Week 6: Maximum Likelihood Estimation

Marcelo Coca Perraillon

University of Colorado
Anschutz Medical Campus

Health Services Research Methods I
HSMP 7607
2019

Updated notes are here: https://clas.ucdenver.edu/marcelo-perraillon/teaching/health-services-research-methods-i-hsmp-7607

Outline

- An alternative way of estimating parameters: Maximum likelihood estimation (MLE)
- Simple examples: Bernoulli and Normal with no covariates
- Adding explanatory variables
- Variance estimation
- Why MLE is so important?
- Likelihood ratio tests

Bernoulli example

- Suppose that we know that the following ten numbers were simulated using a Bernoulli distribution: 0 0 0 1 1 1 0 1 1 1
- We can denote them by $y_1, y_2, ..., y_{10}$. So $y_1 = 0$ and $y_{10} = 1$
- Recall that the pdf of a Bernoulli random variable is $f(y; p) = p^y(1-p)^{1-y}$, where $y \in \{0, 1\}$
- The probability of 1 is p while the probability of 0 is $(1 - p)$
- We want to figure out what is the p that was used to simulate the ten numbers
- All we know is that 1) they come from a Bernoulli distribution and 2) they are independent from each other

Bernoulli example

- Since we know the pdf that generated the numbers is Bernoulli, we know that the probability of the first number is $p^{y_1}(1-p)^{1-y_1}$
- The probability of the second is $p^{y_2}(1-p)^{1-y_2}$ and so on...
- We could replace the y_i with the actual numbers. For example, the first one is $y_1 = 0$ so the probability is just $(1-p)$. I'll keep the symbols because we are going to make the problem more general
- What we **do not know is the value of the parameter** p
- Since we know that they are **independent** we could also write down the probability of observing all 10 numbers. That is, their **joint probability**
- Since they are independent their joint distribution is the multiplication of the 10 pdfs. Recall: $p(A \cap B) = P(A)P(B)$ if A and B are independent

Bernoulli example

- We use the product symbol $\prod$ to simplify the notation. For example, $\prod_{i=1}^{2} x_i = x_1 * x_2$
- So we can write the *joint probability* or the likelihood (L) of seeing those 10 numbers as:

  $L(p) = \prod_{i=1}^{10} p^{y_i}(1-p)^{1-y_i}$

Bernoulli example

- Remember that we are trying to find the p that was used to generate the 10 numbers. That's our unknown
- In other words, we want to find the p that **maximizes** the likelihood function $L(p)$. Once we find it, we could it write as our estimated parameter as $\hat{p}$
- Yet another way: **we want to find the $\hat{p}$ that makes the joint likelihood of seeing those numbers as high as possible**
- Sounds like a calculus problem... We can take the derivative of $L(p)$ with respect to p and set it to zero to find the optimal $\hat{p}$
- Of course, the second step is to verify that it's a maximum and not a minimum (take second derivative) and also verify that is unique, etc. We will skip those steps

Bernoulli example

- Taking that derivative is complicated because we would need to use the chain rule several times. A lot easier to make it a sum so we take the log; the log function is a **monotonic transformation**, it won't change the optimal $\hat{p}$ value
- We will use several properties of the log, in particular: $log(x^a y^b) = log(x^a) + log(y^b) = a * log(x) + b * log(y)$
- So now we have (for n numbers rather than 10): $lnL(p) = \sum_{i=1}^{n} y_i ln(p) + \sum_{i=1}^{n} (1 - y_i) ln(1 - p)$
- Which simplifies to: $lnL(p) = n\bar{y} ln(p) + (n - n\bar{y}) ln(1 - p)$
- This looks a lot easier; all we have to do is take $\frac{dln(p)}{dp}$, set it to zero, and **solve for p**

Bernoulli example

$\frac{dln(p)}{dp} = \frac{n\bar{y}}{p} - \frac{(n-n\bar{y})}{(1-p)} = 0$

- After solving, we'll find that $\hat{p}(y_i) = \bar{y} = \sum_{i=1}^{n} \frac{y_i}{n}$
- So that's the MLE estimator of p. This is saying more or less the obvious: our best guess for the p that generated the data is the proportion of 1s, in this case $p = 0.6$
- We would need to verify that our estimator satisfies the three basic properties of an estimator: bias, efficiency, and consistency (this will be in your exam)
- Note that we can plug in the optimal $\hat{p}$ back into the ln likelihood function:
- $lnL(\hat{p}) = n\bar{y}ln(\hat{p}) + (n - n\bar{y})ln(1 - \hat{p}) = a$, where a will be a number that represents the **highest likelihood we can achieve** (we chose $\hat{p}$) that way

Example

- Simulated 100 Bernoulli rvs with p = 0.4

```
set obs 100
gen bernie = uniform()<0.4
sum bernie
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      bernie |        100         .46    .5009083          0          1
* We just showed that p hat is 0.46

* Let's get the highest value of the ln likelihood
* Plug in p hat and the other values

di 100*0.46*ln(0.46) + (100-100*0.46)*ln(1-0.46)
-68.994376
```

- And we just did **logistic regression** "by hand." A logistic model with only a constant (no covariates), also known as the **null model**

Example

- We will use the logit command to model indicator variables, like whether a person died

```
logit bernie
Iteration 0:   log likelihood = -68.994376
Iteration 1:   log likelihood = -68.994376

Logistic regression                               Number of obs     =        100
                                                  LR chi2(0)        =      -0.00
                                                  Prob > chi2       =          .
Log likelihood = -68.994376                       Pseudo R2         =    -0.0000

------------------------------------------------------------------------------
      bernie |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |  -.1603427   .2006431    -0.80   0.424    -.5535959    .2329106
------------------------------------------------------------------------------
di 1/(1+exp( .1603427 ))
.45999999
```

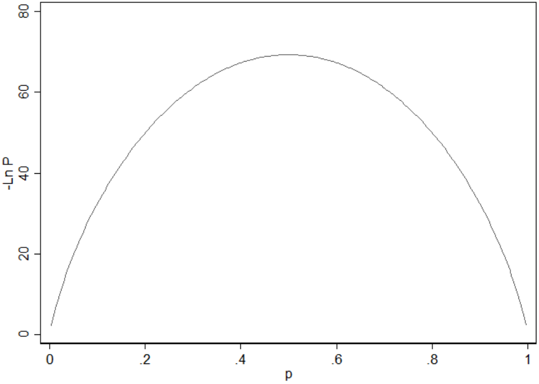
- Note that Log likelihood = -68.994376 matches what we found "by hand;" the coefficient is in the log-odds scale
- This is a model with no explanatory variables. We can **easily make the parameter p be a linear function of predictors**

Some things to note

- Statistical software maximizes the log likelihood numerically (also the *log* likelihood because of numerical precision)
- The algorithm is given a starting value for some parameters (often using using the null model)
- Each iteration “improves” the maximization
- The second derivatives are also computed (we will see why in a sec)
- In many cases, we need to be mindful of the difference between the scale of estimation and the scale of interest
- Logit models report coefficients in the log-odds scale
- Not the only way of deriving logit models. You could also assume a Binomial pdf. The Bernoulli is a special case of the Binomial when the number of trials is 1

Let's plot the - ln(L) function with respect to p

```
twoway function y= -(100*x*ln(x) + (100-100*x)*ln(1-x)), range(0 1) ///
       xtitle("p") ytitle("-Ln P") saving(1100.gph, replace)
```



What about the precision (standard error) of the estimate?

- There is some intuition in the plot above. The precision of the estimate $\hat{p}$ can be measured by the curvature of the $lnL(\theta)$ function around its peak
- A flatter curve has more uncertainty
- The Fisher information function, $I(\theta)$ formalizes that intuition: $I(\theta) = -E[\frac{\partial^2 lnL(\theta)}{\partial^2 \theta}]$
- It turns out that we can calculate $var(\theta)$ using the inverse of $I(\theta)$
- For the Bernoulli, $I(\hat{p}) = \frac{n}{[\hat{p}(1-\hat{p})]}$ (evaluated at $\hat{p}$)
- The variance is $1/I(\hat{p}) = \frac{\hat{p}(1-\hat{p})}{n}$
- Note something. **Once we know $\hat{p}$ we also know its variance**. The Normal distribution is unique in that the variance is not a function of the mean

What about covariates?

- In most applications we want to estimate the effect of covariates on the probability p
- So we could just make p a function of covariates: $p = f(x_1, x_2, ..., x_p)$
- We can't just make it a linear function like $p = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$. **Why?**
- But we can use a function that guarantees that p will be bounded between 0 and 1
- Enters the **logistic or logit** function: $\frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}}$
- Now we don't want to estimate p. The unknows are the parameters β_j
- Hence the logit or logistic model name. **See, piece of cake**. Careful with Chatterjee's textbook...

Normal example

- What about if we do the same but now we have numbers like

```
90.46561
105.1319
117.5445
102.7179
102.7788
107.6234
94.87266
95.48918
75.63886
87.40594
```

- I tell you that they were **simulated from a normal distribution** with parameters μ and σ^2. The numbers are independent. Your job is to come up with the best guess for the two parameters
- Same problem as with the Bernoulli example. We can solve it in exactly the same way

Normal example

- As before, we know the pdf of a Normal random variable and because the observations are independent we can multiply the densities:

  $L(\mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i-\mu)^2}{2\sigma^2}\right)$

- Remember the rules of exponents, in particular $e^a e^b = e^{a+b}$. We can write the likelihood as:

  $L(\mu, \sigma^2) = (\frac{1}{\sqrt{2\pi\sigma^2}})^n exp(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2)$

Normal example

- As before, we can simplify the problem by **taking the log to help us take the derivatives**. But before:
- **Alert**: Perhaps you are wondering, why are we using the pdf of the normal if we know that the probability of one number is zero? Because we can think of the pdf as giving us the probability of $y_i + d$ when $d \rightarrow 0$
- We need computers with lots of floating number ability. MLE was invented in the 50s/60s. Super difficult to implement. In the 80s, we had Commodore 64s `https://en.wikipedia.org/wiki/Commodore_64`
- Researchers could use mainframe computers with punch cards. Your iPhone is faster than mainframes that used fit in a building... `https://en.wikipedia.org/wiki/Mainframe_computer`
- Maybe you were not wondering that but I was at some point. I wonder a lot in general. And daydream on a minute by minute basis...

Normal example

- After taking the ln, we have:
  $lnL(\mu, \sigma^2) = -\frac{n}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2$
- All we have left is to take the derivative with respect to our two unknowns, μ and σ^2 and set them to zero. Let's start with μ:
  $\frac{\partial ln(L(\mu,\sigma^2))}{\partial \mu} = 2\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu) = 0$
- The above expression reduces to (I added the ˆ to emphasize that's the optimal):

  $\sum_{i=1}^{n} (y_i - \hat{\mu}) = 0$
- **Does it look familiar?** Replace $\hat{\mu}$ with $\hat{y}_i$. That's exactly the same as the first order condition we saw when minimizing the sum of squares
- Solving, we find that $\hat{\mu} = \frac{\sum_{i=1}^{n} y_i}{n} = \bar{y}$. In other words, our best guess is just the mean of the numbers

Normal example

- We can also figure out the variance by taking the derivative with respect to σ^2
- We will find that $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{\mu})}{n}$
- If you remember the review lecture on probability and statistics, we know that this formula is biased. We need to divide by $(n-1)$ instead
- (What is the definition of bias?)
- **This happens often in MLE**. The MLE estimate of the variance is often biased but it is easy to correct for it

Normal example Stata

- We just figured out that the best guess is to calculate the sample mean and sample variance
- We can easily verify in Stata

```
clear
set seed 1234567
set obs 100
gen ynorm = rnormal(100, 10)
sum ynorm
   Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      ynorm |        100    98.52294    10.03931   74.16368   123.5079
```

- The sum commands divides the sample variance by (n-1)

Linear regression: adding covariates

- What about if I told you that the number I generated is a linear function of one variable, say, x_1? In other words, I'm saying that the mean of the normal distribution is $\mu = \beta_0 + \beta_1 x_1$
- Now we want to find the parameters $\beta_0, \beta_1, \sigma^2$ that maximize the likelihood function. Once we know the optimal $\hat{\beta_0}, \hat{\beta_1}$ we find the optimal $\hat{\mu}$
- The likelihood function is now:
  $L(\beta_0, \beta_1, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{1i})^2)$
- The ln likelihood is:
  $lnL(\mu, \sigma^2) = -\frac{n}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{1i})^2$

Linear regression

- If we take the derivatives with respect to β_0 and β_1 we will find exactly the **same first order conditions** as we did with OLS. For example, with respect to β_1:

  $\sum_{i=1}^{n} x_1(y_i - \beta_0 - \beta_1 x_1) = 0$
- **All the algebraic properties of OLS still hold true here**
- The MLE estimate of σ^2 will be biased but we divide by (n-p-1) instead as we saw before
- So what do we gain with MLE?
- **We do gain a lot in the understanding of linear regression**

The regression command again

- The regression command does not use MLE but it does give you the log likelihood

```
use auto
qui reg price weight mpg
ereturn list
scalars:
                  e(N) =  74
               e(df_m) =  2
               e(df_r) =  71
                  e(F) =  14.7398153853841
                 e(r2) =  .2933891231947529
               e(rmse) =  2514.028573297152
                e(mss) =  186321279.739451
                e(rss) =  448744116.3821706
               e(r2_a) =  .27348459145376
                 e(ll) =  -682.8636883111164
               e(ll_0) =  -695.7128688987767
               e(rank) =  3
```

- The log likelihood of the estimated model is stored in $e(ll)$. The log likelihood of the null model (with no covariates) is stored in $e(ll_0)$.
- From the numbers above $e(ll) > e(ll_0)$

The regression command again

- Stata uses a formula to go from SSE to log likelihood. Remember, SSE is Stata is stored in the scalar r(rss)

```
sysuse auto, clear
qui reg price weight mpg
* Save sample size and SSE
local N   = e(N)
local rss = e(rss)
* Use formula
local ll  = -0.5*‘N’*(ln(2*_pi)+ln(‘rss’/‘N’)+1)
display %20.6f ‘ll’
-682.863688
display %20.6f e(ll)
-682.863688
```

- The formula is $-0.5N(ln(2\pi) + ln(\frac{SSE}{N}) + 1)$

Easy MLE in Stata

- To estimate in MLE using Stata you need to write a program but Stata now makes it a lot easier (for teaching purposes) with the mlexp command

```
mlexp (ln(normalden(price, {xb: weight mpg _cons}, {sigma})))
initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -803.76324
rescale:       log likelihood = -729.85758
rescale eq:    log likelihood =  -697.2346
Iteration 0:   log likelihood =  -697.2346
Iteration 1:   log likelihood =  -687.4506
Iteration 2:   log likelihood = -682.92425
Iteration 3:   log likelihood = -682.86401
Iteration 4:   log likelihood = -682.86369
Iteration 5:   log likelihood = -682.86369
Maximum likelihood estimation
Log likelihood = -682.86369                     Number of obs     =         74
------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
xb           |
      weight |   1.746559   .6282189     2.78   0.005     .5152727    2.977846
         mpg |  -49.51222   84.39157    -0.59   0.557    -214.9167    115.8922
       _cons |   1946.069   3523.382     0.55   0.581    -4959.634    8851.771
-------------+----------------------------------------------------------------
      /sigma |   2462.542   202.4197    12.17   0.000     2065.806    2859.277
------------------------------------------------------------------------------
```

Almost same

- The SEs are slightly different and so is Root MSE. Stata is using the second derivatives to calculate SEs using MLE

```
. reg price weight mpg

      Source |       SS           df       MS      Number of obs   =        74
-------------+----------------------------------   F(2, 71)        =     14.74
       Model |   186321280         2  93160639.9   Prob > F        =    0.0000
    Residual |   448744116        71  6320339.67   R-squared       =    0.2934
-------------+----------------------------------   Adj R-squared   =    0.2735
       Total |   635065396        73  8699525.97   Root MSE        =      2514

------------------------------------------------------------------------------
       price |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |   1.746559   .6413538     2.72   0.008      .467736    3.025382
         mpg |  -49.51222   86.15604    -0.57   0.567    -221.3025     122.278
       _cons |   1946.069    3597.05     0.54   0.590    -5226.245    9118.382
------------------------------------------------------------------------------
```

Asymptotic properties are so important in stats

- The auto dataset has only 74 obs. What about we use the MEPS that has about 15000? (That’s really an overkill but just to make the point)

```
mlexp (ln(normalden(lexp, {xb: age _cons}, {sigma})))
initial:       log likelihood =     -<inf>  (could not be evaluated)
could not find feasible values
* I tried giving it starting values but didn’t work. Easier to do it the old fashioned way

* Create program defining model and likelihood function
capture program drop lfols
program lfols
  args lnf xb lnsigma
  local y "$ML_y1"
  quietly replace ‘lnf’ = ln(normalden(‘y’, ‘xb’,exp(‘lnsigma’)))
end
*Estimate model
 ml model lf lfols (xb: lexp =  age female) (lnsigma:)
 ml maximize
 * Sigma estimated in the log scale so it’s positive
 display exp([lnsigma]_cons)
 reg lexp age female
```

Asymptotic properties are so important in stats

```
 ml maximize
initial:       log likelihood =  -453412.9
alternative:   log likelihood = -163550.49
.
Iteration 5:   log likelihood =  -29153.79
                                                Number of obs     =     15,946
                                                Wald chi2(2)      =    2981.22
Log likelihood =  -29153.79                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------
        lexp |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
xb           |
         age |   .0358123   .0006779    52.82   0.000     .0344836    .0371411
      female |   .3511679    .024252    14.48   0.000      .303635    .3987009
       _cons |   5.329011   .0373155   142.81   0.000     5.255874    5.402148
-------------+----------------------------------------------------------------
lnsigma      |
       _cons |   .4093438   .0055996    73.10   0.000     .3983687    .4203189
------------------------------------------------------------------------------
display exp([lnsigma]_cons)
1.5058293
```

Asymptotic properties are SO IMPORTANT in stats

```
reg lexp age female
      Source |       SS           df       MS      Number of obs   =    15,946
-------------+----------------------------------   F(2, 15943)     =   1490.33
       Model |  6759.97668         2  3379.98834   Prob > F        =    0.0000
    Residual |  36157.9049    15,943  2.26794862   R-squared       =    0.1575
-------------+----------------------------------   Adj R-squared   =    0.1574
       Total |  42917.8816    15,945  2.69162004   Root MSE        =     1.506

------------------------------------------------------------------------------
        lexp |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .0358123    .000678    52.82   0.000     .0344834    .0371413
      female |   .3511679   .0242542    14.48   0.000     .3036269     .398709
       _cons |   5.329011    .037319   142.80   0.000     5.255861     5.40216
------------------------------------------------------------------------------
```

- One should be tolerant of small decimal differences. Which brings me back to why is everybody (including medical journals) so obsessed with $p = 0.05$ being the sacred line? What's the difference between $p = 0.05$ and $p = 0.06$ or $p = 0.04$? Makes little sense

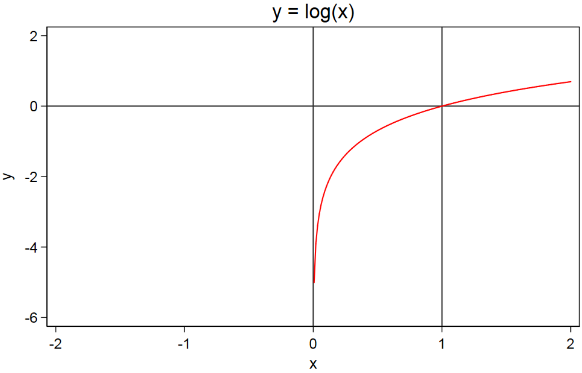
So is Stata taking derivatives and finding formulas? Nope

- Stata uses numerical methods to maximize the likelihood. There are many and some work better than others in some situations. Type “help mle” for the gory details
- A classic one is the **Newton-Raphson algorithm**
- The idea requires Taylor expansions (a way to approximate nonlinear functions using linear functions)
- The steps are:
  1. Make a guess about the parameters, say just one parameter θ_0
  2. Approximate the log likelihood function using Taylor series at θ_0 and set it equal to zero (easier to solve because it's a linear function)
  3. Find the new θ, say, θ_1. Check if the log likelihood has improved
  4. Repeat until the -2 log likelihood changes by only a small amount, say 0.02
- The idea of using -2 log likelihood < 0.02 is that that amount would not change statistical inference since -2 log likelihood is in the Chi-square scale (more on this in a sec)

Why is the log likelihood function negative?

- The likelihood function $L(p)$ is a small number since it's the joint likelihood of observing the outcome values
- Different type of MLE methods

```
twoway function y =log(x), range(-2 2) xline(0 1) yline(0) ///
       color(red) title("y = log(x)")
graph export logy.png, replace
```



What we get from MLE

- 1) It is clear that we are modeling a **conditional expectation function**: $E[Y|X]$
- Perhaps this got lost but it's worth repeating. We started with the normal density:
- $f(y;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\big(\frac{-(y_i-\mu)^2}{2\sigma^2}\big)$
- We then said that the mean μ is a **function of one or more covariates** x and we made **no assumptions** about the distribution of x:
- $f(y;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\big(\frac{-(y_i-(\beta_0+\beta_1 x_i))^2}{2\sigma^2}\big)$
- That's why I said many times that the assumption $\epsilon \sim N(0,\sigma^2)$ is the same as saying that the assumption is $y \sim N(\beta_0+\beta_1 x,\sigma^2)$, since $\mu = \beta_0 + \beta_1 x$
- Note that with MLE we did **not assume anything about the errors**. In fact, the errors are not even in the equations

What we get from MLE

- 2) It is clear from the start of setting up the problem that we are assuming that Y distributes normal **conditional on the values of** X. **Remember the example of heights for men and women**. In some cases, perfectly valid to use a linear model **even if the distribution of** Y **does not look like a normal**. See `http://tiny.cc/1r26qy`
- 3) It is clear that we assume that the observations are independent; otherwise, we **cannot multiply the densities**
- 4) The value of the optimal log likelihood function gives us a measure of the goodness of fit, much like SSR (i.e. the explained part) did. By comparing the log likelihood of alternative models, we will test if the reduced model is adequate like we did with the F test
- 5) The **curvature** of the log likelihood function provides information about the precision of the estimates (i.e. standard errors)

What we get from MLE

- 6) MLE is much more general than OLS. You will use MLE for logit, Probit, Poisson, mixture models, survival models. Pretty much all the standard models an applied researcher needs
- 7) Learning to model using likelihood ratio tests is more useful for more type of models than using the SSE for nested models
- 8) AIC and BIC to compare non-nested models are based on the log likelihood function
- Here is a more detailed proof of MLE for the normal: `https://www.statlect.com/fundamentals-of-statistics/normal-distribution-maximum-likelihood`

Likelihood ratio test (LRT)

- The null H_0 is that the restricted (constrained) model is adequate
- The alternative H_1 is that the full (unconstrained) model is adequate
- The likelihood ratio test compares the log-likelihoods of both models and can be written as:
- $LR = -2[L(RM) - L(FM)]$, where $L(RM)$ is the log-likelihood of the restricted model and $L(FM)$ that of the full model
- Under the null that the restricted model is adequate, the test statistics LR distributes χ^2 with degrees of freedom given by $df = df_{full} - df_{restricted}$; that is, the difference in degrees of freedom between the restricted and full models

Likelihood ratio test: sketch of theory

- The theory of LRTs is a bit dense but the intuition is not that difficult to understand
- We could re-write as $LR = -2L(\frac{RM}{FM})$ since $log(\frac{a}{b}) = log(a) - log(b)$
- So we are comparing the likelihood of the reduced model to the full model and wondering if the reduced model alone is just fine. Sounds familiar? Not that different from the F test comparing SSEs of nested models
- Keep in mind that the estimated model parameters are those that **maximized** the value of the likelihood
- The more theoretical part is to figure out how the LRT distributes and under which conditions the LRT is valid (models must be nested)

Recall the F test

- We have $LR = -2L(\frac{RM}{FM})$
- The F test was $F = \frac{[SSE(RM)-SSE(FM)]/(p+1-k)}{SSE(FM)/(n-p-1)}$
- Both are using a measure of fit to compare models
- With MLE, we want to know if reaching a higher likelihood is due to chance under the null
- With the F test, we want to know if the additional reduction in the residual variance is due to chance under the null
- The requirement is that models must be nested

Example

- Compare the likelihood and other criteria

```
qui reg colgpa
est sto m1
...
est table m1 m2 m3, star stat(r2 r2_a ll bic aic) b(%7.3f)
-----------------------------------------------------
    Variable |     m1           m2           m3
-------------+---------------------------------------
       hsgpa |                0.482***     0.459***
     skipped |                            -0.077**
       _cons |   3.057***     1.415***     1.579***
-------------+---------------------------------------
          r2 |   0.000        0.172        0.223
        r2_a |   0.000        0.166        0.211
          ll | -60.257      -46.963      -42.493
         bic | 125.462      103.823       99.832
         aic | 122.513       97.925       90.985
-----------------------------------------------------
             legend: * p<0.05; ** p<0.01; *** p<0.001
```

- Note that the log likelihood (ll) gets larger for better fitting models; we will cover AIC and BIC later

Example

- LR tests

```
lrtest m3 m2
Likelihood-ratio test                                 LR chi2(1)  =      8.94
(Assumption: m2 nested in m3)                         Prob > chi2 =    0.0028

. lrtest m3 m1

Likelihood-ratio test                                 LR chi2(2)  =     35.53
(Assumption: m1 nested in m3)                         Prob > chi2 =    0.0000
```

- It seems logical that LRT and F-test comparing nested models should be equivalent (asymptotically)

LRT and F-tests

- Compare tests

```
qui reg colgpa
est sto m0
scalar ll0 = e(ll)
reg colgpa male campus
      Source |       SS           df       MS      Number of obs   =       141
-------------+----------------------------------   F(2, 138)       =      0.62
       Model |  .171856209         2  .085928105   Prob > F        =    0.5413
    Residual |  19.2342432       138  .139378574   R-squared       =    0.0089
-------------+----------------------------------   Adj R-squared   =   -0.0055
       Total |  19.4060994       140  .138614996   Root MSE        =    .37333
...

est sto m1
scalar ll1 = e(ll)

lrtest m0 m1
Likelihood-ratio test                                 LR chi2(2)  =      1.25
(Assumption: m0 nested in m1)                         Prob > chi2 =    0.5341

 * By hand
di -2*[ll0 - ll1]
1.2542272
```

- p-value of both 0.5341 (I chose bad predictors so p-values would be high)

Summary

- MLE is not more difficult than OLS
- The advantage of learning MLE is that it is by far the most general estimation method
- Learning the concept of log-likelihood and LRT will help us when modeling linear models, logistics, Probit, Poisson and many more
- AIC and BIC use the log-likelihood
- We are using the log-likelihood in a similar way we used SSR, although we did the F-test in terms of SSE but we know that SST = SSE + SSR
- Never forget the main lesson of MLE with a normal: **We are modeling the mean as a function of variables**
- See more examples under Code on my website: `http://tinyurl.com/mcperraillon`