{i=1}^{n} \hat{u}_i^2 X_i X_i'$$

giving

$$Var(\hat{\beta}) = (X'X)^{-1} \left(\sum_{i=1}^{n} \hat{u}_i^2 X_i X_i' \right) (X'X)^{-1}$$

Side note: The Matrix Math for our meat looks like this:

X' x $\hat{\Sigma}$ x X = meat

we get a p X p matrix at end.

Doing it in R (sandwich package)

Get the "variance covariance" matrix for the model. HC0 is the basic type. Other types exist (HC1, HC2)

```
> M0 = lm( ATTIT ~ 1 + AGE + FEMALE, data=nys1.solo )
> library( sandwich )
> vcov_sand = vcovHC(M0, type = "HC0")
> vcov_sand
             (Intercept)         AGE    FEMALE
(Intercept)      0.02067 -0.001589 -0.001327
AGE             -0.00159  0.000125  0.000061
FEMALE          -0.00133  0.000061  0.000994

> SE.sand <- sqrt( diag( vcov_sand ) )
> SE.sand
(Intercept)         AGE      FEMALE
     0.1438      0.0112      0.0315

> se.coef(M0)
(Intercept)         AGE      FEMALE
     0.1471      0.0112      0.0316
```

This is a p x p matrix. The diagonal are your SEs squared

Here our original SEs are the same—so we did all of this for security, but it didn't change anything.

Sometimes it can be a big deal, sometimes not.

Doing it in R (sandwich package)

lmtest does testing with the sandwich package quite nicely.

```
> library( lmtest )
> coeftest( M0, vcov. = vcov_sand )
t test of coefficients:

             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.5231     0.1438   -3.64  0.00034 ***
AGE           0.0692     0.0112    6.18  2.8e-09 ***
FEMALE       -0.1009     0.0315   -3.20  0.00156 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1

> coeftest( mod, vcov. = vcovHC )
```

You can hand coeftest the name of the var-cor generating function instead (a shortcut)

Recap of robust standard errors

So what does all of this mean?

Take-aways

- ★ The sandwich estimator is about getting a good estimate of the variance-covariance matrix for your fixed effect estimator.
- ★ The core idea of thinking of a single "marginal model" and then a giant NxN covariance matrix of the residuals, is what we will next build on when extending these ideas to clustering
  - Cluster robust standard errors just uses this idea of residuals estimating themselves, but within each group of observations
  - Alternatively we can put a parametric model on each cluster, giving us a *parameterized* giant NxN covariance matrix.
  - The next two classes unpack these two ideas.

Appendix

(Optional) Doing it in R (by hand)

```
> mod <- lm(mathach ~ ses + sector, data = dat)
> resids = resid( mod )

> X <- model.matrix(mathach ~ ses + sector, data = dat)

> V <- solve(t(X) %*% X) ## the bread
> vcov_hw = V %*% t(X) %*% diag(resids^2) %*% X %*% V
```

The diag() command makes a big matrix. Iteration is often better.

```
> vcov_hw
             (Intercept)          ses       sector
(Intercept)  0.012142174  0.001957716 -0.012535538
ses          0.001957716  0.008997088 -0.003992666
sector      -0.012535538 -0.003992666  0.023942897
> sqrt(diag(vcov_hw)) ## standard errors
(Intercept)         ses      sector
 0.11019153  0.09485298  0.15473493
> sqrt( diag( vcov( mod ) ) )
(Intercept)         ses      sector
 0.10610213  0.09783058  0.15249341
```

Our SEs are from the diagonal of our matrix

The conventional estimates happen to be basically the same in this case.