Standard Errors in OLS

Luke Sonnet

Contents

This document reviews common approaches to thinking about and estimating uncertainty of coefficients estimated via OLS. Much of the document is taken directly from these very clear notes, Greene's Econometric Analysis, and slides by Chad Hazlett. This document was originally designed for first-year students in the UCLA Political Science statistics sequence.

Variance-Covariance of $\hat{\boldsymbol{\beta}}$

Take the classic regression equation

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ is an $n \times 1$ outcome vector, $\mathbf{X}$ is an $n \times p$ matrix of covariates, $\boldsymbol{\beta}$ is an $n \times 1$ vector of coefficients, and $\boldsymbol{\epsilon}$ is an $n \times 1$ vector of noise, or errors. Using OLS, our estimate of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

This is just an estimate of the coefficients. We also would like to understand the variance of this estimate to quantify our uncertainty and possibly to perform significance tests. We can derive an explicit function that represents the variance of our estimates, $\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]$, given that $\mathbf{X}$ is fixed.

What we are interested in is $\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]$, which is the variance of all the estimated coefficients $\hat{\boldsymbol{\beta}}$ and the covariance between our coefficients. We can represent this as

$$\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}] = \begin{bmatrix} \mathbb{V}[\hat{\beta}_0|\mathbf{X}] & \text{Cov}[\hat{\beta}_0, \hat{\beta}_1|X] & \cdots & \text{Cov}[\hat{\beta}_0, \hat{\beta}_p|X] \\ \text{Cov}[\hat{\beta}_1, \hat{\beta}_0|X] & \mathbb{V}[\hat{\beta}_1|\mathbf{X}] & \cdots & \text{Cov}[\hat{\beta}_1, \hat{\beta}_p|X] \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}[\hat{\beta}_p, \hat{\beta}_0|X] & \text{Cov}[\hat{\beta}_p, \hat{\beta}_1|X] & \cdots & \mathbb{V}[\hat{\beta}_p|\mathbf{X}] \end{bmatrix}$$

Our goal is to estimate this matrix. Why? Often because we want the standard errors of the jth coefficient, $\text{se}(\hat{\beta}_j)$. We get this by taking the square root of the diagonal of $\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]$. Therefore, our focal *estimand* is,

$$\text{se}(\hat{\boldsymbol{\beta}}) = \begin{bmatrix} \sqrt{\mathbb{V}[\hat{\beta}_0|\mathbf{X}]} \\ \sqrt{\mathbb{V}[\hat{\beta}_1|\mathbf{X}]} \\ \vdots \\ \sqrt{\mathbb{V}[\hat{\beta}_p|\mathbf{X}]} \end{bmatrix}$$

To show how we get to an estimate for this quantity, first note that,

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \\ &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\epsilon}) \\ &= \boldsymbol{\beta} + (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\epsilon} \\ \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\epsilon}\end{aligned}$$

$$\begin{aligned}\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}] &= \mathbb{E}[(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})^\top|\mathbf{X}] \\ &= \mathbb{E}[(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\epsilon}((\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\epsilon})^\top|\mathbf{X}] \\ &= \mathbb{E}[(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}|\mathbf{X}] \\ &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}]\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\end{aligned}$$

This then is our answer for the variance-covariance matrix of our coefficients $\hat{\boldsymbol{\beta}}$. While we have $\mathbf{X}$, we do not have $\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}]$, which is the variance-covariance matrix of the errors. What is this matrix? It captures the scale of the unobserved noise in our assumed data generating process as well as how that noise is covaries between units.

This matrix has $n \times n$ unknown parameters that define the variance of each units' error and the covariance between errors of different units. Because these parameters are unknown, there are many of them, and they describe fairly complex processes, we often make simplifying assumptions to estimate fewer of these parameters. In general we cannot estimate the full matrix $\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}]$.

What if we assume that all units have errors with the same variance? Then we are assuming homoskedasticity. Google heteroskedasticity for graphical representations of when this is violated. If we assume that errors covary within particular groups, then we should build this structure into your estimates of $\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}]$, as one does when they estimate cluster robust standard errors. In this document, I run through three of the most common cases. The standard case when we assume spherical errors (no serial correlation and no heteroskedasticity), the case where we allow heteroskedasticity, and the case where there is grouped correlation in the errors. In all cases we assume that the conditional mean of the error is 0. Precisely $\mathbb{E}[\epsilon|X] = 0$.

If we get our assumptions about the errors wrong, then our standard errors will be biased, making this topic pivotal for much of social science. Of course, your assumptions will often be wrong anyays, but we can still strive to do our best.

Standard Estimation (Spherical Errors)

Assuming spherical errors–no heteroskedasticity and no serial correlation in the errors–is historically the chief assumption in estimating variance of OLS estimates. However, because it is relatively easy to allow for heteroskedasticity (as we will see below), and because assuming spherical errors is often incredibly unrealistic, these errors are not longer used in the majority of published work. Nonetheless, I present it here first as it is the simplest and one of the oldest ways of estimating variance of OLS estimates.

In this case, we assume that all errors have the same variance and that there is no correlation across errors. This looks like the following:

$$\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}] = \begin{bmatrix}\sigma^2 & 0 & \cdots & 0 \\ 0 & \sigma^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma^2\end{bmatrix} = \sigma^2\mathbf{I}$$

Therefore, all errors have the same variance, some scalar σ^2. Then the variance of our coefficients simplifies,

$$\begin{aligned}
\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}] &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}]\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} \\
&= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\sigma^2\mathbf{I}\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} \\
&= \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}
\end{aligned}$$

Now all we need is an estimate of σ^2 in order to get our estimate for $\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]$. I do not show this here, but an unbiased estimate for σ^2 is,

$$\hat{\sigma^2} = \frac{\mathbf{e}^\top\mathbf{e}}{n-p}$$

where $\mathbf{e} = \hat{\mathbf{y}} - \mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{y}$ is the vector of residuals, and n is the number of observations and p is the number of covariates.

Thus our estimate of $\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]$ is

$$\widehat{\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]} = \frac{\mathbf{e}^\top\mathbf{e}}{n-p}(\mathbf{X}^\top\mathbf{X})^{-1}$$

The diagonal of this matrix is our estimated variance for each coefficient, the square root of which is the familiar standard error that we often use to construct confidence intervals or perform significance tests.

Let's see this in `R`

```r
## Construct simulated data and errors
set.seed(1)
X <- cbind(1, rnorm(100), runif(100))

set.seed(2)
eps <- rnorm(100)

beta <- c(1, 2, 3)
y <- X %*% beta + eps

## Manual solutions
## Beta hat
beta_hat <- solve(t(X) %*% X, t(X) %*% y)
beta_hat
```

```
##          [,1]
## [1,] 1.067999
## [2,] 1.806047
## [3,] 2.821665
```

```r
## Residuals
resid <- y - X %*% beta_hat
## Estimate of sigma_2
sigma2_hat <- (t(resid) %*% resid) / (nrow(X) - ncol(X))
sigma2_hat
```

```
##          [,1]
## [1,] 1.338826
```

```r
## Estimate of V[\hat{\bbeta}]
vcov_beta_hat <- c(sigma2_hat) * solve(t(X) %*% X)
vcov_beta_hat
```

```
##               [,1]          [,2]         [,3]
## [1,]  0.0463264144  0.0001312435 -0.075750093
## [2,]  0.0001312435  0.0168795926 -0.004526778
## [3,] -0.0757500928 -0.0045267783  0.175265100
```

```
## Estimate of standard errors
sqrt(diag(vcov_beta_hat))
```

```
## [1] 0.2152357 0.1299215 0.4186467
```

This leaves us with the following coefficients and standard error estimates:

```
cbind(beta_hat, sqrt(diag(vcov_beta_hat)))
```

```
##          [,1]      [,2]
## [1,] 1.067999 0.2152357
## [2,] 1.806047 0.1299215
## [3,] 2.821665 0.4186467
```

Let's show the same thing using `lm`.

```
lm_out <- lm(y ~ 0 + X)
cbind(lm_out$coefficients, coef(summary(lm_out))[, 2])
```

```
##        [,1]      [,2]
## X1 1.067999 0.2152357
## X2 1.806047 0.1299215
## X3 2.821665 0.4186467
```

Looks good!

Robust Estimation (Heteroskedasticity Constistent Errors)

Almost always, the assumption that our errors are homoskedastic is unrealistic. A simple example would be where variance is greater for units with higher values of some covariate X. A concrete example could be where income is the outcome and age is the explanatory variable. Among young individuals, income is probably less variable than among older individuals and thus the spread of income around the average income is greater for older individuals than for younger individuals. Another way to think of this is that our observations are still independent, but they are not identically distributed because they have different variance. In any case, one generally need not come up with an explanation for why they might use heteroskedasticity robust standard errors, as it is generally assumed that heteroskedasticity is likely to be a problem and the cost of estimating variance this way is low.

Heteroskedasticity is not a problem for coefficients, but it does bias our estimates of the standard errors. We can get White's heteroskedasticity consistent standard errors, or robust standard errors, by assuming something else for the variance-covariance of the errors ($\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}]$) and choosing a different estimator.

Instead of forcing all diagonal elements of $\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}]$ to be a single scalar, what if we allow them all to be different? This accounts for all kinds of heteroskedasticity, because each error is allowed to have a different variance. Precisely,

$$\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}] = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix}$$

Thus we now have n different variances, σ_i^2. Then the variance of our coefficients simplifies,

$$\begin{aligned}\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}] &= (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbb{E}[\text{diag}[\sigma_i^2]|\mathbf{X}]\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1} \\ &= (\mathbf{X}^\top\mathbf{X})^{-1}\frac{1}{n}\sum_{i=1}^{n}\sigma_i^2\mathbf{x_i}\mathbf{x_i}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\end{aligned}$$

Then, White shows in his often cited 1980 paper, that, $\frac{1}{n}\sum_{i=1}^{n} e_i^2\mathbf{x_i}\mathbf{x_i}^\top$ is a consistent, but biased, estimator for $\frac{1}{n}\sum_{i=1}^{n}\sigma_i^2\mathbf{x_i}\mathbf{x_i}^\top$ where $\mathbf{x_i}$ is the $p\times 1$ vector of covariates for observation i. So $\mathbb{E}[\mathbf{X}^\top\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\mathbf{X}|\mathbf{X}]$ is consistently but biasedly estimated by $\frac{1}{n}\sum_{i=1}^{n} e_i^2\mathbf{x_i}\mathbf{x_i}^\top$. Thus, we can write our estimate for the variance as

$$\widehat{\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]}_{HC} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\text{diag}[e_i^2]\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}$$

To be clear $\text{diag}[e_i^2]$ is a diagonal matrix with each element on the diagonal being observation i's residual squared. All of these quantities are all observed, so we can directly compute the heteroskedasticity robust variance covariance matrix and standard errors.

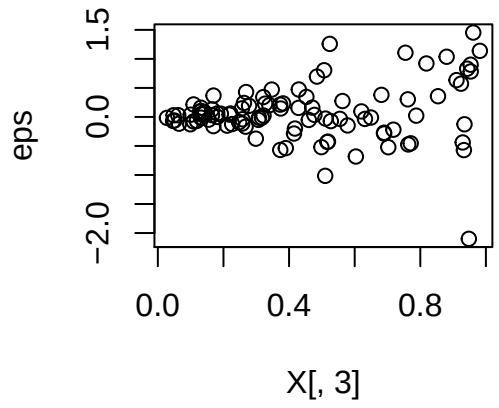
However, it is now standard to use a finite sample correction for the bias in this estimator. While the estimate is consistent, it is biased and thus when the sample is not infinite, a correction can be used to improve the bias. There are several different corrections we can use. A simple one, and the one used by default in Stata, is the `HC1` robust variance covariance matrix. This is simply

$$\widehat{\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]}_{HC1} = \frac{n}{n-p}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\text{diag}[e_i^2]\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}$$

Thus all we are doing is multiplying the elements by $\frac{n}{n-p}$ which will be close to 1 if we have many more observations n than covariates p. However, it is probably preferable to use `HC2` or `HC3`, but I will not go into those here for the sake of simplicity.

Let's do this in `R`:

```
## Noise that is large for higher values of X[, 3]
set.seed(1)
eps <- rnorm(100, 0, sd = X[, 3])
plot(X[, 3], eps)
```



```r
y <- X %*% beta + eps

## Manual solutions
## Beta hat
beta_hat <- solve(t(X) %*% X, t(X) %*% y)
beta_hat
```

```
##           [,1]
## [1,] 0.9503923
## [2,] 2.4367714
## [3,] 3.1610179
```

Now let's get the `HC1` robust standard errors.

```r
## Residuals
resid <- y - X %*% beta_hat
sigma2_hat <- t(resid) %*% resid / (nrow(X) - ncol(X))
## Standard, non-robust estimate
vcov_beta_hat <- c(sigma2_hat) * solve(t(X) %*% X)
vcov_beta_hat
```

```
##               [,1]          [,2]          [,3]
## [1,]  2.479749e-03  7.025170e-06 -0.0040547332
## [2,]  7.025170e-06  9.035269e-04 -0.0002423083
## [3,] -4.054733e-03 -2.423083e-04  0.0093815492
```

```r
## Robust HC1 stimate of V[\hat{\bbeta}]
vcov_rob_beta_hat <- nrow(X)/(nrow(X) - ncol(X)) *
  solve(t(X) %*% X) %*% t(X) %*% diag(c(resid^2)) %*% X %*% solve(t(X) %*% X)
vcov_rob_beta_hat
```

```
##              [,1]         [,2]         [,3]
## [1,]  0.003743534  0.000355192 -0.008265779
## [2,]  0.000355192  0.003046248 -0.002539765
## [3,] -0.008265779 -0.002539765  0.022678946
```

```r
## Display results
outmat <- cbind(beta_hat, sqrt(diag(vcov_beta_hat)),  sqrt(diag(vcov_rob_beta_hat)))
colnames(outmat) <- c("Beta Hat", "Standard SE", "HC1 Robust SE")
outmat
```

```
##       Beta Hat Standard SE HC1 Robust SE
## [1,] 0.9503923  0.04979708    0.06118443
## [2,] 2.4367714  0.03005872    0.05519282
## [3,] 3.1610179  0.09685840    0.15059531
```

We can do this using `lm` and the `sandwich` package.

```r
lmout <- lm(y ~ 0 + X)
library(sandwich)
## HC1 Robust
vcov_rob_beta_hat <- vcovHC(lmout, type = "HC1")
## HC2 Robust
vcov_robHC2_beta_hat <- vcovHC(lmout, type = "HC2")
## HC3 Robust
vcov_robHC3_beta_hat <- vcovHC(lmout, type = "HC3")
outmat <- cbind(lmout$coefficients,
                coef(summary(lmout))[, 2],
```

```
                sqrt(diag(vcov_rob_beta_hat)),
                sqrt(diag(vcov_robHC2_beta_hat)),
                sqrt(diag(vcov_robHC3_beta_hat)))
colnames(outmat) <- c("Beta Hat",
                      "Standard SE",
                      "HC1 Robust SE",
                      "HC2 Robust SE",
                      "HC3 Robust SE")
outmat
```

```
##     Beta Hat Standard SE HC1 Robust SE HC2 Robust SE HC3 Robust SE
## X1 0.9503923  0.04979708    0.06118443    0.06235143    0.06454567
## X2 2.4367714  0.03005872    0.05519282    0.05704224    0.05989300
## X3 3.1610179  0.09685840    0.15059531    0.15474172    0.16155457
```

The biggest difference is between the regular standard errors and the robust standard errors. The finite corrections are only slightly diferent from one another.

Shameless plug: we can easily get robust standard erors using `lm_robust` from the `estimatr` package.

```
library(estimatr)
lm_robust(y ~ 0 + X, se_type = "HC1")
```

```
##     Estimate Std. Error  t value     Pr(>|t|)  CI Lower CI Upper DF
## X1 0.9503923 0.06118443 15.53324 4.650495e-28 0.8289582 1.071826 97
## X2 2.4367714 0.05519282 44.15015 4.952694e-66 2.3272289 2.546314 97
## X3 3.1610179 0.15059531 20.99015 7.609783e-38 2.8621279 3.459908 97
```

```
lm_robust(y ~ 0 + X, se_type = "HC2")
```

```
##     Estimate Std. Error  t value     Pr(>|t|) CI Lower CI Upper DF
## X1 0.9503923 0.06235143 15.24251 1.715659e-27 0.826642 1.074143 97
## X2 2.4367714 0.05704224 42.71872 1.037656e-64 2.323558 2.549984 97
## X3 3.1610179 0.15474172 20.42770 6.555414e-37 2.853898 3.468137 97
```

```
lm_robust(y ~ 0 + X, se_type = "HC3")
```

```
##     Estimate Std. Error  t value     Pr(>|t|)  CI Lower CI Upper DF
## X1 0.9503923 0.06454567 14.72434 1.803574e-26 0.8222871 1.078498 97
## X2 2.4367714 0.05989300 40.68541 9.181786e-63 2.3179004 2.555642 97
## X3 3.1610179 0.16155457 19.56626 1.908508e-35 2.8403768 3.481659 97
```

Cluster Robust Estimation

Another problem is that your data may be clustered. You may have groups of observations that are exposed to similar random events, or whose responses to an event are not unrelated to the responses of others in that group. In this case, we will assume no dependence across groups, but estimate variance and covariance of uncertainty within gruops. For example, imagine studying the performance of students in different classrooms. Those in the same classroom are likely to receive similar "shocks" or random effects that those in other classrooms will not. We need to account for this clustering in our data.

Again, this is not a problem for our coefficients. However, the variance covariance matrix of the errors now has a clustered structure. Let's imagine we have m groups, and each group has n_m observations. Then we can write the variance covariance matrix of the errors as

$$\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top|\mathbf{X}] = \begin{bmatrix} \sigma^2_{(1,1)1} & \cdots & \sigma^2_{(1,n_1)1} & 0 & \cdots & 0 & & & \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & & \\ \sigma^2_{(n_1,1)1} & \cdots & \sigma^2_{(n_1,n_1)1} & 0 & \cdots & 0 & & & \\ 0 & \cdots & 0 & \sigma^2_{(1,1)2} & \cdots & \sigma^2_{(1,n_2)2} & & & \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & & \\ 0 & \cdots & 0 & \sigma^2_{(n_2,1)2} & \cdots & \sigma^2_{(n_2,n_2)2} & & & \\ & & & & & & \ddots & & \\ & & & & & & & \sigma^2_{(1,1)m} & \cdots & \sigma^2_{(1,n_m)m} \\ & & & & & & & \vdots & \ddots & \vdots \\ & & & & & & & \sigma^2_{(n_m,n_m)m} & \cdots & \sigma^2_{(n_m,n_m)m} \end{bmatrix}$$

Thus we can write the variance covariance of our coefficients as

$$\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}] = (\mathbf{X}^\top\mathbf{X})^{-1}\sum_{g=1}^{m}\mathbf{x_g}^\top\boldsymbol{\epsilon}_g\boldsymbol{\epsilon}_g^\top\mathbf{x_g}(\mathbf{X}^\top\mathbf{X})^{-1}$$

where $\mathbf{x_g}$ is an $n_g \times p$ matrix of all p covariates for the observations in group g and $\boldsymbol{\epsilon}_g$ is an $n_g \times 1$ vector of errors for the n_g observations in group g. So we have this block structure where we have a full variance covariance matrix and we need to estimate the blocks of errors. Without getting into the derivation, we can use $\sum_{g=1}^{m}\mathbf{e}_g\mathbf{e}_g\mathbf{x_g}\mathbf{x_g}^\top$ to estimate $\sum_{g=1}^{m}\boldsymbol{\epsilon}_g\boldsymbol{\epsilon}_g^\top\mathbf{x_g}\mathbf{x_g}^\top$. Thus our estimated variance covariance matrix of the coefficients is

$$\widehat{\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]}_{CR} = (\mathbf{X}^\top\mathbf{X})^{-1}\sum_{g=1}^{m}\mathbf{x_g}^\top\mathbf{e}_g\mathbf{e}_g\mathbf{x_g}(\mathbf{X}^\top\mathbf{X})^{-1}$$

We also apply a finite sample correction to this estimator because it is biased in finite samples. The standard "fancy" corrected estimator that Stata uses is

$$\widehat{\mathbb{V}[\hat{\boldsymbol{\beta}}|\mathbf{X}]}_{CR_{fancy}} = \frac{m}{m-1}\frac{n-1}{n-p}(\mathbf{X}^\top\mathbf{X})^{-1}\sum_{g=1}^{m}\mathbf{x_g}^\top\mathbf{e}_g\mathbf{e}_g\mathbf{x_g}(\mathbf{X}^\top\mathbf{X})^{-1}$$

Again, as m and n go to infinite, the correction will go to 1. This should make it obvious that a small number of clusters will require a bigger correction from the first term.

Let's do this in R.

```
## Generate epsilon from correlated matrix
## 10 groups, same blocks but this is not necessary
library(clusterGeneration)
```

```
## Loading required package: MASS
```

```
library(mvtnorm)
block_eps <- genPositiveDefMat(10)
sigma_eps <- kronecker(diag(10), block_eps$Sigma)
eps <- rmvnorm(1, mean = rep(0, 100), sigma = sigma_eps/4)
groups <- rep(1:10, each = 10)
groups
```

```
##   [1]  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  3  3  3
##  [24]  3  3  3  3  3  3  3  4  4  4  4  4  4  4  4  4  4  5  5  5  5  5  5
##  [47]  5  5  5  5  6  6  6  6  6  6  6  6  6  6  7  7  7  7  7  7  7  7  7
```

```
##  [70]  7  8  8  8  8  8  8  8  8  8  9  9  9  9  9  9  9  9  9 10 10
##  [93] 10 10 10 10 10 10 10 10
```

```
y <- X %*% beta + t(eps)

## Manual solutions
## Beta hat
beta_hat <- solve(t(X) %*% X, t(X) %*% y)
beta_hat
```

```
##           [,1]
## [1,] 0.8392765
## [2,] 2.1686256
## [3,] 3.3014213
```

```
## Residuals
resid <- y - X %*% beta_hat
sigma2_hat <- 1/(nrow(X) - ncol(X)) * c(t(resid) %*% resid)
## Standard, non-robust estimate
vcov_beta_hat <- c(sigma2_hat) * solve(t(X) %*% X)
vcov_beta_hat
```

```
##               [,1]          [,2]         [,3]
## [1,]  0.0382446856  0.0001083478 -0.062535349
## [2,]  0.0001083478  0.0139349164 -0.003737073
## [3,] -0.0625353487 -0.0037370734  0.144689779
```

```
## Cluster Robust estimate of V[\hat{\bbeta}]
meat <- matrix(0, nrow = ncol(X), ncol = ncol(X))
for (g in 1:10) {
  meat <- meat + t(X[groups == g, ]) %*% resid[groups == g] %*%
    t(resid[groups == g]) %*% X[groups == g, ]
}
vcov_crob_beta_hat <- (10/(10-1)) * ((100 - 1)/(100 - 3)) *
  solve(t(X) %*% X) %*% meat %*% solve(t(X) %*% X)
vcov_crob_beta_hat
```

```
##              [,1]         [,2]         [,3]
## [1,]  0.039699729  0.009047246 -0.058446415
## [2,]  0.009047246  0.022368271 -0.005527682
## [3,] -0.058446415 -0.005527682  0.125846996
```

```
## Display results
outmat <- cbind(beta_hat, sqrt(diag(vcov_beta_hat)),  sqrt(diag(vcov_crob_beta_hat)))
colnames(outmat) <- c("Beta Hat", "Standard SE", "Cluster Robust SE")
outmat
```

```
##       Beta Hat Standard SE Cluster Robust SE
## [1,] 0.8392765   0.1955625         0.1992479
## [2,] 2.1686256   0.1180462         0.1495603
## [3,] 3.3014213   0.3803811         0.3547492
```

R does not have a built in function for cluster robust standard errors. Also, while there are scripts online to do this, estiamting cluster robust standard errors in `estimatr` is very easy.

```
## Put data in data.frame
df <- as.data.frame(cbind(y, X, groups))
names(df) <- c("y", "x1", "x2", "x3", "groups")
```

```r
## Fit model
library(estimatr)
lm_robustout <- lm_robust(y ~ x2 + x3, data = df, clusters = groups, se_type = "stata")

## Display results
outmat <- cbind(beta_hat, sqrt(diag(vcov_beta_hat)),  lm_robustout$std.error)
colnames(outmat) <- c("Beta Hat", "Standard SE", "Cluster Robust SE")
outmat
```

```
##              Beta Hat Standard SE Cluster Robust SE
## (Intercept) 0.8392765   0.1955625         0.1992479
## x2          2.1686256   0.1180462         0.1495603
## x3          3.3014213   0.3803811         0.3547492
```

Same as above!

Some comments

Why would you use regular standard errors if heteroskedastic standard errors and clustered standard errors both allow for more complicated error structures?

Homoskedasticity is simply a special case of the heteroskedastic error structure; it is simply the case where $\sigma_j = \sigma_i$ for all i and j. So using heteroskedastic standard errors will always handle the case of homoskedasticity and will always be safe in that way. However:

- Regular standard errors do not have finite sample bias. So if we truly believe homoskedasticity to be true, then we can avoid finite sample bias by using the regular standard errors.
- Furthermore, if homoskedasticity actually is true, then our estimates of the standard errors will be more efficient. This means it will approach the true value faster (as the sample size grows), then heteroskedastic standard errors.
- However, we rarely believe that errors actually are homoskedastic, and it is often best to use the heteroskedasticity robust standard errors

Remember, the error structure is not important for unbiasedness of $\hat{\beta}$ as long as it has conditional mean 0

Review your notes for the proof that $\hat{\boldsymbol{\beta}}$ is an unbiased estimator for $\boldsymbol{\beta}$. Never do we use the variance-covariance matrix, $\mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top | X]$. All we use is the conditional mean of $\boldsymbol{\epsilon}$. This whole discussion is about the biasedness of our estimates for $\mathbb{V}[\hat{\boldsymbol{\beta}}]$, which is our estimate of uncertainty and is how we do hypothesis testing.

Normally Distributed Errors

We have been very focused on the variance-covariance of $\boldsymbol{\epsilon}$, but not on how those errors have been distributed. For example, it is often stated that we assume that the errors are **normally distributed**. The normality of the errors is not necessary for unbiasedness of either $\hat{\boldsymbol{\beta}}$ or $\widehat{\mathbb{V}[\hat{\boldsymbol{\beta}}]}$. So why do people make that assumption?

- Normality is somewhat important for significance testing. Specifically, with normal, independent standard errors we can be assured the $\hat{\boldsymbol{\beta}}$ is distributed normally even in finite samples. This means we can get t-statistic that is actually t-distributed. Thus it is important for significance testing, but not for the standard errors themselves. Nonetheless, even without normal errors, $\hat{\boldsymbol{\beta}}$ will still be distributed normally asymptotically, meaning as the sample size goes to ∞. Furthermore, our test statistic will also be normally distributed and thus asymptotically significance testing will also be valid. These are

the result of the central limit theorem. This generally means that if you have a very large sample (where "very large" is intentionally vague), then the assumption is not necessary for significance testing. However, in finite samples, the normality assumption guarantees that your confidence intervals and p-values are correct.

- Normality (or perhaps other *specific* distributional assumptions) is necessary for the "best" or minimum variance of the OLS estimator in finite samples.
- Normality of the errors is needed for the standard normal linear model if you fit it using maximum likelihood. You will learn about this later in the sequence; it returns the same coefficients as OLS, but the framework is different. So this is largely about how you conceptualize regression.