PLSC 503: "Multivariate Analysis for Political Research"

MLE: Testing, Inference, and "Robust" Variance Estimators
April 6, 2017

Inference & Testing

Inference, In General

1. Pick some $\mathbf{H}_A : \mathbf{\Theta} = \mathbf{\Theta}_A$
2. Estimate $\hat{\mathbf{\Theta}}$
3. Determine distribution of $\hat{\mathbf{\Theta}}$ under $\mathbf{H}_A$
4. Use (2) and (3) $\rightarrow \hat{\mathbf{S}} \sim h(\mathbf{\Theta}, \hat{\mathbf{\Theta}})$ (*test statistic*)
5. Assess $\Pr(\hat{\mathbf{S}}|\mathbf{H}_A)$

MLEs and Inference

$$\hat{\mathbf{\Theta}} \overset{a}{\sim} \mathbf{N}[\mathbf{\Theta}, \mathbf{I}(\hat{\mathbf{\Theta}})]$$

Means that

$$\frac{\hat{\theta}_k - \theta_k}{\sqrt{\hat{\sigma}_k^2}} \sim N(0,1)$$

Single Coefficients: Significance Testing

- Choose θ_A
- Estimate $\hat{\theta}_k$, $\hat{\sigma}_k^2$
- Compare $z_k = \frac{\hat{\theta}_k - \theta_A}{\sqrt{\hat{\sigma}_k^2}}$ to a z-table
- (Or, just look at your output...)

Single Coefficients: Confidence Intervals

- $\alpha \in (0,1)$ = desired level of "significance"
- $(1-\alpha) \times 100$-percent confidence intervals for $\hat{\theta}_k$:

$$\hat{\theta}_k \pm \left(z_\alpha \sqrt{\hat{\sigma}_k^2}\right)$$

- (Or just look at your output...)

More general tests: "The Trinity"

- Likelihood-Ratio
- Wald
- Lagrangian Multiplier ("Score")

LR Tests

$$L(\hat{\mathbf{\Theta}}) \geq L(\mathbf{\Theta_A}), \text{ but}$$

By how much?

Odds of one things vs. another:

$$\frac{\Pr(\text{Something})}{\Pr(\text{Something Else})}$$

$$\frac{L(\mathbf{\Theta_A})}{L(\hat{\mathbf{\Theta}})} \ \ (\leq 1)$$

Suggests

$$\ln L(\mathbf{\Theta_A}) - \ln L(\hat{\mathbf{\Theta}}) \ \ (\leq 0)$$

$$-2[\ln L(\mathbf{\Theta_A}) - \ln L(\hat{\mathbf{\Theta}})] \overset{a}{\sim} \chi^2_r$$

Restrictions

$$\mathbf{R\Theta} = \mathbf{r}$$

$$\theta_2 = -2 \iff (0 \ 1 \ 0) \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix} = -2$$

$$r = \text{rows}(\mathbf{R}) \in \{0, K\}$$

LR Tests, Practically

- Intuition: Difference in $\ln L$ under constraint(s)
- Asymptotic
- Unreliable if $r > 100$ (or so)
- Easy to compute, but
- Requires that we have $\ln L(\mathbf{\Theta_A})$ *and* $\ln L(\hat{\mathbf{\Theta}})$

Wald Tests

Idea: If $\mathbf{\Theta_A}$, then

$$\mathbf{R\Theta} = \mathbf{r}$$

$$\mathbf{R\Theta} - \mathbf{r} = \mathbf{0}$$

But...

- We have only $\hat{\mathbf{\Theta}}$ (from sample data)
- Possible that $\mathbf{R}\hat{\mathbf{\Theta}} - \mathbf{r} = \mathbf{0}$ *due to chance* (sampling variability).
- Solution: Account for *variability* in $\hat{\mathbf{\Theta}}$.

Test:

$$\mathbf{W} = (\mathbf{R}\hat{\mathbf{\Theta}} - \mathbf{r})' \left[\mathbf{R}\,\mathrm{Var}(\hat{\mathbf{\Theta}})\,\mathbf{R}'\right]^{-1} (\mathbf{R}\hat{\mathbf{\Theta}} - \mathbf{r})$$

$$\mathbf{W} \overset{a}{\sim} \chi^2_r$$

<u>Two-Handed Wald Tests</u>

+ Easy, fast
+ No need for $\ln L(\mathbf{\Theta_A})$
− Uses $\mathrm{Var}(\hat{\mathbf{\Theta}})$, not $\mathrm{Var}(\mathbf{\Theta_A})$ (potentially poor coverage)
− Can yield nonsensical results

Lagrange Multiplier ("LM," a/k/a "Score") Tests

Idea: If $\mathbf{\Theta_A}$, then

$$\left.\frac{\partial \ln L}{\partial \theta}\right|_{\mathbf{\Theta_A}} \approx \mathbf{0}$$

Consider a new problem:

$$\max_{\mathbf{\Theta}} \left[L(\mathbf{\Theta}) - \boldsymbol{\lambda}(\mathbf{\Theta} - \mathbf{\Theta_A})\right]$$

Yields:

$$\tilde{\mathbf{\Theta}} = \mathbf{\Theta_A}$$

$$\tilde{\boldsymbol{\lambda}} = \mathbf{g}(\tilde{\mathbf{\Theta}})$$

Suggests

$$LM = \mathbf{g}(\tilde{\mathbf{\Theta}})' \, \mathbf{I}(\tilde{\mathbf{\Theta}})^{-1} \mathbf{g}(\tilde{\mathbf{\Theta}})$$

$$LM \overset{a}{\sim} \chi^2_r$$

Note: No need for $\hat{\mathbf{\Theta}}$!

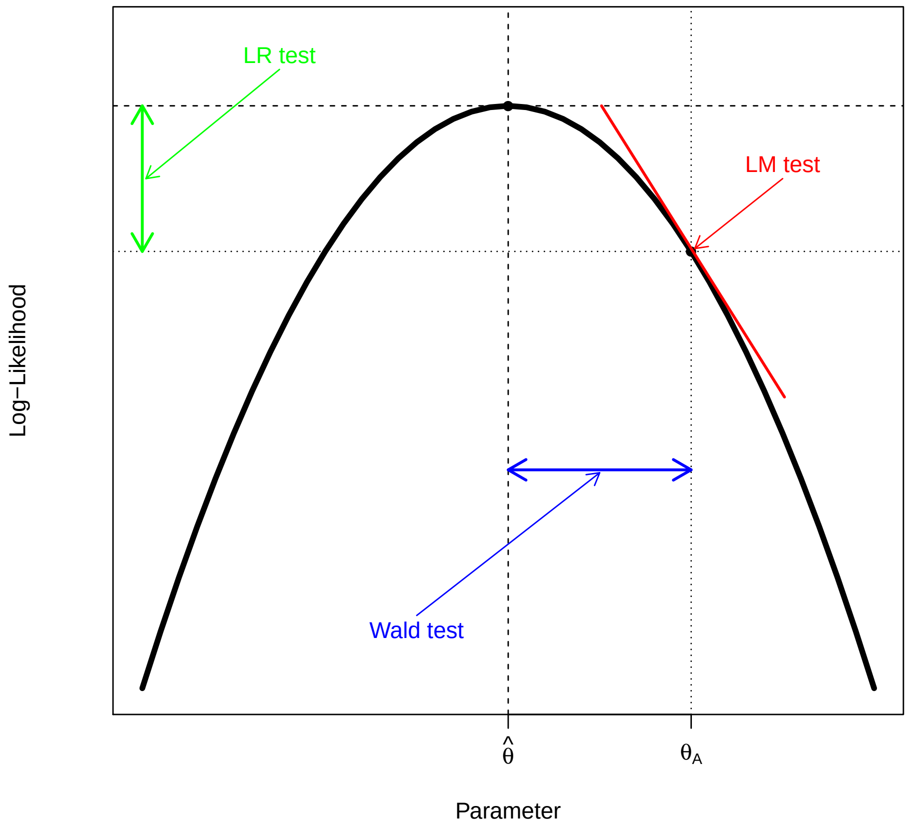
Tests, Conceptually (Charles Franklin remix)

- The LR asks, "Did the likelihood change much under the null hypotheses versus the alternative?"
- The Wald test asks, "Are the estimated parameters very far away from what they would be under the null hypothesis?"
- The LM test asks, "If I had a less restrictive likelihood function, would its derivative be close to zero here at the restricted ML estimate?"

Tests, Conceptually (h.t.: Buse 1982)

- LR test ≈ manic mountaineer
- Wald test ≈ tired mountaineer
- LM test ≈ lazy mountaineer

LR, Wald, and LM Tests: A Conceptual Illustration



Tests, Practically

- All are asymptotically identical...
- In a linear model, it can be shown that the values of the test statistics are arrayed Wald $\geq$ LR $\geq$ LM.
- Require different estimates, but similar information
- In terms of preference, generally, LR $>$ Wald $>$ LM

Software: R

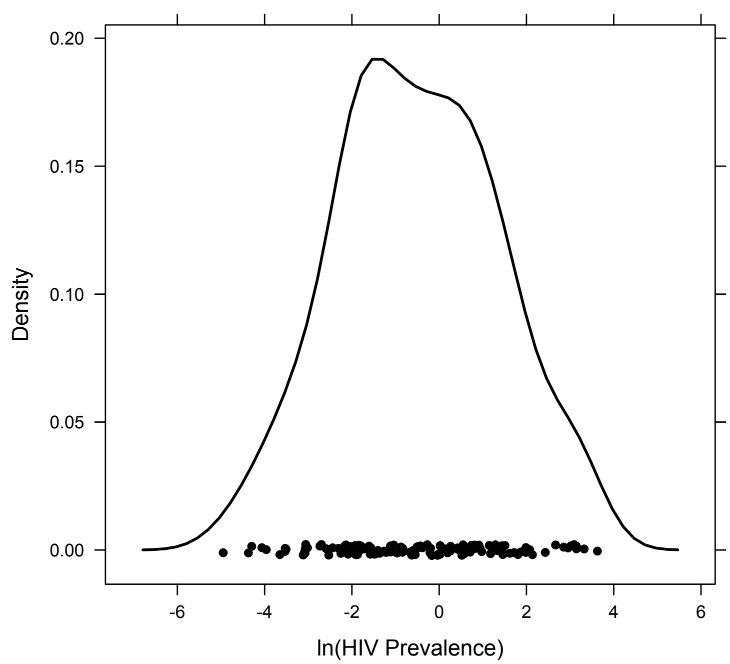
- Wald tests: `waldtest` (in `lmtest`), `wald.test` (in `aod`), etc.
- LR tests: `lrtest` (in `lmtest`), `RLRsim`, many others
- "by-hand" straightforward...

Software: Stata

- `test`, `testnl` → Wald tests
- `lrtest` → LR tests
- `waldtest` in `ml`
- LM tests require `enumopt`, `testomit` (see the example here)

Example: HIV Rates, 2005

- HIV prevalence rates, 144 countries
- Source: UNAIDS
- (Badly) Skewed → logged
- We're guessing $\sim N(\mu, \sigma^2)$...



Preliminaries

```
> library(maxLik)
> library(aod)
> library(lmtest)
> HIV<-read.dta("HIV2005.dta")
```

```
> attach(HIV)

> HIVll <- function(param) {
+   mu <- param[1]
+   sigma <- param[2]
+   ll <- -0.5*log(sigma^2) - (0.5*((x-mu)^2/sigma^2))
+   ll
+  }

> x<-logHIV
```

Estimation

```
> hats <- maxLik(HIVll, start=c(0,1))
> summary(hats)
--------------------------------------------
Maximum Likelihood estimation
Newton-Raphson maximisation, 7 iterations
Return code 1: gradient close to zero. May be a solution
Log-Likelihood: -159.5
2  free parameters
Estimates:
     Estimate Std. error t value Pr(> t)
[1,]   -0.500      0.153   -3.27  0.0011 **
[2,]    1.836      0.108   16.97  <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
--------------------------------------------
```

≡ Mean-Only Linear Model

```
> HIVLM<-lm(logHIV~1)
> summary(HIVLM)

Call:
lm(formula = logHIV ~ 1)

Residuals:
    Min      1Q  Median      3Q     Max
-4.4493 -1.3474 -0.0622  1.3012  4.1264

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.500      0.154   -3.26   0.0014 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
Residual standard error: 1.84 on 143 degrees of freedom
```

Moving parts...

```
> hats$estimate
[1] -0.5002  1.8357

> hats$gradient
[1] -1.645e-05  1.749e-04

> hats$hessian
       [,1]   [,2]
[1,] -42.73  -2.09
[2,]  -2.09 -63.63

> -(solve(hats$hessian))
           [,1]       [,2]
[1,]  2.340e-02 -2.432e-07
[2,] -2.432e-07  1.170e-02


> sqrt(-(solve(hats$hessian)))
       [,1]   [,2]
[1,] 0.1530    NaN
[2,]    NaN 0.1082
```

Wald tests

```
> wald.test(Sigma=vcov(hats),b=coef(hats),Terms=1:2,verbose=TRUE)

Wald test:
----------
Coefficients:
[1] -0.5  1.8

Var-cov matrix of the coefficients:
     [,1]     [,2]
[1,]  0.02344 -0.00077
[2,] -0.00077  0.01574
```

```
Test-design matrix:
   [,1] [,2]
L1    1    0
L2    0    1

Positions of tested coefficients in the vector of coefficients: 1, 2

H0:  -0.5002095 = 0; 1.8357192 = 0

Chi-squared test:
X2 = 298.7, df = 2, P(> X2) = 0.0
```

More Wald tests

```
> wald.test(Sigma=vcov(hats),b=coef(hats),Terms=1:2,H0=c(0,2))
Wald test:
----------

Chi-squared test:
X2 = 12.8, df = 2, P(> X2) = 0.0017

> wald.test(Sigma=vcov(hats),b=coef(hats),Terms=1:2,H0=c(-0.5,2))
Wald test:
----------

Chi-squared test:
X2 = 1.7, df = 2, P(> X2) = 0.42
```

Even More Wald tests (equivalence)

```
> wald.test(Sigma=vcov(hats),b=coef(hats),Terms=2:2,H0=2)
Wald test:
----------

Chi-squared test:
X2 = 2.3, df = 1, P(> X2) = 0.13


> ((1.836-2)/.108)^2
[1] 2.306

> pchisq(2.306,df=1,lower.tail=FALSE)
```

```
[1] 0.1289
```

A Nonsensical Wald Test

```
> wald.test(Sigma=vcov(hats),b=coef(hats),Terms=1:2,H0=c(1,-2))
Wald test:
----------

Chi-squared test:
X2 = 1353.4, df = 2, P(> X2) = 0.0
```

LR tests: Preliminaries

```
> HIVllOne <- function(param) {
+       mu <- param[1]
+    ll <- -0.5*log(4) - (0.5*((x - mu)^(2)/4))
+    ll
+  }

> hatsF <- maxLik(HIVll, start=c(0,1))
> hatsR <- maxLik(HIVllOne, start=c(0))
```

LR tests

```
> hatsF$maximum

[1] -159.5

> hatsR$maximum
[1] -160.5

> -2*(hatsR$maximum-hatsF$maximum)
[1] 2

> pchisq(-2*(hatsR$maximum-hatsF$maximum),df=1,lower.tail=FALSE)
[1] 0.1573
```

"Robust" Variance-Covariance Estimators

Linear Model: $\mathrm{Var}(\hat{\boldsymbol{\beta}})$ with $\mathbf{uu'} = \sigma^2\Omega$:

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

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

Rewrite:

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

White's Insight:

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

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

What about MLE?

Recall:

$$
\begin{aligned}
\mathrm{Var}(\hat{\theta}) &= \mathrm{E}[(\hat{\theta}-\theta)(\hat{\theta}-\theta)'] \\
&= \mathrm{E}\left[\left(-\frac{\partial^2 \ln L}{\partial \theta^2}\right)^{-1} \frac{\partial \ln L}{\partial \theta}\frac{\partial \ln L'}{\partial \theta}\left(-\frac{\partial^2 \ln L}{\partial \theta^2}\right)^{-1}\right]
\end{aligned}
$$

We assumed:

$$
\mathrm{E}\left[\frac{\partial \ln L}{\partial \theta}\frac{\partial \ln L'}{\partial \theta}\right] = \mathrm{E}\left[\frac{\partial^2 \ln L}{\partial \theta^2}\right]
$$

So,

$$
\begin{aligned}
\mathrm{Var}(\hat{\theta}) &= \left[-\mathrm{E}\left(\frac{\partial^2 \ln L}{\partial \theta^2}\right)\right]^{-1} \\
&= [\mathbf{I}(\theta)]^{-1}
\end{aligned}
$$

Alternatively:

$$
\mathrm{Var}(\hat{\theta})_{\text{Robust}} = [\mathbf{I}(\theta)]^{-1} \left(\frac{\partial \ln L}{\partial \hat{\theta}} \frac{\partial \ln L'}{\partial \hat{\theta}}\right) [\mathbf{I}(\theta)]^{-1}
$$

"Clustering"

Suppose N "clusters" $i = \{1, 2, ...N\}$, each with n_i observations $j = \{1, 2, ...n_i\}$.

Model:

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

Then:

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

An Illustration: “Regular” OLS

```
> id<-seq(1,100,1) # 100 observations
> x<-rnorm(100) # N(0,1) noise
> y<-1+1*x+rnorm(100)
> library(rms)
> fit<-ols(y~x,x=TRUE,y=TRUE)
> fit

Linear Regression Model

         n Model L.R.       d.f.         R2      Sigma
       100      76.15          1      0.533      1.004

Coefficients:
           Value Std. Error     t Pr(>|t|)
Intercept 1.0386    0.10064 10.32        0
x         0.9674    0.09147 10.58        0

Residual standard error: 1.004 on 98 degrees of freedom
Adjusted R-Squared: 0.5283
```

Further Illustration: “Robust” $\hat{V}$

```
> RVCV<-robcov(fit)
> RVCV

Linear Regression Model

         n Model L.R.       d.f.         R2      Sigma
       100      76.15          1      0.533      1.004

Coefficients:
           Value Std. Error     t Pr(>|t|)
Intercept 1.0386    0.10036 10.35        0
x         0.9674    0.08666 11.16        0

Residual standard error: 1.004 on 98 degrees of freedom
Adjusted R-Squared: 0.5283

> diag(fit$var) / diag(RVCV$var)
Intercept         x
 1.005559  1.114029
```

Attack of the Clones

```
> bigID<-rep(id,16)
> bigX<-rep(x,16)
> bigY<-rep(y,16)
> bigdata<-as.data.frame(cbind(bigID,bigY,bigX))
> bigOLS<-ols(bigY~bigX,data=bigdata,x=TRUE,y=TRUE)
> bigOLS

         n Model L.R.       d.f.         R2      Sigma
      1600       1218          1      0.533     0.9946

Residuals:
     Min       1Q   Median       3Q      Max
-2.44262 -0.78348  0.02094  0.72053  2.48939

Coefficients:
           Value Std. Error     t Pr(>|t|)
Intercept 1.0386    0.02492 41.67        0
bigX      0.9674    0.02265 42.71        0

Residual standard error: 0.9946 on 1598 degrees of freedom
Adjusted R-Squared: 0.5327
```

Peter and Hal To The Rescue

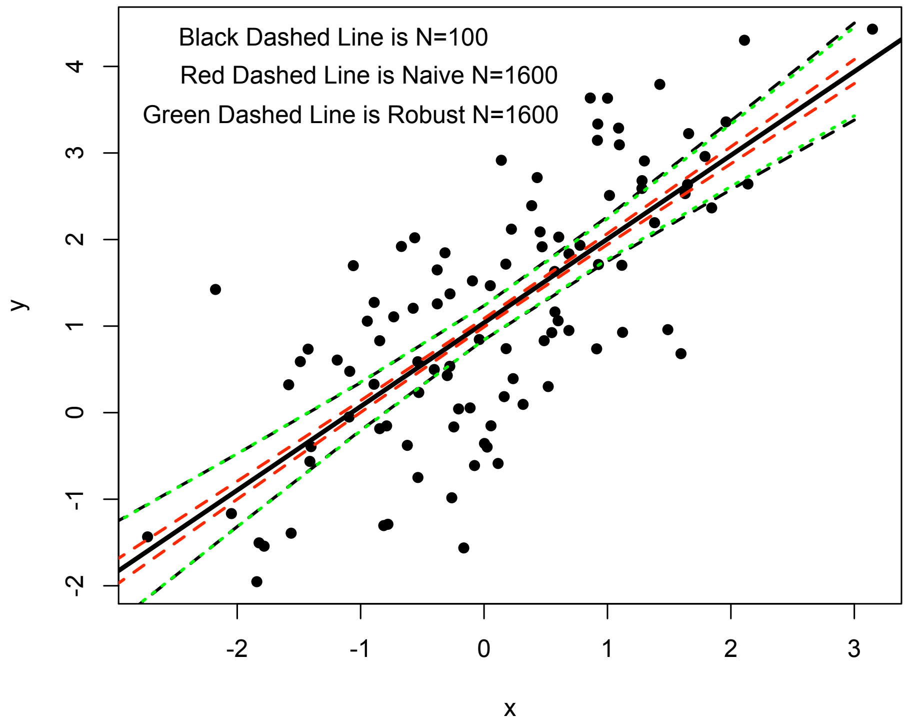
```
> BigRVCV<-robcov(bigOLS,bigdata$bigID)
> BigRVCV

         n Model L.R.       d.f.         R2      Sigma
      1600       1218          1      0.533     0.9946

Residuals:
     Min       1Q   Median       3Q      Max
-2.44262 -0.78348  0.02094  0.72053  2.48939

Coefficients:
           Value Std. Error     t Pr(>|t|)
Intercept 1.0386    0.10036 10.35        0
bigX      0.9674    0.08666 11.16        0

Residual standard error: 0.9946 on 1598 degrees of freedom
Adjusted R-Squared: 0.5327
```



'Robust" Variance Estimators: Cautions

- Are *only* consistent (Chesher and Jewitt 1987)
- Efficiency loss if homoscedastic (Kauermann and Carroll 2001)
- "Even if the key assumption holds, bias should be of greater interest than variance, especially when the sample is large and causal inferences are based on a model that is incorrectly specied. Variances will be small, and bias may be large." (Freedman 2006)

Things you should read...

Freedman, D. A. 2006. "On the So-Called 'Huber Sandwich Estimator' and 'Robust' Standard Errors." *The American Statistician* 60:299-302.

Huber, P. J. 1967. "The Behavior of Maximum Likelihood Estimates under Nonstandard Conditions." *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* I:221-33.

White, H. 1994. *Estimation, Inference, and Specication Analysis.* New York: Cambridge University Press.

ML Software: **Stata**

`ml` is it...

- Syntax is

```
   .ml model <method> <progname> <eq>...
.ml maximize
```

- Optimizers are Newton, BHHH, BFGS, and DFP

- Many, many options...

Stata : Example

The Rayleigh again...

```
. set obs 100
. gen rayleigh = 3*sqrt(-2*ln(1-(uniform())))
. program define loglik
     args lnf beta
     qui replace `lnf' = (ln($ML_y1)-ln(`beta'^2))
        + ((-$ML_y1^2)/(2*`beta'^2))
  end
```

Stata : Example

```
. ml model lf loglik (rayleigh = one, noconstant)
. ml search
. ml maximize
                                                Number of obs   =        100
                                                Wald chi2(1)    =     400.00
Log likelihood = -198.07937                     Prob > chi2     =     0.0000

------------------------------------------------------------------------------
 rayleigh |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
----------+-------------------------------------------------------------------
      one |   2.886389   .1443195    20.00   0.000     2.603528     3.16925
------------------------------------------------------------------------------
```

HIV Example: **Stata** Remix

```
. program define HIV
   args lnf beta theta
   qui replace `lnf' = ln(normalden(($ML_y1-`beta') / `theta'))
   - ln(`theta')
end
```

```
. gen one=1
. ml model lf HIV (logHIV = one) /sigma
. ml search
```

HIV Example Redux: Results

```
. ml maximize
initial:       log likelihood = -302.50517
rescale:       log likelihood = -302.50517
rescale eq:    log likelihood = -302.50517
Iteration 0:   log likelihood = -302.50517
Iteration 1:   log likelihood = -294.30557
Iteration 2:   log likelihood = -291.79938
Iteration 3:   log likelihood = -291.79798
Iteration 4:   log likelihood = -291.79798
                                                  Number of obs   =        144
                                                  Wald chi2(0)    =          .
Log likelihood = -291.79798                       Prob > chi2     =          .
------------------------------------------------------------------------------
      logHIV |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
eq1          |
       _cons |  -.5002095   .1529766    -3.27   0.001    -.8000381   -.2003808
-------------+----------------------------------------------------------------
sigma        |
       _cons |   1.835719   .1081708    16.97   0.000     1.623708     2.04773
------------------------------------------------------------------------------
```

HIV Example Redux: Tests

```
. test [eq1]_cons [sigma]_cons

 ( 1)  [eq1]_cons = 0
 ( 2)  [sigma]_cons = 0

           chi2(  2) =  298.69
         Prob > chi2 =    0.0000



. test ([eq1]_cons=0) ([sigma]_cons=2)

 ( 1)  [eq1]_cons = 0
 ( 2)  [sigma]_cons = 2
```

```
          chi2(  2) =   13.00
        Prob > chi2 =    0.0015
```

HIV Example Redux: More Tests

```
. test ([eq1]_cons=-0.5) ([sigma]_cons=2)

 ( 1)  [eq1]_cons = -.5
 ( 2)  [sigma]_cons = 2

          chi2(  2) =    2.31
        Prob > chi2 =    0.3156


. test [sigma]_cons=2

 ( 1)  [sigma]_cons = 2

          chi2(  1) =    2.31
        Prob > chi2 =    0.1288
```

HIV Example Redux: Even More Tests

```
. testnl ([eq1]_cons=0) ([sigma]_cons=-2)

  (1)  [eq1]_cons = 0
  (2)  [sigma]_cons = -2

              chi2(2) =     1268.09
          Prob > chi2 =        0.0000




. test [sigma]_cons=2

 ( 1)  [sigma]_cons = 2

          chi2(  1) =    2.31
        Prob > chi2 =    0.1288
```