Linear Regression Model

Statistical Inference and Extensions

Fernando Rios-Avila

Introduction

- Linear Regression is the most basic, and still most useful, tool for analyzing data.
- The goal is to find what the relationship between the outcome y and explanatory variables $X's$ is.
- Say that we start with a very simple "***model***" that states tries to describe the population function as the following:

$$y = h(X, \varepsilon)$$

Here, X represents a set of observed covariates and ε the set of unobserved characteristics, with no no pre-defined relationship between these components.

- For now, we will make standard exogeneity assumptions for the identification of the model

Estimation

- The functional form is unknowable. However, under the ***small*** assumption of Exogeneity of X, we could instead consider the Conditional Expectation function (CEF):

$$E(y_i|X_i = x) = \int y f_{y|x}(y) dy$$

- This implies a fully **non-parametric** estimation of the Linear function.
- With this, the outcome y can be decomposed into factors determined by observed characteristics (CEF) and on the error ε.

$$y = E(y|X) + \varepsilon$$

- The CEF is a convenient abstract, but to estimate it, we require assumptions. (Recall the assumptions for unbiased OLS?)
- Namely, we need to impose a linearity assumption, namely:

$$E(y_i|X_i = x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k = X_i'\beta$$

- And the solution for β is given by:

$$\beta = \underset{b}{arg}\ E(L(y_i - X_i'b))$$

Where the loss function $L(x) = x^2$. (Square loss function)

- This implies the following condition: $E[X_i(y_i - X_i'b)] = 0 \rightarrow \beta = E[X_i'X_i]^{-1}E[X_i'y_i]$

Mata: OLS Estimator

The estimator using Sample equivalents become:

$$\hat{\beta} = \left(\frac{1}{N}\sum_i X_i'X_i\right)^{-1}\frac{1}{N}\sum_i X_i'y_i = (X'X)^{-1}X'y$$

```
frause oaxaca, clear
keep if lnwage !=.
mata:
  y = st_data(.,"lnwage")
  n = rows(y)
  x = st_data(.,"female age educ"),J(n,1,1)
  exx = cross(x,x)/n
  exy = cross(x,y)/n
  b   = invsym(exx)*exy
  b
end
```

```
<IPython.core.display.HTML object>


(Excerpt from the Swiss Labor Market Survey 1998)
(213 observations deleted)

. mata:
------------------------------------------------- mata (type end to exit) -----
:   y = st_data(.,"lnwage")

:   n = rows(y)

:   x = st_data(.,"female age educ"),J(n,1,1)

:   exx = cross(x,x)/n

:   exy = cross(x,y)/n

:   b = invsym(exx)*exy

:   b
                1
    +---------------+
  1 |  -.145393595  |
  2 |  .0161424301  |
  3 |  .0719321873  |
  4 |  1.970020725  |
    +---------------+

: end
--------------------------------------------------------------------------------

.
```

Inference - Distribution of $\beta's$

Given the model and OLS estimator:

$$y = X\beta + \varepsilon$$
$$\hat{\beta} = (X'X)^{-1}X'y$$

If we substitute y in the second equation, we get:

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

Finally:

$$\sqrt{N}(\hat{\beta} - \beta) = \sqrt{N}\Big[\frac{1}{N}\sum(X_i X_i')\Big]^{-1}\frac{1}{N}\sum(X_i\varepsilon_i)$$

- Here ε is the true population error. $\hat{\beta}$ is unbiased if the second term has an expectation of Zero. (the error is independent from X).
- The first term is assumed fixed $E(X_i X_i')$. And, because $E(X_i\varepsilon) = 0$, and $\frac{1}{\sqrt{N}}\sum(X_i\varepsilon)$ is normalized, by CLT we have that:

$$\sqrt{N}(\hat{\beta} - \beta) \sim N(0, E(X_i X_i')^{-1}\ E(X_i X_i'\varepsilon_i^2)\ E(X_i X_i')^{-1})$$

- From here, the main question is : How do we estimate $E(X_i X'\varepsilon_i^2)$?

Inference: Estimating SE

- Lets First Rewrite the last expression:

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

where:

$$\Omega = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1N} \\ \sigma_{21} & \sigma_2^2 & \dots & \sigma_{2N} \\ \dots & \dots & \dots & \dots \\ \sigma_{N1} & \sigma_{N2} & \dots & \sigma_{NN}^2 \end{pmatrix}$$

In other words, the variance of $\hat{\beta}$ allows for arbitrary relationship among the errors, as well as heteroskedasticity. This, however is impossible to estimate!, thus we require assumptions

Homoskedasticity and independent samples

With homoskedastic errors $\sigma^2 = \sigma_i^2 \ \forall i \in 1, ..., N$.

With independent samples $\sigma_{ij} = 0 \ \forall \ i \neq j$.

$$\Omega_0 0 = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & ... & \sigma_{1N} \\ \sigma_{21} & \sigma_2^2 & ... & \sigma_{2N} \\ ... & ... & ... & ... \\ \sigma_{N1} & \sigma_{N2} & ... & \sigma_{NN}^2 \end{pmatrix} = I(N) * \sigma^2$$

Thus

$$Var(\hat{\beta})_{00} = (X'X)^{-1}X'I(N)\sigma^2X(X'X)^{-1} = \sigma^2(X'X)^{-1}$$
$$\sigma^2 = E(\varepsilon^2)$$

```
mata: e=err = y:-x*b
mata: var_b_000 = mean(err:^2) * invsym(x'x)
mata: b,sqrt(diagonal(var_b_000))
```

```
                1              2
    +------------------------------+
  1 |  -.145393595   .0243547399   |
  2 |  .0161424301   .0010962465   |
  3 |  .0719321873    .005029506   |
  4 |  1.970020725   .0724744138   |
    +------------------------------+
```

But, σ^2 is not known, so we have to use $\hat{\sigma}^2$ instead, which depends on the sample residuals:

$$\hat{\sigma}^2 = \frac{1}{N-k-1}\sum \hat{e}^2$$

Where we account for the fact true errors are not observed, but rather residuals are estimated, adjusting the degrees of freedom.

```
mata:
    N = rows(y); k = cols(x)
    var_b_00 = sum(err:^2)/(N-k) * invsym(x'x)
    b,sqrt(diagonal(var_b_00))
end
```

```
. mata:
------------------------------------------------- mata (type end to exit) -----
:     N = rows(y); k = cols(x)

:     var_b_00 = sum(err:^2)/(N-k) * invsym(x'x)

:     b,sqrt(diagonal(var_b_00))
                  1             2
    +-----------------------------+
  1 |  -.145393595   .0243887787  |
  2 |  .0161424301   .0010977786  |
  3 |  .0719321873   .0050365354  |
  4 |  1.970020725   .0725757058  |
    +-----------------------------+

: end
--------------------------------------------------------------------------------

.
```

Lifting Assumptions: Heteroscedasticity

- We start by lifting this assumption, which implies the following:

$$\sigma_i^2 \neq \sigma_j^2 \ \forall \ i \neq j$$

But to estimate this, we need an approximation for $\sigma_i^2 = E(\varepsilon_i^2) = \varepsilon_i^2$.

- With this, we can obtain what is known as th White or Eicker-White or Heteroskedasiticy Robust Standard errors.

$$\begin{aligned} Var(\hat{\beta})_0 &= (X'X)^{-1}(X \cdot \hat{e})'(X \cdot \hat{e})(X'X)^{-1} \\ &= (X'X)^{-1} \sum (X_i X_i' \hat{e}^2)(X'X)^{-1} \end{aligned}$$

Which imposes **NO** penalty to the fact that we are using residuals not errors. If we account for that however, we obtain what is known as HC1, SE, the standard in `Stata`. (when you type `robust`)

$$Var(\hat{\beta})_1 = \frac{N}{N-K-1} Var(\hat{\beta})_0$$

```
mata:
    ixx = invsym(x'x)
    var_b_0 = ixx * (x:*e)'(x:*e) * ixx
    var_b_1 = N/(N-k)*var_b_0
    b,sqrt(diagonal(var_b_0)),sqrt(diagonal(var_b_1))
end
```

```
. mata:
------------------------------------------------- mata (type end to exit) -----
:     ixx = invsym(x'x)

:     var_b_0 = ixx * (x:*e)'(x:*e) * ixx

:     var_b_1 = N/(N-k)*var_b_0

:     b,sqrt(diagonal(var_b_0)),sqrt(diagonal(var_b_1))
                  1             2             3
    +-------------------------------------------+
  1 |  -.145393595   .0243162137   .0243501986  |
  2 |  .0161424301   .0013544849   .0013563779  |
  3 |  .0719321873    .005690214   .0056981668  |
  4 |  1.970020725   .0875757052   .0876981032  |
    +-------------------------------------------+

: end
--------------------------------------------------------------------------------

.
```

But error is not the same as residual!

A residual is model dependent, and should not be confused with the model error $\hat{\varepsilon} \neq \varepsilon$. Because of this, additional corrections are needed to obtained unbiased $var(\hat{\beta})$ estimates. (Degrees of freedom). But other options exists.

Redefine the Variance Formula:

$$Var(\hat{\beta}) = (X'X)^{-1}(\sum X_i X_i \psi_i)(X'X)^{-1}$$

From here Mackinnon and White (1985) suggest few other options:

$$HC0 : \psi_i = \hat{e}^2 \qquad HC1 : \psi_i = \frac{N}{N-K}\hat{e}^2$$
$$HC2 : \psi_i = \hat{e}^2 \frac{1}{1-h_{ii}} \qquad HC3 : \psi_i = \hat{e}^2 \frac{1}{(1-h_{ii})^2}$$

Where h_{ii} is the ith diagonal element of $X(X'X)^{-1}X'$ and allows you to see how dependent a model is to a single observation.

HC2 and HC3 Standard errors are better than HC1 SE, specially when Samples are small.

> NOTE: this h_{ii} element is also used to measure the degrees of freedom of a model. Sum it up, and you will see!.

Coding Robust SE

```
mata:
    // h = diagonal(X invsym(X'x) X') Wrong Way, too many calculations
    h = rowsum(x*invsym(x'x):*x)
    psi0 = e:^2           ;   psi1 = e:^2*N/(N-k)
    psi2 = e:^2:/(1:-h)   ;   psi3 = e:^2:/((1:-h):^2)
    var_b_0 = ixx * cross(x,psi0,x) * ixx
    var_b_1 = ixx * cross(x,psi1,x) * ixx
    var_b_2 = ixx * cross(x,psi2,x) * ixx
    var_b_3 = ixx * cross(x,psi3,x) * ixx
    b,sqrt(diagonal(var_b_0)),sqrt(diagonal(var_b_1)),
    sqrt(diagonal(var_b_2)),sqrt(diagonal(var_b_3))
end
```

```
. mata:
------------------------------------------------- mata (type end to exit) -----
:     h = rowsum(x*invsym(x'x):*x)

:     psi0 = e:^2 ; psi1 = e:^2*N/(N-k)

:     psi2 = e:^2:/(1:-h) ; psi3 = e:^2:/((1:-h):^2)

:     var_b_0 = ixx * cross(x,psi0,x) * ixx

:     var_b_1 = ixx * cross(x,psi1,x) * ixx
```

```
:     var_b_2 = ixx * cross(x,psi2,x) * ixx

:     var_b_3 = ixx * cross(x,psi3,x) * ixx

:     b,sqrt(diagonal(var_b_0)),sqrt(diagonal(var_b_1)),
>     sqrt(diagonal(var_b_2)),sqrt(diagonal(var_b_3))
                 1             2             3             4             5
    +------------------------------------------------------------------------+
  1 |  -.145393595   .0243162137   .0243501986   .0243568124   .0243975204  |
  2 |  .0161424301   .0013544849   .0013563779   .0013573922   .0013603079  |
  3 |  .0719321873    .005690214   .0056981668   .0057079191    .005725691  |
  4 |  1.970020725   .0875757052   .0876981032   .0878131672   .0880514838  |
    +------------------------------------------------------------------------+

: end
------------------------------------------------------------------------------


.
```

Or in Stata:

```
regress y x1 x2 x3, vce(robust)
regress y x1 x2 x3, vce(hc2)
regress y x1 x2 x3, vce(hc3)
```

GLS and Weighted Least Squares

- GLS is a generalization of OLS, that could be used to address heteroskedasticity.
- There are two ways to do this:
    1. Transform/weight the data to make it homoskedastic (WLS)
    2. Modify the variance covariance matrix of the errors (GLS)
- Call $\hat{h}(x)$ the predicted error variance. The GLS estimator for $V_{gls}(\beta)$ is given by:

$$V_{gls}(\beta) = (X'X)^{-1} \sum (X_i X_i' \hat{h}(x))(X'X)^{-1}$$

- That way, Heteroskedasticity is addressed, but without changing the model estimates $\beta's$

Lifting Even more Assumptions: Correlation

- One assumption we barely consider last semester was the possibility that errors could be correlated across observations. (except for time series and serial correlation)
- For example, families may share similar unobserved factors, So would people interviewed from the same classroom, cohort, city, etc. There could be many dimensions to consider possible correlations!
- In that situation, we may be missmeasuring the magnitude of the errors (probably downward), because the Ω is no longer diagonal: $\sigma_{ij} \neq 0$ for some $i \neq j$.
  - But, estimate all parameters in an NxN matrix is unfeasible. We need assumptions!

New Assumptions

- Say we have G groups $g = (1...G)$. We can rewrite the expression for $\hat{\beta}$ as follows:

$$\begin{aligned}\hat{\beta} - \beta &= (X'X)^{-1} \sum_{g=1}^{G} X_g' \varepsilon_g \\ &= (X'X)^{-1} \sum_{g=1}^{G} s_g\end{aligned}$$

- We can assume that individuals are correlated within groups $E(s_g' s_g) = \Sigma_g$, but they are uncorrelated across groups $E(s_g s_g') = 0 \ \forall \ g \neq g'$.
- These groups are typically known as "**clusters**"

Addressing Correlation

- The idea of correcting for clusters is pretty simple. We just need to come up with an estimator for Σ_g for every cluster, so that:

$$\begin{aligned}Var(\hat{\beta}) &= (X'X)^{-1} \left(\sum_{g=1}^{N} \Sigma_g \right) (X'X)^{-1} \\ \Sigma_g &= E(X_g' \Omega_g X_g)\end{aligned}$$

- Here Ω_g should be an approximation of the variance covariance matrix among the errors of ALL individuals that belong to the same cluster. But how do we approximate it?
- As with the EW - HC standard errors, there are many ways to estimate Clustered Standard errors. See MacKinnon et al (2023) for reference. We will refer only to the simpler ones CV0 and CV1.

- Recall we approximate σ_i^2 with ε_i^2. Then we can approximate σ_{ij} with $\varepsilon_j \varepsilon_i$. More specifically:

$$\Omega_g \simeq \varepsilon\varepsilon' \text{ or } \Sigma_g = X_g'\varepsilon\varepsilon' X_g = (X_g'\varepsilon)(\varepsilon' X_g)$$

- Change ε with $\hat{\varepsilon}$, do that for every group, and done! (almost).

- As mentioned earlier, there are many CCSE (clustered consistent SE).

$$CV_0 = (X'X)^{-1} \sum_{g=1}^{G} \hat{\Sigma}_g (X'X)^{-1}$$

$$CV_1 = \frac{G(N-1)}{(G-1)(N-k-1)} (X'X)^{-1} \sum_{g=1}^{G} \hat{\Sigma}_g (X'X)^{-1}$$

- Similar to HC. CV0 does not correct for degrees of freedom. CV1, however, accounts for Degrees of freedom in the model, and clusters.

```
sort isco
mata:
    // 1st Sort Data (easier in Stata rather than Mata) and reload
    y   = st_data(.,"lnwage")
    x   = st_data(.,"educ exper female"),J(1434,1,1)
    cvar= st_data(.,"isco")
    ixx = invsym(cross(x,x)); xy = cross(x,y)
    b   = ixx * xy
    e   = y:-x*b
    // Set the panel info
    info = panelsetup(cvar,1); g=rows(info); n=rows(y)
    // get X_g'e for all groups:
```

```
    s_xg_e = panelsum(x:*e,info)
    // Sum Sigma_g
    sigma_g = s_xg_e's_xg_e
    cv0 = ixx*sigma_g*ixx
    cv1 =g/(g-1)*(n-1)/(n-k)*ixx*sigma_g*ixx
    b,sqrt(diagonal(cv0)),sqrt(diagonal(cv1))
end
```

```
. mata:
------------------------------------------------- mata (type end to exit) -----
:     y = st_data(.,"lnwage")

:     x = st_data(.,"educ exper female"),J(1434,1,1)

:     cvar= st_data(.,"isco")

:     ixx = invsym(cross(x,x)); xy = cross(x,y)

:     b = ixx * xy

:     e = y:-x*b

:     info = panelsetup(cvar,1); g=rows(info); n=rows(y)

:     s_xg_e = panelsum(x:*e,info)

:     sigma_g = s_xg_e's_xg_e

:     cv0 = ixx*sigma_g*ixx

:     cv1 =g/(g-1)*(n-1)/(n-k)*ixx*sigma_g*ixx

:     b,sqrt(diagonal(cv0)),sqrt(diagonal(cv1))
                  1              2              3
    +-----------------------------------------------+
  1 |   .0858251775    .0140570765    .0149254126  |
  2 |   .0147342796    .0014534593    .0015432426  |
  3 |  -.0949227416    .0525121234    .0557559112  |
  4 |   2.218849962    .1947497649    .2067798804  |
    +-----------------------------------------------+
```

```
: end
------------------------------------------------------------------------------
```

.

or compare it to

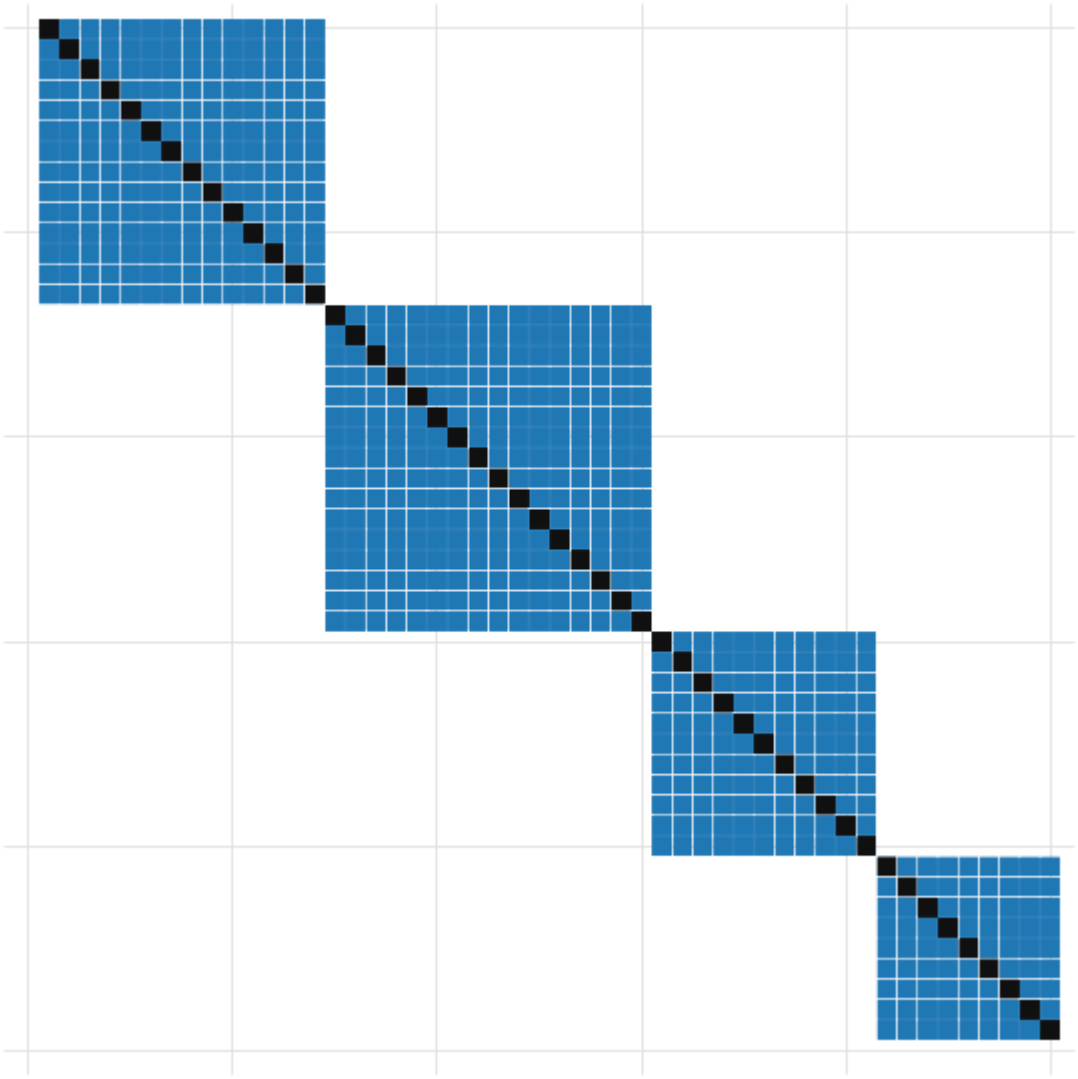
```
reg lnwage educ exper female, cluster(isco)
```

```
Linear regression                               Number of obs     =      1,434
                                                F(3, 8)           =      59.13
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2217
                                                Root MSE          =     .46897

                                  (Std. err. adjusted for 9 clusters in isco)
------------------------------------------------------------------------------
             |               Robust
      lnwage | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
        educ |   .0858252   .0149254     5.75   0.000     .0514071    .1202432
       exper |   .0147343   .0015432     9.55   0.000     .0111756     .018293
      female |  -.0949227   .0557559    -1.70   0.127    -.2234961    .0336506
       _cons |    2.21885   .2067799    10.73   0.000     1.742015    2.695685
------------------------------------------------------------------------------
```

Visualizing the difference

```
clear
set scheme white2
color_style tableau
set seed 1
set obs 50
gen r1=runiformint(1,4)
gen r2=runiformint(1,4)
gen id=_n
sort r1  r2
qui:mata:
r1=st_data(.,"r1")
```

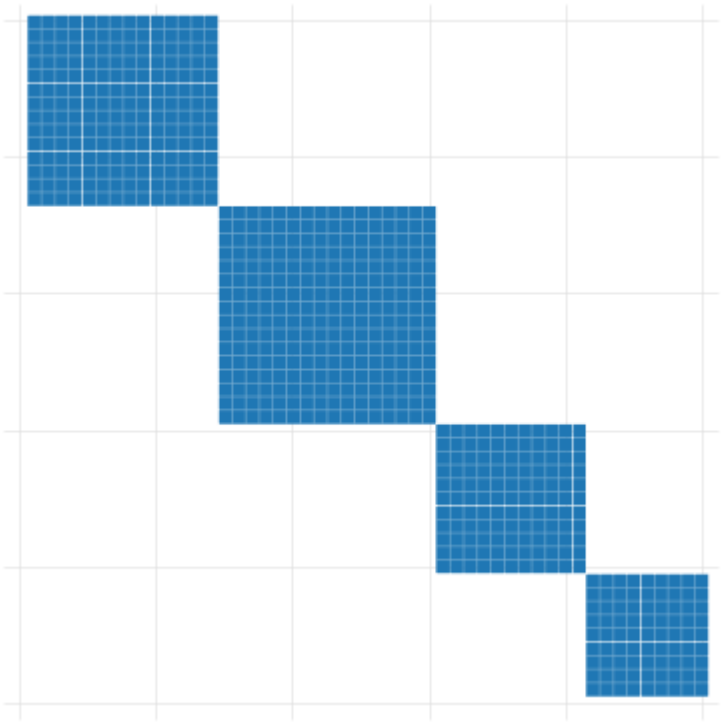
```
r2=st_data(.,"r2")
rr1=J(rows(r1)*rows(r2),4,0)
k=0
for(i=1;i<=50;i++){
    for(j=1;j<=50;j++){
        if ((r1[i]==r1[j]) | (r2[i]==r2[j])) {
            k++
            rr1[k,]=(51-i,j,(r1[i]==r1[j]),(r2[i]==r2[j]) )
        }
    }
}
rr1=rr1[1..k,]
end
getmata rr1*=rr1, replace force

two (scatter rr11 rr12 if rr13==1,  ms(s) msize(2.1))  ///
    (scatter rr11 rr12 if 51-rr11 == rr12, ms(s) msize(2.1) color(gs1)  ) ///
    , aspect(1) legend(off)  xtitle("") ytitle("") yscale(off) xscale(off) xsize(6) ysize(6)
```



Visualizing Multi-way Clustering

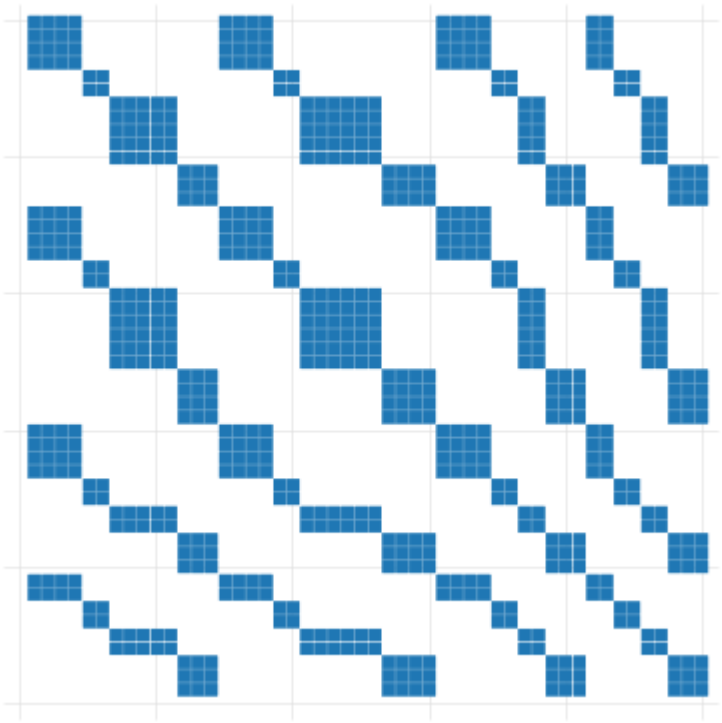
First Cluster

```
two (scatter rr11 rr12 if rr13==1,  ms(s) msize(2.1))  ///
    , aspect(1) legend(off)  xtitle("") ytitle("") yscale(off) xscale(off) name(m1, replace)
```



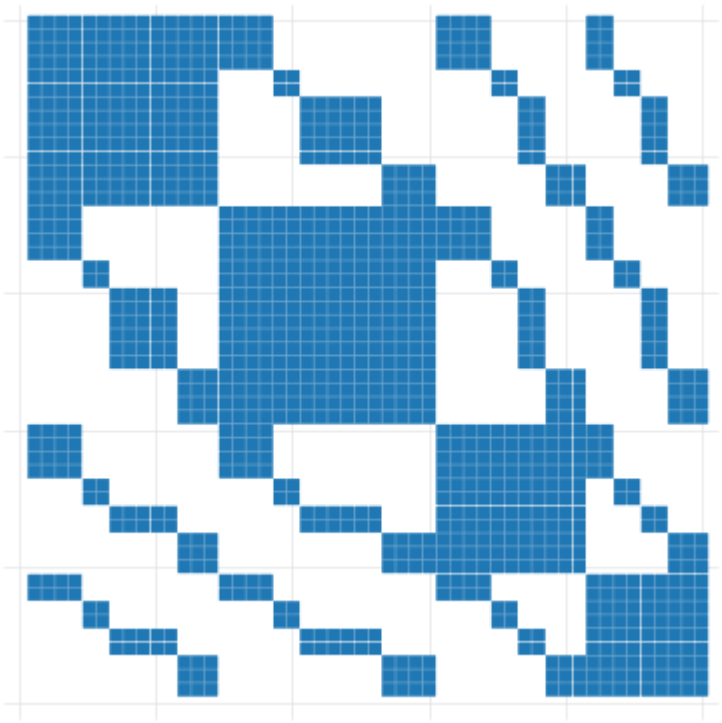
Second Cluster

```
two (scatter rr11 rr12 if rr14==1,  ms(s) msize(2.1))  ///
    , aspect(1) legend(off)  xtitle("") ytitle("") yscale(off) xscale(off) name(m2, replace)
```



Combining Clusters

```
two (scatter rr11 rr12 if rr14==1 | rr13==1,  ms(s) msize(2.1))  ///
    , aspect(1) legend(off)  xtitle("") ytitle("") yscale(off) xscale(off) name(m3, replace)
```



Beware of over-clustering

While clustering helps address a problem of "intragroup" correlation, it can/should be done with care. It is important to be aware about some unintended problems of over-clustering.

1. CV0 and CV1 work well when you have a large number of Clusters. How many? MHE(2009) says…42 (this is like having large enough samples for Asymptotic variance). If # clusters are small, you would do better with other approaches (including CV2 and CV3).
2. When you cluster your standard errors, you will "most-likely" generate larger standard errors in your model. Standard recommendation (MHE) is to cluster at the level that makes sense (based on data) and produces largest SE (to be conservative).

Role of clusters

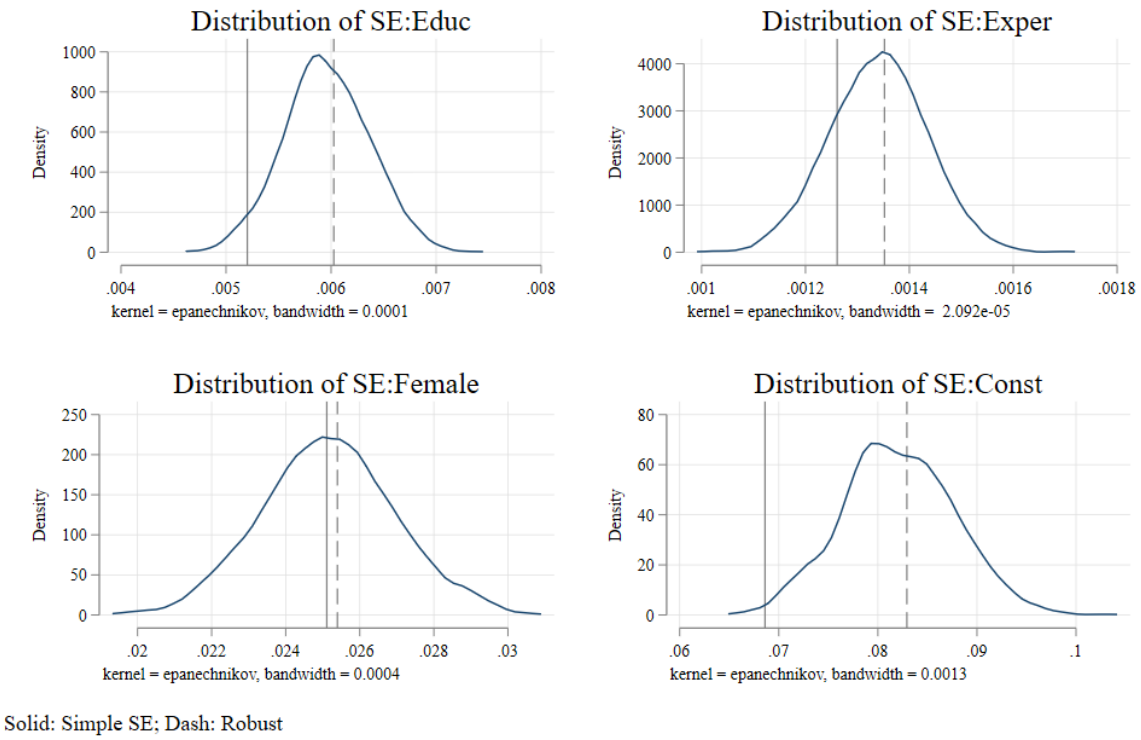


Figure 1: Standard Errors

3. You may also consider that clustering does not work well when sample sizes within cluster are to diverse (micro vs macro clusters)
4. And there is the case where clustering is required among multiple dimensions (see `vcemway`). Where the unobserved correlation could be present in different dimensions.

So what to cluster and how?

- Mackinnon et al (2023) provides a guide on how and when to cluster your standard errors. (some are quite advanced)

- General practice, At least use Robust SE (HC2 or HC3 if sample is small), but use clustered SE for robustness.
- You may want to cluster SE based on some theoretical expectations. Choose -broader-groups for conservative analysis.
- In treatment-causal effect analysis, you may want to cluster at the "treatment" level.

  But…Beyond hc0/1 and CV0/1 there is not much out there for correcting Standard errors in nonlinear models.

The Bootstrap

If you can't Sandwich , you can re-Sample

- The discussion above refered to the estimation of SE using $Math$. In other words, it was based on the asymptotic properties of the data. Which may not work in small samples.
- An alternative, often used by practitioners, is using re-sampling methods to obtain approximations to the coefficient distributions of interest.

But… How does it work?

First ask yourself, how does Asymptotic theory work (and econometrics)?

> NOTE: I RECOMMEND READING THE -SIMULATION- CHAPTER IN THE EFFECT, AND SIMULATION METHODS CHAPTER IN CT.

A Brief Review…again

If I were to summarize most of the methodologies (ok all) we used last semester, and this one, the properties that have been derived and proofed are based on the assumption that we "could" always get more data (frequentist approach).

There is population (or super population) from where we can get samples of data (and never repeat data).

1. We get a sample (y, X) (of size N)
2. Estimate our model : $\texttt{method}(y, X) \rightarrow \beta's$
3. Repeat to infinitum
4. Collect all $\beta's$ and summarize. (Mean and Standard deviations)

Done.

The distributions you get from the above exercise should be the same as what your estimation method produces. (in average) (if not, there there is something wrong with the estimation method)

But we only get 1 Sample!

The truth is we do not have access to multiple samples. Getting more data, is in fact, very expensive. So what to do ?

- Rely on Asymptotic theory
- learn Bayesian Econometrics
- or-resample? and do Bootstrap!

Basic idea of Bootstrapping

- In the ideal scenario, you get multiple samples from your population, Estimate parameters, and done.
- If not possible you do the next best thing. You get your sample (assume is your mini-population),
  - Draw subsamples of same size (with replacement) (y_i^s, X_i^s)
  - estimate your model and obtain parameters β_i^s
  - Summarize those parameters...and done, you get $Var(\hat{\beta})$ for . (or is it?)

Bootstrapping

- Bootstrapping is a methodology that allows you to obtain empirical estimations of standard errors making use of the data in hand, and without even knowing about Asymptotic theory (other than how to get means and variances).

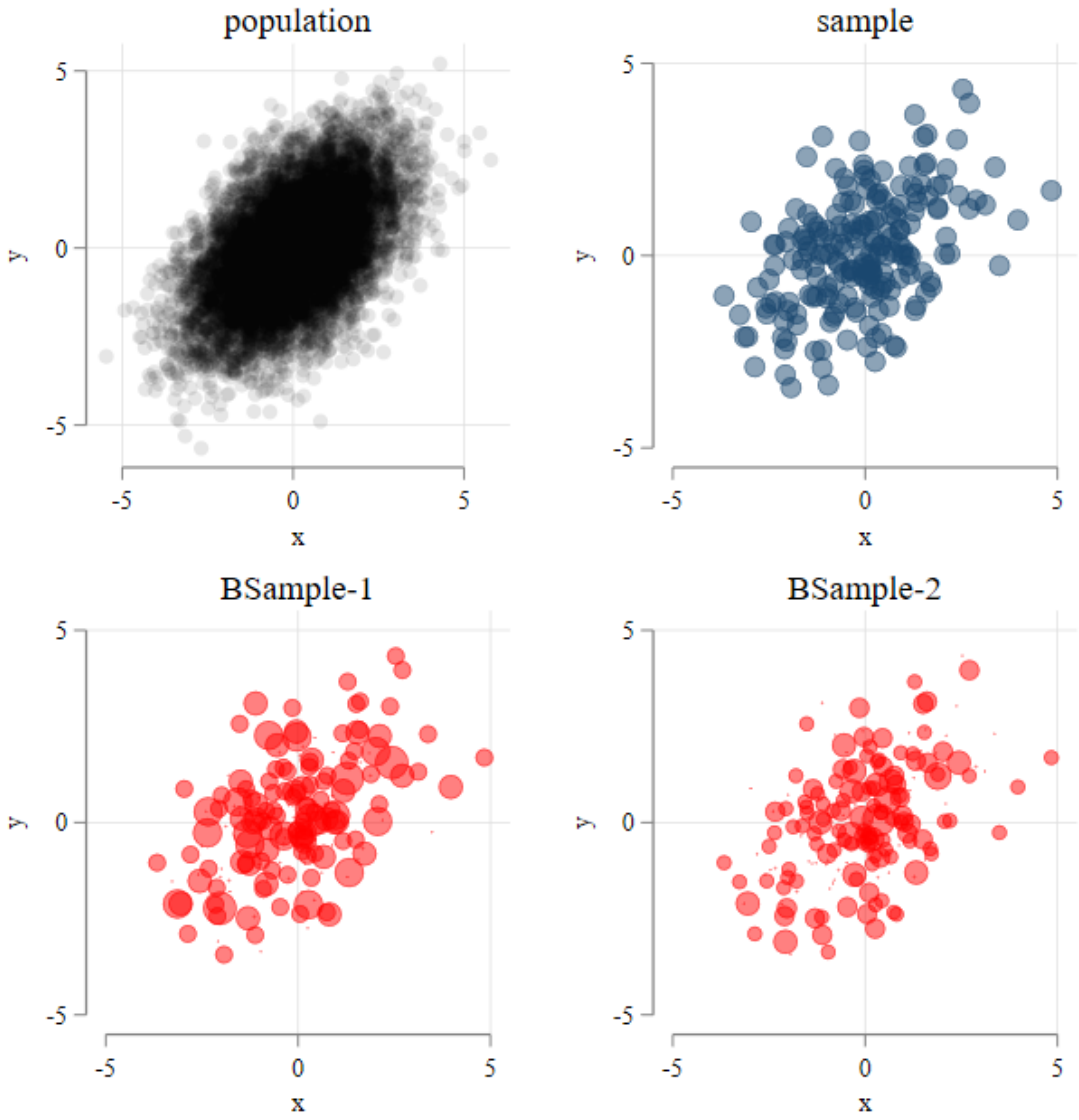


Figure 2: Bootstrap Sample

- And of course, it comes in different flavors.

Standard Bootstrap:

- **Non-parametric Bootstrap**: You draw subsamples from the main sample. Each observation has the same pr of being selected.
  - Easiest to implement (`see bootstrap:`)
  - Works in almost all cases, but you may have situations when some covariates are rare.
  - Can allow for "clusters" using "block bootstrapping".

Standard Bootstrap:

- **Parametric Bootstrap:** You estimate your model, make assumptions of your model error.
  - You need to implement it on your own. $y^s = x\hat{b} + \tilde{e}$ for $\tilde{e} \sim f(\hat{\theta})$
  - It will not work well if the assumptions of the error modeling are wrong.

Standard Bootstrap:

- **Residual bootstrap:** Estimate your model, obtain residuals. Re-sample residuals
  - Again, implement it on your own. $y^s = x\hat{b} + \tilde{e}$ for $\tilde{e} \sim \hat{e}_1, ..., \hat{e}_N$
  - It depends even more on the assumptions of the error modeling.

Wild Bootstrap

Then there are the more advanced (but faster) Bootstrap methods: WildBootstrap

- **UWild bootstrap**: Estimate your model, obtain residuals, and re-sample residual weights.
  - Again...on your own: $y^s = x\hat{b} + \hat{e} * v$, where $v \sim ff()$ where $ff()$ is a "good" distribution function. $E(v) = 0$ & $Var(v) = 1$
  - Re-estimate the model and obtain $\hat{\beta}'s$. Repeat and summarize.
  - Actually quite flexible, and works well under heteroskedasticity!
  - It can also allow clustered standard errors. The error v no longer changes by individual, but by group. It also works well with weights.

Wild Bootstrap:

- **UWild bootstrap-2** : Estimate your model, obtain Influence functions , and re-sample residual weights.
  - This is an extension to the previous option. But with advantages
    - you do not need to *re-estimate* the model. Just look into how the the mean of IF's change.
    - it can be applied to linear and nonlinear model (if you know how to build the IF's)
  - Works well with clustered and weights.

Wild Bootstrap:

- **CWild bootstrap:** Similar UWild Bootstrap, Obtain Influence functions under the Null (imposing restrictions), and use that to test the NULL.
  - No, you do not need to do it on your own. `see bootest` in `Stata`.
  - Works pretty well with small samples and small # clusters. Probably the way to go if you really care about Standard errors.

How to Bootstrap? in `Stata`

I have a few notes on Bootstrapping here Bootstrapping in Stata. But let me give you the highlights for the most general case.

1. Most (if not all commands) in `Stata` allow you to obtain bootstrap standard errors, by default. see:`help [cmd]`

   they usually have the following syntax:

   ```
   [cmd] y x1 x2 x3, vce(bootstrap, options)
   regress lnwage educ exper female, vce(bootstrap, reps(100))
   ```

2. However, you can also Bootstrap that commands that do not have their own `bootstrap` option.

   ```
   bootstrap:[cmd] y x1 x2 x3,
   bootstrap, reps(100):regress lnwage educ exper female
   bootstrap, reps(100) cluster(isco):regress lnwage educ exper female
   ```

3. This last command may allow you to bootstrap multiple models at the same time, although it does require a bit of programming. (and a do file)

```
frause oaxaca, clear
gen tchild = kids6 + kids714
capture program drop bs_wages_children
program bs_wages_children, eclass // eclass is for things like equations
    ** Estimate first model
    reg lnwage educ exper female
    matrix b1 = e(b)
    matrix coleq b1 = lnwage
    ** Estimate second model
    reg tchild educ exper female
    matrix b2 = e(b)
    matrix coleq b2 = tchild
    ** Put things together and post
    matrix b = b1 , b2
    ereturn post b
end
bootstrap: bs_wages_children
```

```
(Excerpt from the Swiss Labor Market Survey 1998)

(running bs_wages_children on estimation sample)

warning: bs_wages_children does not set e(sample), so no observations will be
         excluded from the resampling because of missing values or other
         reasons. To exclude observations, press Break, save the data, drop
         any observations that are to be excluded, and rerun bootstrap.

Bootstrap replications (50): .........10.........20.........30.........40......
> ...50 done

Bootstrap results                                       Number of obs = 1,647
                                                        Replications  =    50

------------------------------------------------------------------------------
             |   Observed   Bootstrap                         Normal-based
             | coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
lnwage       |
        educ |   .0858252   .0062606    13.71   0.000     .0735547    .0980957
       exper |   .0147343    .001283    11.48   0.000     .0122196     .017249
      female |  -.0949227   .0305222    -3.11   0.002    -.1547452   -.0351003
       _cons |    2.21885   .0855954    25.92   0.000     2.051086    2.386614
-------------+----------------------------------------------------------------
tchild       |
        educ |   .0177854   .0087606     2.03   0.042      .000615    .0349558
       exper |  -.0047747   .0017462    -2.73   0.006    -.0081972   -.0013522
      female |  -.1306332   .0395711    -3.30   0.001    -.2081911   -.0530753
       _cons |   .4163459   .1201824     3.46   0.001     .1807927    .6518991
------------------------------------------------------------------------------
```

Why does it matter? because you may want to test coefficients individually, or across models. This is only possible if the FULL system is estimated jointly

What about Wild Bootstrap?

- Wildbootstrap is available using `boottest` (`ssc install bootest`)
- And in `Stata18+`, you have `wildbootstrap` (although is meant for clustered SE)

```
frause oaxaca, clear
regress lnwage educ exper female, robust
boottest educ, nograph
boottest exper, nograph
boottest female, nograph
```

(Excerpt from the Swiss Labor Market Survey 1998)

```
Linear regression                               Number of obs     =      1,434
                                                F(3, 1430)        =      97.11
                                                Prob > F          =     0.0000
                                                R-squared         =     0.2217
                                                Root MSE          =     .46897

------------------------------------------------------------------------------
             |               Robust
      lnwage | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
        educ |   .0858252   .0060342    14.22   0.000     .0739883     .097662
       exper |   .0147343    .001354    10.88   0.000     .0120783    .0173903
      female |  -.0949227   .0254309    -3.73   0.000    -.1448086   -.0450369
       _cons |    2.21885   .0830438    26.72   0.000     2.055949    2.381751
------------------------------------------------------------------------------

Wild bootstrap-t, null imposed, 999 replications, Wald test, Rademacher weights
> :
  educ

                        t(1430) =    14.2231
                       Prob>|t| =     0.0000

95% confidence set for null hypothesis expression: [.07399, .09766]

Wild bootstrap-t, null imposed, 999 replications, Wald test, Rademacher weights
> :
  exper

                        t(1430) =    10.8822
                       Prob>|t| =     0.0000

95% confidence set for null hypothesis expression: [.01208, .01739]
```

```
Wild bootstrap-t, null imposed, 999 replications, Wald test, Rademacher weights
> :
  female

                          t(1430) =    -3.7326
                         Prob>|t| =     0.0000

95% confidence set for null hypothesis expression: [-.1433, -.04703]
```

Final words on Bootstrap:

So bootstrap (and its many flavors) are convenient approaches to estimate standard errors and elaborate statistical Inference, but its not infallible.

1. If the re-sampling process does not simulate the true sampling design, we may miss important information when constructing SE.
2. When the parameters are estimated using "hard" cutoffs or restricted distributions, it may not produce good approximations for SE.
3. You usually require MANY repetitions (standard = 50, but you probably want 999 or more). The more the better, but has some computational costs. (specially simple bs)
4. Some methods play better with weighted samples, clusters, and other survey designs than others. And some require more know-how than others.

So choose your weapon wisely!

Small Diversion : The Delta Method

Variance of nonlinear functions

- Some times (perhaps not with simple OLS) you many need to estimate Standard errors for transformations of your main coefficient of interest, or combinations of those coefficients.
- Say that you estimated $\theta \sim N(\mu_\theta, \sigma^2_\theta)$ but are interested in the distribution of $g(\theta)$. How do you do this?
- Two options:

  a) you re estimate $g(\theta)$ instead, or
  b) you make an approximation, using the **Delta Method**

- How does it work?

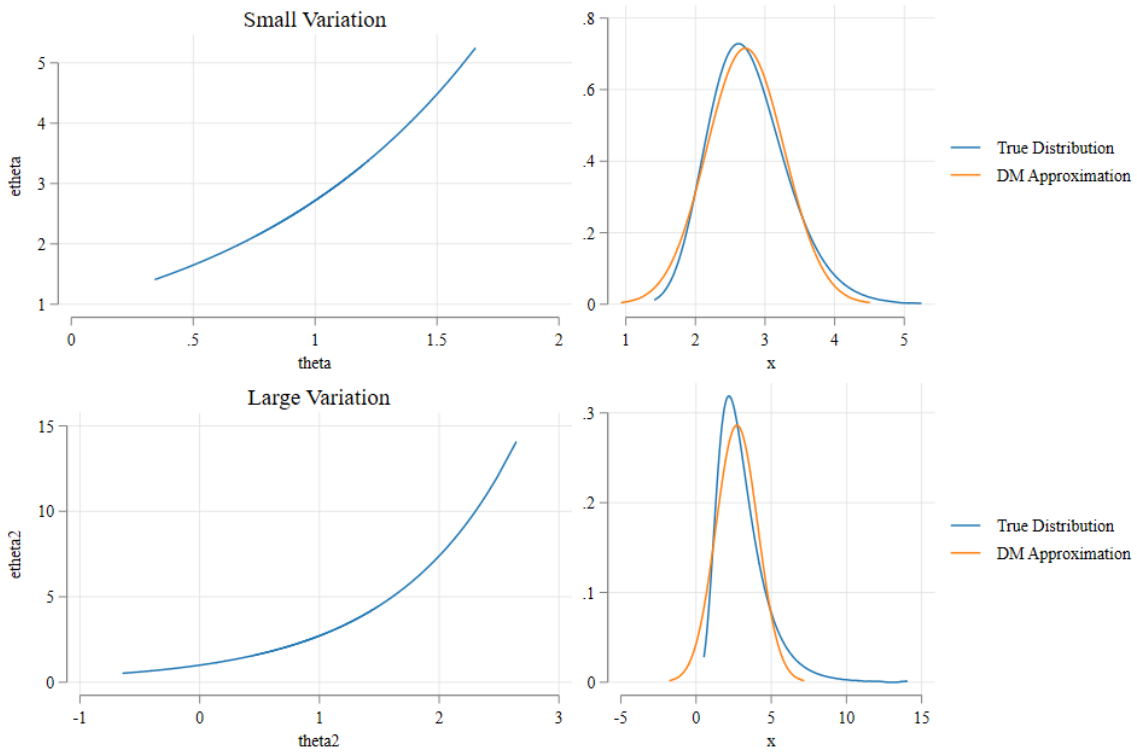
- The **Delta method** uses the linear approximations to *approximate* the otherwise not known distributions.
- Further, It relies on the fact that linear transformations a normal distribution, is on itself normal. For example:

$$g(\hat{\theta}) \simeq g(\theta) + g'(\hat{\theta})(\hat{\theta} - \theta)$$

- This states that the nonlinear function $g(\theta)$ can be "locally" approximated as a linear function in the neighborhood of $g(\theta)$.
- Predictions above or below are approximated using the slope of the function. $g'(\theta)$.
- So, if we take the variance, we get:

$$Var(g(\hat{\theta})) \simeq Var\left(g(\theta) + g'(\hat{\theta})(\hat{\theta} - \theta)\right) = g'(\hat{\theta})^2 Var(\theta)$$

Delta Method: Visualization



It can go multivariate as well:

$$g(\hat{\theta}, \hat{\gamma}) - g(\theta, \gamma) \simeq N(0, \nabla g' \Sigma \nabla g)$$
$$\nabla g' = [dg/d\theta \quad dg/d\gamma]$$

Although you need to get the partial derivatives of $g(\theta, \gamma)$

Example

- Say that you obtain the mean standard error for averages wages for men and women, along with the correlation between the two.

- however, you are insterested in estimating the wage ratio, and its variance. How do you do this?

$$R = \frac{\mu_f}{\mu_m}$$

- Need to obtain the Gradients g

$$g = \begin{bmatrix} \frac{\partial R}{\partial \mu_f} \\ \frac{\partial R}{\partial \mu_m} \end{bmatrix} = \begin{bmatrix} \frac{1}{\mu_m} \\ -\frac{\mu_f}{\mu_m^2} \end{bmatrix}$$

Then the variance of R is:

$$Var(R) = g' \Sigma_\mu g$$

Example in `Stata`

```
frause oaxaca, clear
gen wage = exp(lnwage)
mean wage, over(female)
mata:
  mu = st_matrix("e(b)")
  vcv = st_matrix("e(V)")
  dg = 1/mu[2] \ -mu[1]/mu[2]^2
  var_r = dg'*vcv*dg
  sqrt(var_r)
end
nlcom _b[ c.wage@0.female]/_b[ c.wage@1.female]
```

```
(Excerpt from the Swiss Labor Market Survey 1998)
(213 missing values generated)

Mean estimation                              Number of obs = 1,434

------------------------------------------------------------------
              |       Mean   Std. err.     [95% conf. interval]
--------------+---------------------------------------------------
```

```
c.wage@female |
           0  |   34.33619   .5175882      33.32088    35.35151
           1  |   30.25354   .6805642      28.91853    31.58855
------------------------------------------------------------------------------

. mata:
------------------------------------------------- mata (type end to exit) -----
:   mu = st_matrix("e(b)")

:   vcv = st_matrix("e(V)")

:   dg = 1/mu[2] \ -mu[1]/mu[2]^2

:   var_r = dg'*vcv*dg

:   sqrt(var_r)
  .0307332119

: end
------------------------------------------------------------------------------

.

       _nl_1: _b[ c.wage@0.female]/_b[ c.wage@1.female]

------------------------------------------------------------------------------
        Mean | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       _nl_1 |   1.134948   .0307332    36.93   0.000     1.074712    1.195184
------------------------------------------------------------------------------
```

So why do we care:

Two reasons:

- Nonlinear models need this kind of approximations to do statistical inference (probit/logit)
- Recall that when using Robust Standard errors Joint hypothesis Should be done with Care…

Consider a linear set of restrictions imposed by the $H_0 : R\beta = r$.

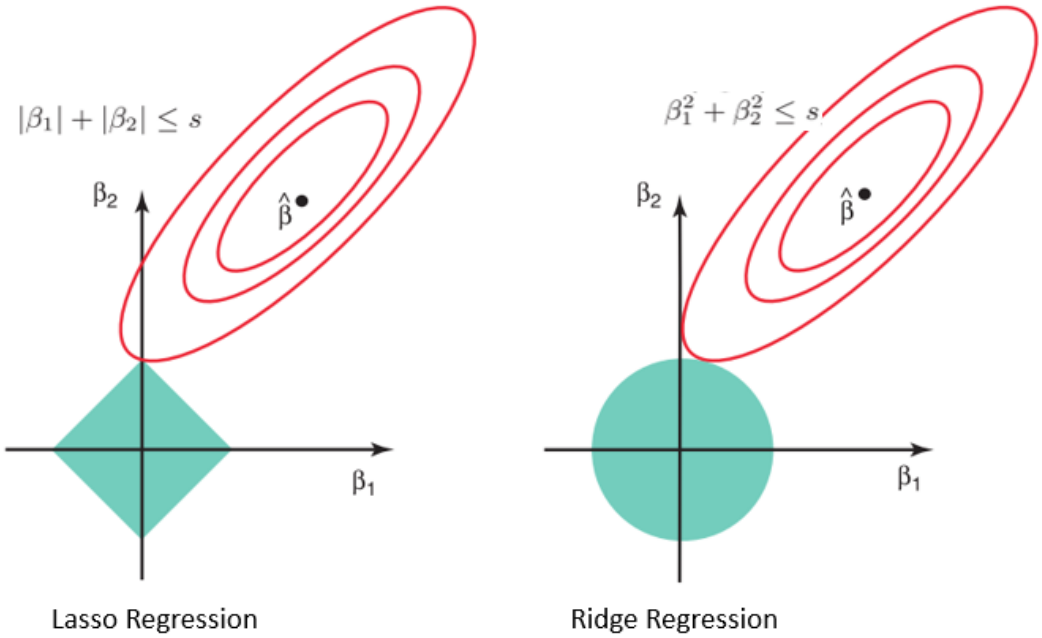
1. Estimate the Variance of $R\beta$

$$Var(R\beta) = \nabla(R\beta)' Var(\beta) R \nabla(R\beta)' = R' Var(\beta) R$$

2. Estimate the F value for the Linear Hypothesis (Wald Test)

$$(R\hat{\beta} - r)' Var(R\beta)^{-1} (R\hat{\beta} - r)/Q \sim F(Q, N - K)$$

Linear Model Selection and Regularization



What happens when K is too big?

- How many variables (max) can you use in a model?
  - $$max\ k = rank(X'X)$$
- What happens when you add too many variables in a model?

  - Increase Multicolinearity and coefficient variance (too much noise)
  - R2 overly large (without explaining much)
  - Far more difficult to interpret (too many factors)
  - May introduce endogeneity (when it wasnt a problem before)

- How can you solve the problem?
  - You select only a few of the variables, based on theory, and contribution to the model

- What if you can't choose?

ML: We let the Choose for you

> Before we start. The methodology we will discuss are usually meant to get models with "good" predictive power, and some times better interpretability, not so much stat-inference (although its possible)

When you do not know how to choose, you could try select a subset of variables from your model such that you maximize **out-of-sample** predictive power

This is typically achieved using the following:

$$AR^2 = 1 - \frac{SSR}{SST}\frac{n-1}{n-k-1} AIC = n^{-1}(SSR + 2k\hat{\sigma}^2) BIC = n^{-1}(SSR + ln(n)k\hat{\sigma}^2)$$

Or using a method known as cross-validation (Comparing predictive power using data not used for model estimation)

However, we can always try to estimate a model with all variables!

Ridge and Lasso and ElasticNet

- Recall that when using OLS to obtain $\beta's$, we try to minimize the following:

$$SSR = \sum_i (y_i - X_i\beta)^2$$

- This has the restrictions of mentioned before ($k < N$). In addition to letting coefficents vary "too much"

- An alternative is to Impose additional restrictions so that coefficients do not vary as much. This is known as **Regularization**.
-

Ridge Regression

- One such approach is **Ridge** regression, which minimizes the following:

$$rSS = \sum_i (y_i - X_i\beta)^2 + \lambda \sum_{k=1}^{K} \beta_k^2$$

- This essentially aims to find parameters that reduces SSR, but also "controls" for how large $\beta's$ can be, using a shrinkage penalty that depends on λ.
- If $\lambda = 0$ you get Standard OLS, and if $\lambda \to \infty$, you get a situation where all betas (but the constant) are zero. For intermediate values, you may have better models than OLS, because you can balance Bias (when $\beta's$ are zero) with increase variance (when all $\beta's$ vary as they "please")

- We usually start with Ridge, because is relatively Easy to implement, since it has a close form Solution:

$$\beta = (X'X + \lambda I)^{-1} X'y$$

```
set linesize 255
frause oaxaca, clear
keep if lnwage!=.
gen male = 1-female
mata:
    y = st_data(.,"lnwage")
    x = st_data(.,"educ exper female male"),J(1434,1,1)
    i0 = I(5);i0[5,5]=0
    xx = (cross(x,x)) ; xy = (cross(x,y))
    bb0 = invsym(xx)*xy
    bb1 = invsym(xx:+i0*1)*xy
    bb10 = invsym(xx:+i0*10)*xy
```

```
    bb100 = invsym(xx:+i0*100)*xy
    bb1000 = invsym(xx:+i0*1000)*xy
    bb0,bb1,bb10,bb100,bb1000
end
```

```
(Excerpt from the Swiss Labor Market Survey 1998)
(213 observations deleted)

. mata:
------------------------------------------------- mata (type end to exit) ------------------
:     y = st_data(.,"lnwage")

:     x = st_data(.,"educ exper female male"),J(1434,1,1)

:     i0 = I(5);i0[5,5]=0

:     xx = (cross(x,x)) ; xy = (cross(x,y))

:     bb0 = invsym(xx)*xy

:     bb1 = invsym(xx:+i0*1)*xy

:     bb10 = invsym(xx:+i0*10)*xy

:     bb100 = invsym(xx:+i0*100)*xy

:     bb1000 = invsym(xx:+i0*1000)*xy

:     bb0,bb1,bb10,bb100,bb1000
                  1              2              3              4              5
    +-------------------------------------------------------------------------------+
  1 |   .0858251775    .0858183338    .0857563567    .0851046501    .0778292498  |
  2 |   .0147342796    .0147345813    .0147372042    .0147554544    .0146298058  |
  3 |  -.0949227416    -.047396817   -.0468240416    -.041806663   -.0208062854  |
  4 |             0     .047396817    .0468240416     .041806663    .0208062854  |
  5 |   2.218849962    2.171466638    2.172174327    2.179690914    2.266275433  |
    +-------------------------------------------------------------------------------+

: end
--------------------------------------------------------------------------------------------

.
```

Lasso and Elastic Net

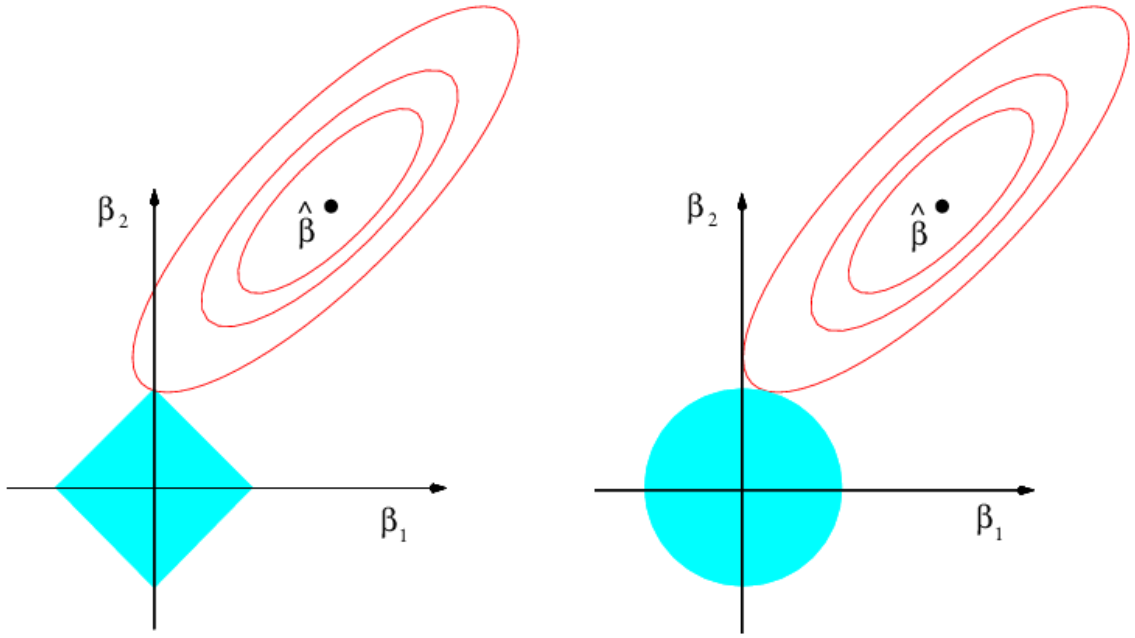
- Ridge is a relatively easy model to understand and estimate, since it has a close form solution. It has the slight disadvantage that you still estimate a coefficient for "every" variable (tho some are very small)
- Another approach, that overcomes this advantage is known as Lasso.

$$LSS = \sum_i (y_i - X_i\beta)^2 + \lambda \sum_{k=1}^{K} |\beta_k|$$

- and the one known as Elastic net

$$eSS = \sum_i (y_i - X_i\beta)^2 + \lambda_L \sum_{k=1}^{K} |\beta_k| + \lambda_r \sum_{k=1}^{K} \beta_k^2$$

Lasso vs Ridge



Considerations:

As with many methodologies, the benefits from this approaches is not free.

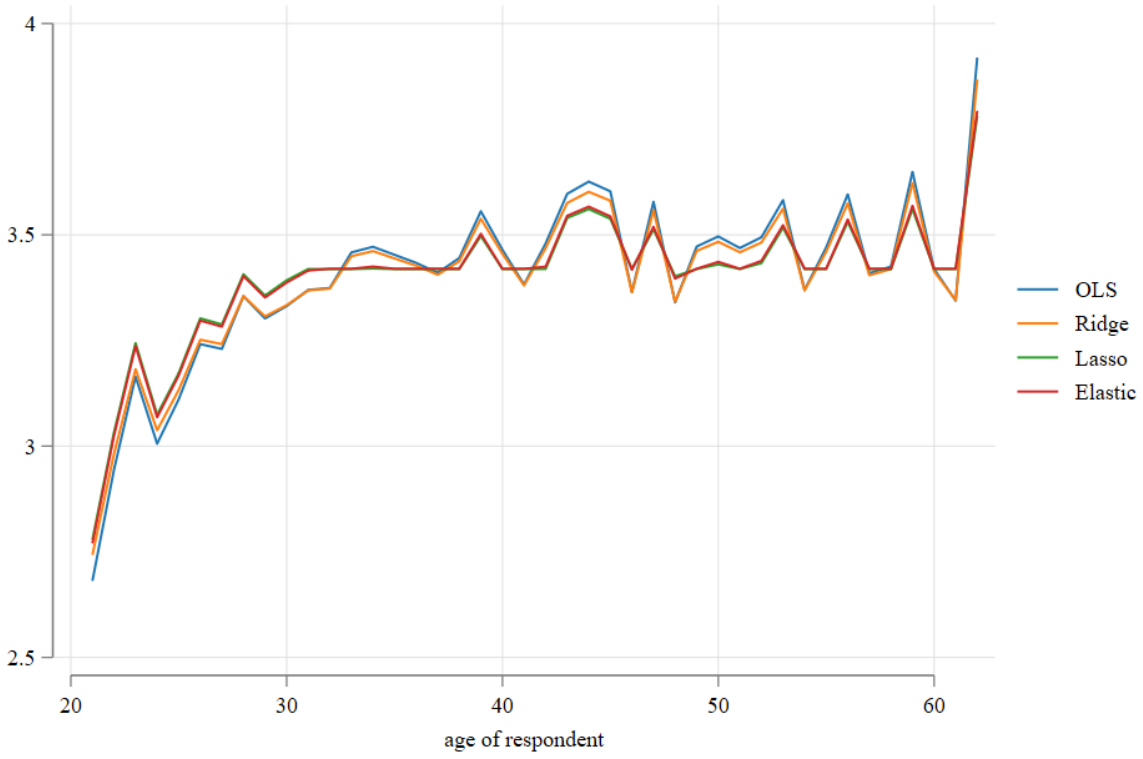
1. You need to choose tuning parameters "wisely" using approaches such as AIC, BIC, or cross validation.
2. The model you get may improve prediction, but inference is not as straight forward.
3. It also requires working with Standardized coefficients. (so the same penalty can be used for all variables in the model.

Nevertheless, they can be used as starting point for model selection.

if interested, look into `Stata` introduction to `Lasso` regression. `help Lasso intro`

Brief Example:

```
qui {
frause oaxaca, clear
keep if lnwage!=.
qui:reg lnwage i.age
predict p_ols
qui:elasticnet linear lnwage i.age, selection(cv, alllambdas)  alpha(0)
predict p_ridge
qui:lasso linear lnwage i.age, selection(cv, alllambdas)
predict p_lasso
qui:elasticnet linear lnwage i.age, selection(cv, alllambdas)
predict p_elastic
}
```


4
3.5
3
2.5
20
30
40
50
60
age of respondent
OLS
Ridge
Lasso
Elastic

Shrinking Coefficients

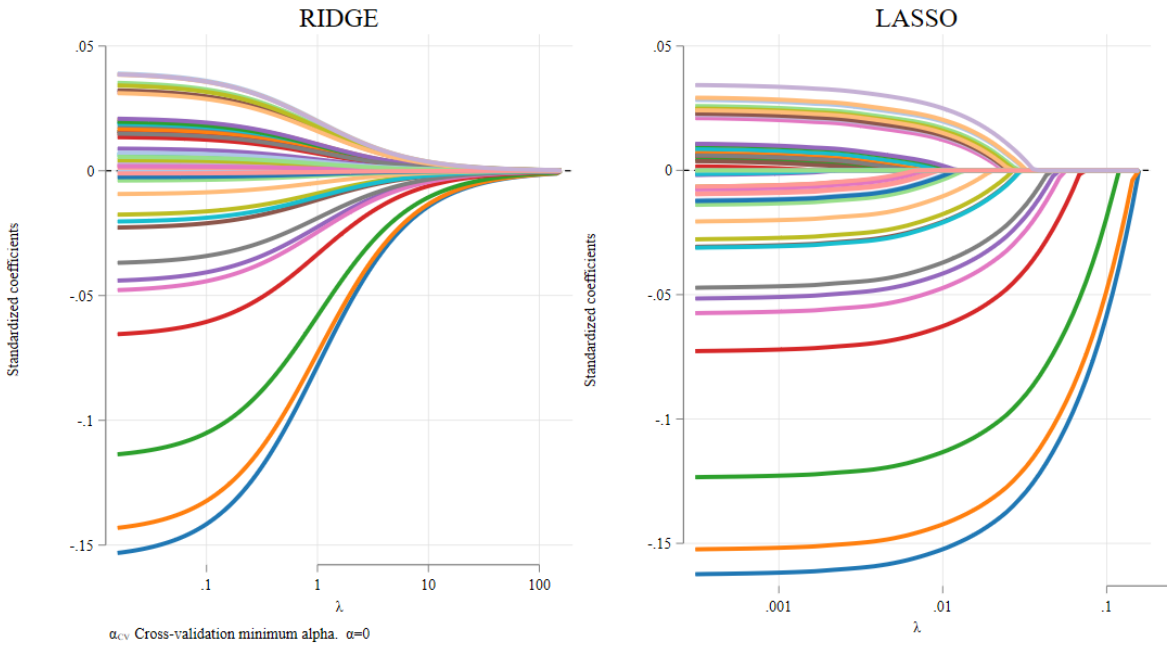


Figure 3: Lasso vs Ridge

Next: Non & Semi Parametric models