Week 6: Maximum Likelihood Estimation

Marcelo Coca Perraillon

University of Colorado
Anschutz Medical Campus

Health Services Research Methods I
HSMP 7607
2020

Outline

- An alternative way of estimating parameters: Maximum likelihood estimation (MLE)
- Simple examples: Bernoulli and Normal with no covariates
- Adding explanatory variables
- Variance estimation
- Intuition about the linear model using MLE
- Likelihood ratio tests, AIC, BIC to compare models
- Logit and probit with a latent variable formulation
- Predictions in logit and probit models

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. That's what we do not know
- What we do know is 1) they come from a Bernoulli distribution and 2) they are independent from each other, meaning that knowing one of them doesn't tell us anything about the other(s)

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 in a bit
- 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

- The joint probability is $[p^{y_1}(1-p)^{1-y_1}] \times [p^{y_2}(1-p)^{1-y_2}] \times \cdots \times [p^{y_1 0}(1-p)^{1-y_0}]$
- To make notation easier we'll use the product symbol $\prod$. 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:

  $L(p) = \prod_{i=1}^{10} p^{y_i}(1-p)^{1-y_i}$
- Again, that is the probability that we observed the 10 numbers, although we do not know p yet

Bernoulli example

- Here comes the key insight. We can turn this problem into a maximization problem. What is the number p that **maximizes** the likelihood function $L(p)$ given the observed data?
- That number will be the estimated parameter p that we'll call $\hat{p}$
- Yet another way: **we want to find the $\hat{p}$ that makes the joint likelihood of seeing those numbers as high as possible**
- That's of course 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 to keep this problem simple

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 by taking 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:
  $log(x^a y^b) = log(x^a) + log(y^b) = a * log(x) + b * log(y)$
- We are going to generalize the problem by assuming we have n numbers instead of 10. It doesn't make it more difficult. We have:

  $lnL(p) = \sum_{i=1}^{n} y_i ln(p) + \sum_{i=1}^{n} (1 - y_i) ln(1 - p)$
- Which simplifies to: $lnL(p) = n\bar{y} ln(p) + (n - n\bar{y}) ln(1 - p)$
- ($\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$) 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 $p = 0.6$, which is the same as $E[Y_i]$
- 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)
- **Drum roll**: We just discovered logistic regression

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
```

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

Replicate using the logit command

- We have used logit before

```
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
```

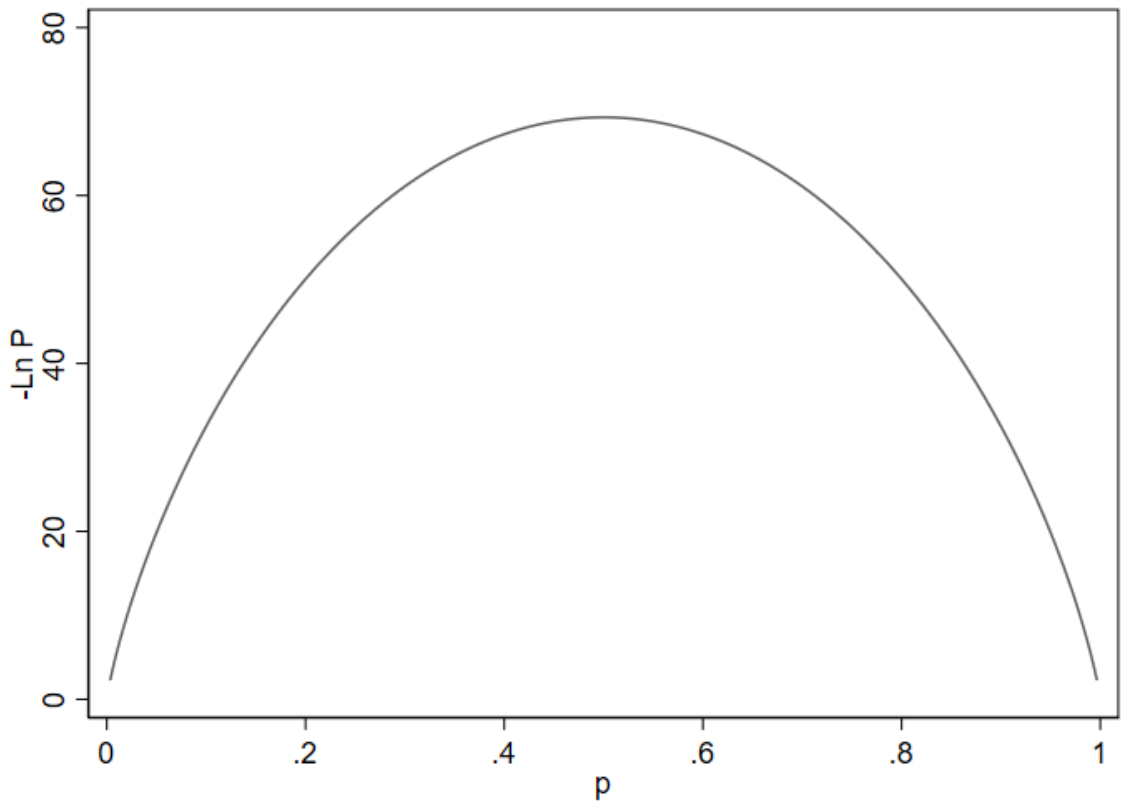
- Note that Log likelihood = -68.994376 matches what we found "by hand;" the coefficient is in the log-odds scale, but we can transform it
- This is a model with no explanatory variables. We can **easily make the parameter p be a linear function of predictors**

Some things to note

- Statistical software maximizes the log likelihood **numerically**. Also the *log* likelihood because of numerical precision. The probability of each y_i is small, so $p(y_1) \times p(y_2) \times \ldots p(y_n)$ could be a very small number
- 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 because the second derivatives are used to compute SEs
- 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. We will see a **variable** approach, too

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(ll00.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. In logistic models, the variance is function of the p
- 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 can change independently of the mean

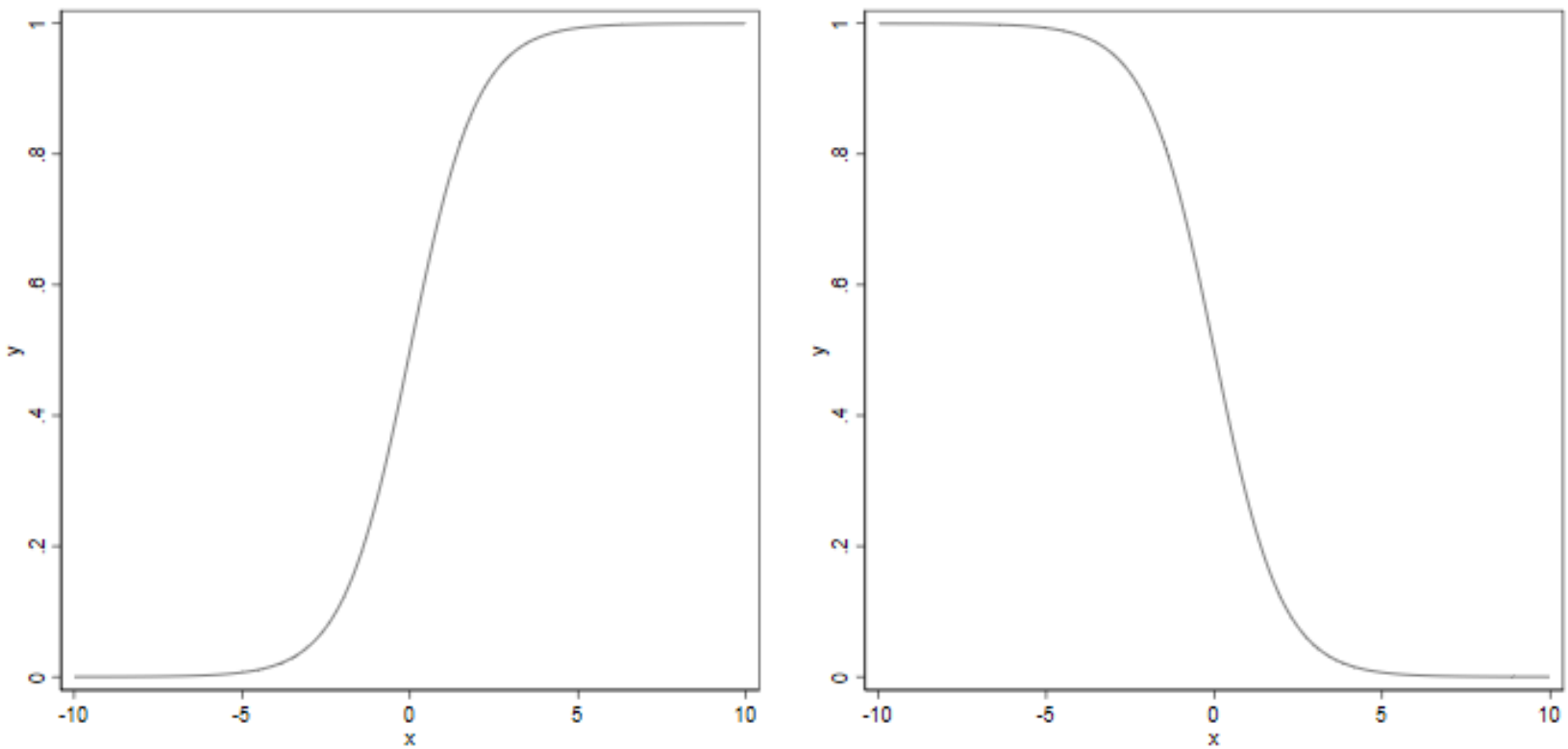
Logit models with covariates

- 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: $p = f(x_1, x_2, ..., x_p)$
- The easiest one is to make them a linear function like $p = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$
- But that could be a problem since we need to guarantee that p will be bounded between 0 and 1 since it's a probability
- Enters the **logistic or logit** function: $\frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}} = \frac{e^{(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}}{1+e^{(\beta_0+\beta_1 x_1+\cdots+\beta_p x_p)}}$
- We have now changed the maximization problem. We do not maximize the likelihood with respect to p. Now we do it with respect to the parameters β_j
- The math is more difficult. We need matrix algebra. The second derivative becomes a matrix too (the Jacobian). The logic is the same, though

Logistic response function

- If we constrain the response to be between 0 and 1, it can't be linear with respect to X

```
twoway function y=exp(x) / (1+ exp(x)), range(-10 10) saving(l1.gph, replace)
twoway function y=exp(-x) / (1+ exp(-x)), range(-10 10) saving(l2.gph, replace)
graph combine l1.gph l2.gph, xsize(20) ysize(10)
graph export lboth.png, replace
```



Logistic or logit model

- Notice a couple of things.The effect of x on π is not linear; the effect depends on the value of x
- But we can make the function linear using the so-called logit transformation
  $ln(\frac{\pi}{1-\pi}) = x$
- I made you go the other way in one homework. If you solve for π you get to the logistic response function
- More general, the model is:
- $ln(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$, which transformed is
  $p = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$
- Can also be written as: $p = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\cdots+\beta_p X_p)}}$

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:

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

-

- **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 $d \to 0$
- If you recall your stats 101 classes, with a continuous rv the probability of a number is zero

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 ^ 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 you saw when minimizing the sum of squares (that also the moment condition in Generalized Method of Moments, GMM)
- 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. $E[Y_i]$ again

Normal example

- We can also figure out the variance by taking the derivative with respect to σ^2
- We will find that $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{\mu})}{n}$
- That formula happens to be wrong. That's a biased estimate of the variance. We need to divide by $(n-1)$ instead
- (What is the definition of bias?)
- **This is not unusual in MLE**. The MLE estimate of the variance is often biased but it is easy to correct them (and less of a problem with large samples)

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

reg ynorm
<... output omitted ...>
------------------------------------------------------------------------------
       ynorm |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
------------+----------------------------------------------------------------
       _cons |   98.52294   1.003931    98.14   0.000     96.53092     100.515
------------------------------------------------------------------------------
```

- **Drum roll 2**: We just did linear regression "by hand" (although without covariates)

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 with. 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** (more in a little bit)

The regression command again

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

```
sysuse auto, clear
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 $ll = -0.5N(ln(2\pi) + ln(\frac{SSE}{N}) + 1)$

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
------------------------------------------------------------------------------
```

But they are consistent

- 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)
- OLS version:

```
use heus_mepssample.dta, clear
gen lexp = log(exp_tot +1)
reg lexp age female
```

```
      Source |       SS           df       MS      Number of obs   =    19,386
-------------+----------------------------------   F(2, 19383)     =   2219.86
       Model |   35570.262         2   17785.131   Prob > F        =    0.0000
    Residual |  155293.552    19,383  8.01184298   R-squared       =    0.1864
-------------+----------------------------------   Adj R-squared   =    0.1863
       Total |  190863.814    19,385   9.8459538   Root MSE        =    2.8305

------------------------------------------------------------------------------
        lexp |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .0692295   .0011701    59.16   0.000     .0669359    .0715231
      female |   1.158043    .040891    28.32   0.000     1.077893    1.238193
       _cons |   2.176568   .0603543    36.06   0.000     2.058268    2.294868
------------------------------------------------------------------------------
```

Compare to MLE version

```
mlexp (ln(normalden(lexp, {xb: age female _cons} , {sigma})))

initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -1539127.9
rescale:       log likelihood =  -52965.16
rescale eq:    log likelihood =  -52965.16
Iteration 0:   log likelihood =  -52965.16
Iteration 1:   log likelihood = -48279.366
Iteration 2:   log likelihood = -47689.334
Iteration 3:   log likelihood = -47676.448
Iteration 4:   log likelihood = -47676.408
Iteration 5:   log likelihood = -47676.408

Maximum likelihood estimation

Log likelihood = -47676.408                     Number of obs     =     19,386

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
xb           |
         age |   .0692295   .0011701    59.17   0.000     .0669362    .0715227
      female |   1.158043   .0408878    28.32   0.000     1.077905    1.238182
       _cons |   2.176568   .0603497    36.07   0.000     2.058285    2.294851
-------------+----------------------------------------------------------------
      /sigma |   2.830301   .0143739   196.91   0.000     2.802129    2.858473
------------------------------------------------------------------------------
```

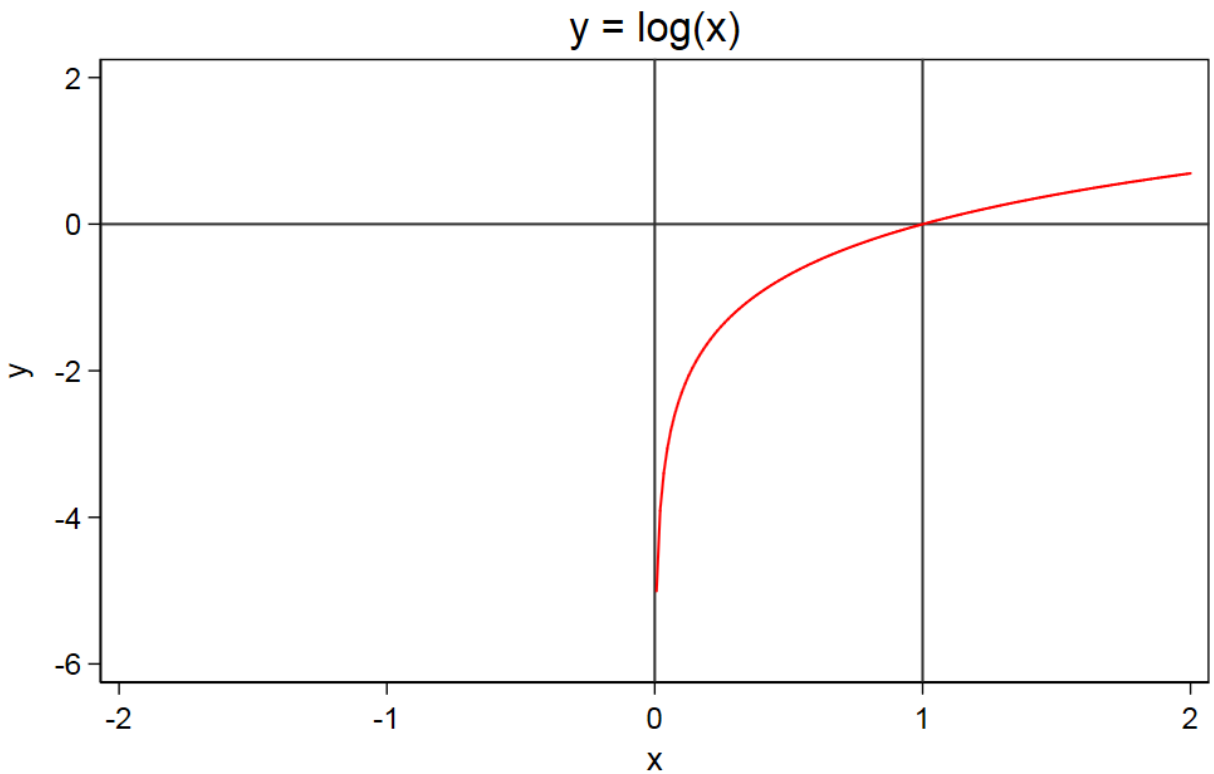
Numerical methods

- 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 the statistical significance -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

```
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 do we get using 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:
- $f(y; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i-\mu)^2}{2\sigma^2}\right)$
- 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:
- $f(y; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i-(\beta_0+\beta_1 x_i))^2}{2\sigma^2}\right)$
- 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. It's a different of deriving the simple linear model

What do 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**
- 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 do 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 (although you need to use the test command with the robust option in reg)
- 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

```
use "GPA1.dta", clear
rename colGPA colgpa
rename hsGPA hsgpa

quietly {
   reg colgpa
   est sto m1
   reg colgpa hsgpa
   est sto m2
   reg colgpa hsgpa skipped
est sto m3
}
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
```

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)

Information criteria to compare models

- A common way to compare models is by using the so-called **information criterion**. It's a measure that balances bias and variance or accuracy (fit) and simplicity (parsimony)
- The first one proposed is the Akaike Information Criterion (AIC). In the **context of the linear model** (estimated using OLS), it is:

  $AIC_p = n * ln(\frac{SSE_p}{n}) + 2 * p$
- p is the number of estimated parameters (including the constant), n is the number of observations, and SSE is the residual sum of squares (recall SST = SSR + SSE)
- **The smaller the AIC the better**. When is a model going to be better? The larger the sample size, the lower the unexplained variance, the fewer parameters we use
- Usually, we compare models with the same sample size, n, or maybe a small difference in n (a covarite may have some missings)

Log-likelihood and AIC

- That $2 * p$ is a **penalty** much like in adjusted R^2. The more parameters we estimate the better the model has to be to justify the additional estimated parameters
- We just saw the formula linking the log-likehood to SSE:

  $ll = -0.5 * n * [ln(2 * \pi) + ln(SEE/n] + 1$
- We could solve for SSE and plug in to get AIC as a function of the log-likelihood
- That's one one way to get some intuition on AIC

Information theory

- AIC comes from information theory. Suppose that we know the population process that generated the data but we are using a model to represent that process
- Think of simulating some data and then using imperfect models to represent the data. There will be some information lost
- If we knew the true data generating process, we could find a way to estimate how much information we are losing in each model and thus choose the model that doesn't lose as much information
- **But we never know the true model/process**. Akaike, in 1974, showed that his information criterion, now called AIC, tells how much more information is lost comparing one model to the other using the estimated log-likelihood

Some caveats

1. Valid **asymptotically**
2. Balance between goodness of fit and parsimony (simplicity). The penalty is incurred to avoid overfitting models because models with more parameters will always fit the data better. If you use as many parameters as data points, model fit is perfect (essentially, the estimated model is a perfect representation of the data)
3. **Most important**: AIC is a **relative** measure that compares one model to another to choose the one that loses less information. It's not a measure of how good a model is. You may have two bad fitting models (say, super low R^2). AIC helps you choose the best among terrible options

AIC and F-tests

- Note that AIC is similar to the F-test we used to compare nested models. One way of writing the F-test was
  $F = \frac{[SSE(RM) - SSE(FM)]/(p+1-k)}{SSE(FM)/(n-p-1)}$
- Also, the F-test can be written in terms of the R^2: $F = \frac{(R_p^2 - R_q^2)/(p-q)}{(1-R^2 p)/(n-p-1)}$
- We are essentially comparing SSE (or R^2) in relationship to the number of estimated parameters
- AIC is helpful because we can use it to **compare non-nested models**, although there is no statistical test like F test or MLE
- **AIC is a relative measure**. The absolute value doesn't tell us much; we use it **compare related models**
- Obviously, it wouldn't make much sense to compare models with different outcomes or different datasets

AIC with log-likelihood

- We also saw that we can use the likelihood ratio test to compare models and that the F-test and the LRT are asymptotically equivalent
- Stata's way to calculate AIC (except in GLM models) is $AIC = -2ll + 2k$
- k is the number of parameters, ll is the log-likelihood function. $2k$ is the penalty due to the number of parameters; the more parameters, the higher AIC must be to compensate
- Maybe the best way to summarize is that we are essentially comparing the log-likelihood of related models

Bayesian Information Criterion (BIC)

- The more commonly used information criterion is BIC:

  $BIC = -2ll + ln(n) * k$
- Note that we now add $ln(n)$ in the penalty term rather than 2
- If n = 8, ln(8) = 2.079. So in most applications, the BIC penalty is always **higher than the AIC penalty** but it has a limit because the log function "flattens"
- ln(50) = 3.91, ln(100) = 4.60, ln(500) = 6.21, ln(1000) = 6.91, ln(2000) = 7.6
- The more observations we have, the lower the growth of the penalty $ln(n) * k$

Logit and probit: Latent variable approach

- There is yet another way to derive logit and probit models. A way that connects both into the same framework
- Suppose that there is a **latent** (unobserved) and **continuous** variable y^* that take values from $-\infty$ to $+\infty$. We assume that the latent variable is a function of covariates **X**. For simplicity, let's just assume a linear relationship and just one covariate: $y_i^* = \beta_0 + \beta_1 x_i + u$
- u plays the same role as ϵ in the linear model: a source of random error
- We do not observe the latent variable $y*$, we only observe if an event happens or not but whether the event happens depends on the value of the latent variable. We use y_i to denote the **observed** variable, which we assume is coded as 1 or 0
- If $y_i* > 0$ then $y_i = 1$. If $y_i* \leq 0$ then $y_i = 0$. Note that in this case 0 is a **threshold**
- Think of $y*$ as intelligence and y is whether a person answers a question correctly or not. Or think of the latent variable a measure of disease severity. If it crosses 0 the person dies

Latent variable approach

- Because of the way we set up the problem, we can write the probability of $y = 1$ conditional on the covariate x as:

  $P(y = 1|x) = P(y* > 0|x)$
- Since we assumed that $y_i^* = \beta_0 + \beta_1 x_i + u$ the above equation becomes

  $P(y = 1|x) = P(\beta_0 + \beta_1 x_i + u > 0|x) = P(u < [\beta_0 + \beta_1 x_i]|X) = F([\beta_0 + \beta_1 x_i]|x)$
- So the probability of **observing** $y = 1$ depends on the distribution of u, the error component of the *latent* variable $y*$. We could calculate that probability if we knew the **cumulative distribution function** $F()$
- It's obvious that we need to make a strong assumption about the distribution of $y*$ in order to calculate $P(y = 1|x)$ (note that $P(y = 0|x) = 1 - P(y = 1|x)$)

Latent variable approach

- In econometrics this type of model is called **index function models** (in this case a single index)
- Two common assumptions about u are used: u distributes either **standard logistic** or **standard normal**
- Both distributions have a mean of 0 and constant variance. In the standard logistic the variance $var(u) = \frac{\pi^2}{3}$. In the standard normal $var(u) = 1$
- The idea of fixing the variance is not that trivial in the sense that if we don't fix it, then we can't estimate it because we only observe a 0 or 1 and the probability of 1 depends on the sign on $y*$ but not the scale (variance)
- In other words, we don't have information to estimate $var(u)$ yet **we lose nothing** by fixing it because $P(y = 1|x)$does not depend on $var(u)$

Latent variable approach

- Remember that the **cumulative distribution function** (cdf) gives you $P(X < a)$. Remember too that to calculate the probability we need to integrate the density f(t) from $-\infty$ to a: $\int_{-\infty}^{a} f(t)dt$
- If we assume **standard normal cdf**, our model becomes $P(y = 1|x) = \int_{-\infty}^{\beta_0+\beta_1 x} \frac{1}{2\pi} e^{(-\frac{t^2}{2})} dt$
- And that's the **probit** model. Note that because we use the cdf, the probability will obviously be constrained between 0 and 1 because since it's a cdf
- If we assume that u distributes **standard logistic** then our model becomes $P(y = 1|x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$, which we saw before as the logit transformation
- Remember that there are two different concepts: **logistic response function** and **logistic distribution**. The **standard logistic cdf** happens to have the above formula (the pdf is different)

Estimation

- Estimation is straightforward with MLE. We did it for the logistic model already
- For probit, the likelihood is just like writing $P(y = 1|x)$ above because that's the probability of seeing the data. We need to multiply n times and also consider that the probability of 0 is $1 - P(y = 1|x)$. If we take the log, it's a sum
- This is often a source of confusion but remember that the likelihood function is the probability of seeing the data given assumptions about the distribution of the data
- So what is the probability of observing a data point $y = 1$? It's $P(y = 1|x) = \int_{-\infty}^{\beta_0+\beta_1 x} \frac{1}{2\pi} e^{(-\frac{t^2}{2})} dt$
- What is the probability of observing a data point $y = 0$? It's $P(y = 0|x) = 1 - \int_{-\infty}^{\beta_0+\beta_1 x} \frac{1}{2\pi} e^{(-\frac{t^2}{2})} dt$
- For the logit, we would do the same, but we would arrive to the same place as before

Estimation

- We can program the probit model easily with Stata
- I have more examples on my site `https://clas.ucdenver.edu/marcelo-perraillon/code-and-topics`
- Note below that writing the likelihood makes it obvious that the betas are shifts in the standard normal cdf scale

```
program probit_lf
        version 12
        args todo b lnf
        tempvar xb lj
        mleval `xb' = `b'
        * latent variable assumed cumm standard normal
        qui gen double `lj'  = normal( `xb')  if $ML_y1 == 1
        qui replace    `lj'  = normal(-`xb')  if $ML_y1 == 0
        qui mlsum `lnf' = ln(`lj')
end
```

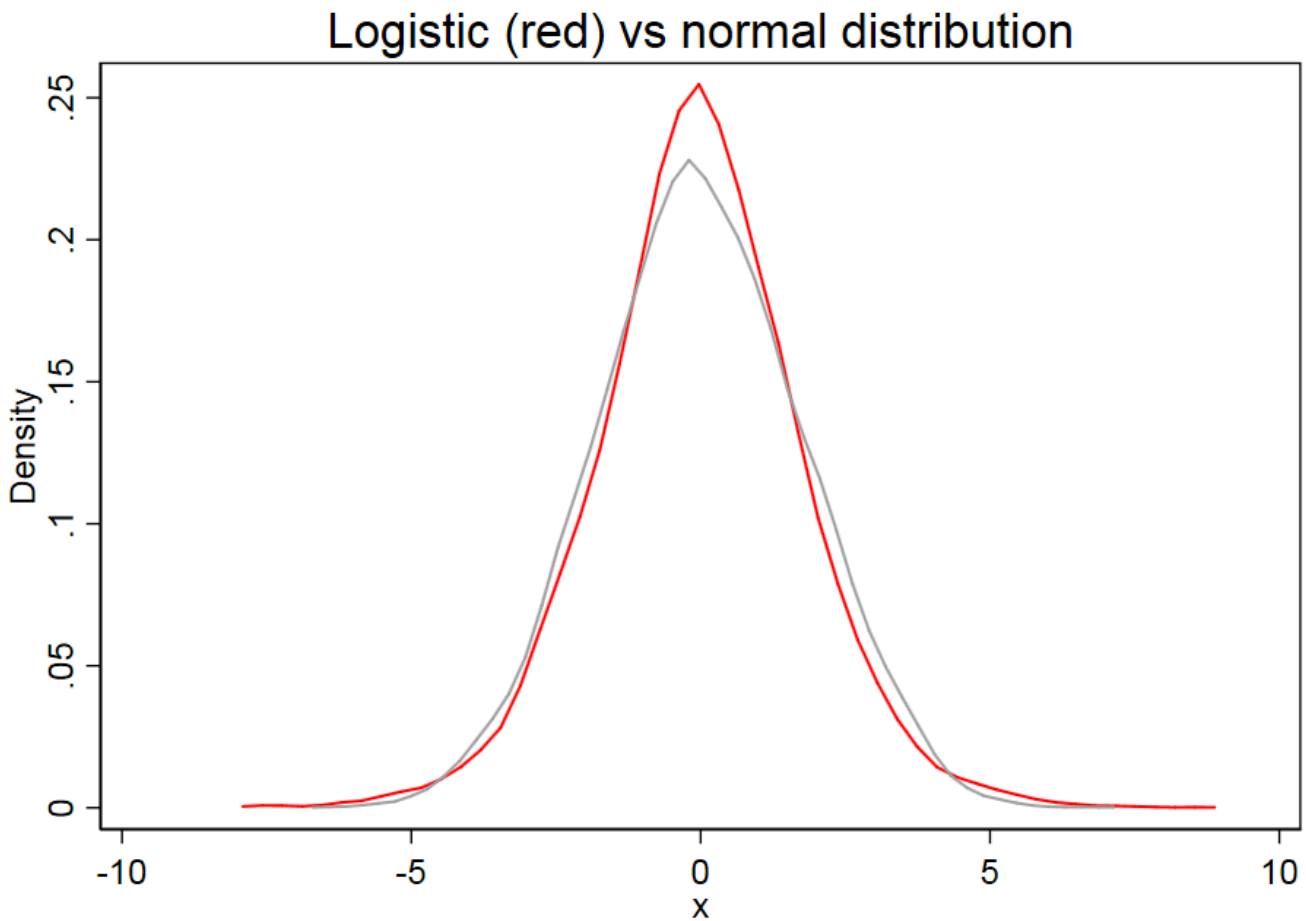
Standard logistic vs standard normal

- Does it make much of a difference if we use one vs the other?

```
clear
set seed 123456
set obs 5000
gen u = uniform()

* Simulate logistic distribution
gen l = -ln((1 - u)/u)
sum l

* Simulated normal with same parameters
gen n = rnormal(r(mean), r(sd))

* Plot
kdensity l, bw(0.3) gen(xl dl)
kdensity n, bw(0.3) gen(xn dn)
line dl xl, sort color(red) || line dn xn, sort ///
   title("Logistic (red) vs normal distribution") ytitle("Density") ///
   xtitle("x") legend(off)
graph export logvsnorm.png, replace
```

Standard logistic vs standard normal normal

- Assuming either one as the latent distributions makes little difference



Digression

- Assuming standard normal cdf or logistic are not the only options
- There is the complementary log-log model commonly used in discrete time survival because the exponent of coefficients are hazard rates
- Or the Gumbel model used to model extreme values
- Or the Burr model. Or the Scobit model
- Statistics and econometrics are large fields... Papers must be written, dissertations must be completed
- Sometimes a proposed new method goes to the Journal Article Graveyard. Sometimes they are resurrected 30 years later when somebody discovers that they are perfect for a particular application
- So many ideas and clever people out there. See Greene (2018) for more details

Example

- Women's labor force participation (inlf); main predictor is "extra" money in family

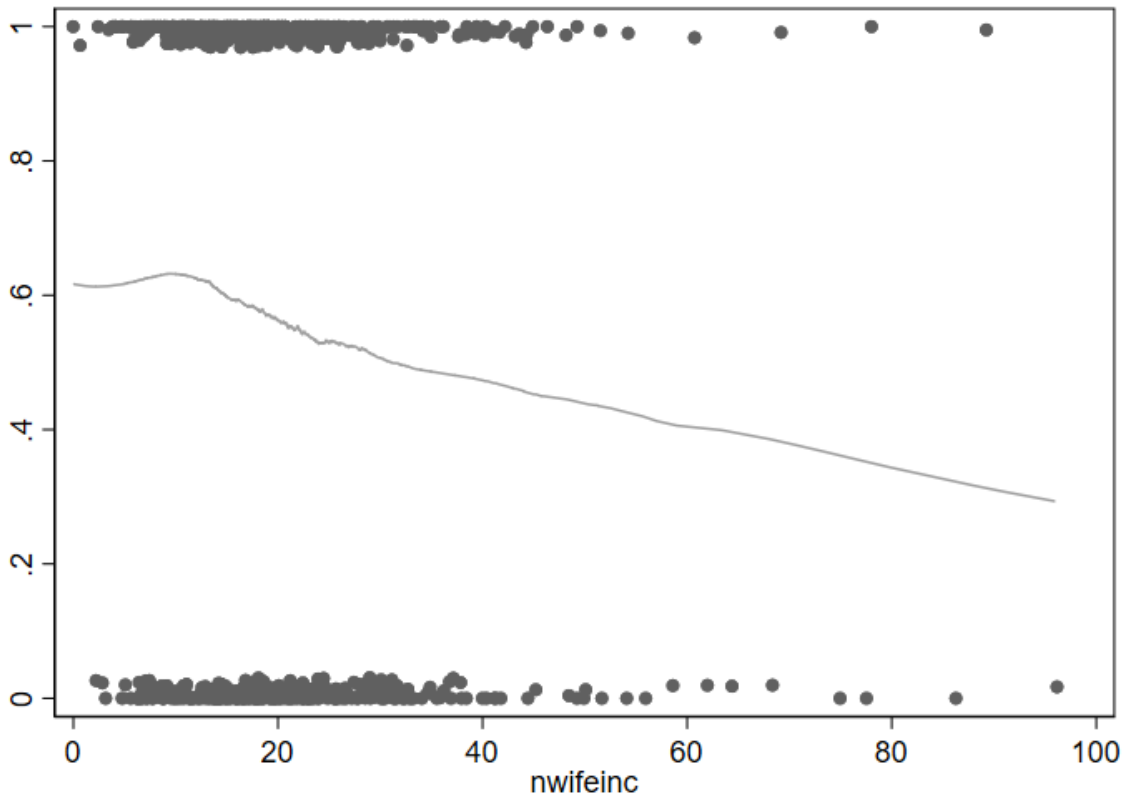
```
bcuse mroz, clear

inlf      =1 if in labor force, 1975
nwifeinc  (faminc - wage*hours)/1000
educ      years of schooling
exper     actual labor mkt exper
age       woman's age in yrs
kidslt6   # kids < 6 years
kidsge6   # kids 6-18
```

Variable	Obs	Mean	Std. Dev.	Min	Max
inlf	753	.5683931	.4956295	0	1
nwifeinc	753	20.12896	11.6348	-.0290575	96
educ	753	12.28685	2.280246	5	17
exper	753	10.63081	8.06913	0	45
age	753	42.53785	8.072574	30	60
kidslt6	753	.2377158	.523959	0	3
kidsge6	753	1.353254	1.319874	0	8

Labor force participation

- The probability of working is decreasing as a function of "extra" income

```
lowess inlf nwifeinc, gen(lflow) nograph
scatter inlf nwifeinc, jitter(5) msize(small) || line lflow nwifeinc, sort ///
        legend(off) saving(lblow.gph, replace)
graph export lblow.png, replace
```



Writing down the model

- We want to estimate the following model:

  $P(inlf_i = 1|nwifeinc_i) = \Lambda(\beta_0 + \beta_1 nwifeinc_i)$
- **By convention** (in economics and health economics), when we write capital lambda, $\Lambda()$, we imply a logistic model (Λ is not a non-linear function). When we write phi, $\phi()$, we imply a probit model
- Write the logistic model this way (no error term!):

  $log(\frac{inlf_i}{1-inlf_i}) = \beta_0 + \beta_1 nwifeinc_i$
- Or

  $logit(inlf_i) = \beta_0 + \beta_1 nwifeinc_i$
- Again, write it like this: $log(\frac{inlf_i}{1-inlf_i}) = \beta_0 + \beta_1 nwifeinc_i$ because this will **match Stata's (or any other statistical package) output**. Remember, we are not directly estimating $P(inlf_i = 1|nwifeinc_i)$

Estimating the model

- So, we will estimate $log(\frac{inlf_i}{1-inlf_i}) = \beta_0 + \beta_1 nwifeinc_i$

```
 logit inlf nwifeinc, nolog

Logistic regression                             Number of obs     =        753
                                                LR chi2(1)        =      10.44
                                                Prob > chi2       =     0.0012
Log likelihood = -509.65435                     Pseudo R2         =     0.0101

------------------------------------------------------------------------------
        inlf |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    nwifeinc |  -.0207569   .0065907    -3.15   0.002    -.0336744   -.0078394
       _cons |   .6946059   .1521569     4.57   0.000      .396384    .9928279
------------------------------------------------------------------------------
```

- A one thousand increase in "extra" income decreases the log-odds of participating in the labor force by 0.021. And it's statistically significant (p-value $= 0.002$). Same Wald test as before: $-.0207569/.0065907 = -3.1494227$. The difference is that the it's not t-student distributed but normally distributed

Overall significance

- The χ^2 (chi-square) test of the overall significance should look familiar. It compares the current model to the null model (without covariates); the null hypothesis is that all the cofficients in current model are zero
- It's the **likelihood ratio test** that we have seen before; the equivalent of ANOVA:

```
* LRT
qui logit inlf nwifeinc, nolog
est sto full

qui logit inlf, nolog
est sto redu

lrtest full redu

Likelihood-ratio test                           LR chi2(1)  =     10.44
(Assumption: redu nested in full)               Prob > chi2 =    0.0012
```

What about that Pseudo R^2?

- We can't partition the variance into explained and unexplained as before so we don't have a nice R^2 that goes from 0 to 1
- But one way to come up with a measure of fit is to use the (log) likelihood function to compare the **current model** to the **model without any explanatory variable** (the null model)
- The formula is: $1 - \frac{ll_{cm}}{ll_{nul}}$, where ll_{cm} is the log-likelihood of the current model and ll_{nul} is the log-likelihood of the null model
- If the current model is as good as the null model, then $\frac{ll_{cm}}{ll_{nul}}$ is going to close to 1 and the $pseudo - R^2$ is going to be close to zero
- In other words, adding variables doesn't improve the likelihood. If adding variables improves the likelihood, then the pseudo R^2 will be greater than zero

Pseudo-R^2

- Replicate Pseudo R^2

```
qui logit inlf nwifeinc, nolog
scalar ll_cm = e(ll)

qui logit inlf, nolog
scalar ll_n = e(ll)

di 1 - (ll_cm/ll_n)

.0101362

di  "cm: " ll_cm  " "  "null: "  ll_n  "  " "(ll_cm/ll_n): "  (ll_cm/ll_n)
cm: -509.65435 null: -514.8732  (ll_cm/ll_n): .9898638
```

- Psuedo R^2 is **not** a measure of how good the model is at prediction; just how better it fits compared to null model. I don't think that calling it pseudo R^2 is a good idea
- **Big picture**: comparing the log-likelihood of models is a way of comparing goodness of fit. If nested, we have the a test (LRT); if not nested, we have BIC or AIC

Not the only pseudo R^2?

- Stata uses one version of pseudo R^2 but there are plenty more. Other software may use different metrics
- Long and Freese (2014) have a laundry list of different pseudo R^2 (it's an excellent book, by the way)
- There is the McFadden one, MLE, Cragg and Uhler (also known as Nagelkerke), Efron's, Tjur's... (page 127)
- In any case, none of them have the same meaning as the R^2 in linear regression
- In particular, they **don't mean that predictions are good**. Recall that in linear regression the R^2 is also the square of the correlation between observed and predicted values
- **See, context matters a lot**

Let's try a different predictor

- We will estimate $log(\frac{inlf_i}{1-inlf_i}) = \beta_0 + \beta_1 hsp_i$, where *hsp* if education > 12

```
gen hsp = 0
replace hsp = 1 if educ > 12 & educ ~= .

logit inlf hsp, nolog

Logistic regression                             Number of obs     =        753
                                                LR chi2(1)        =      15.08
                                                Prob > chi2       =     0.0001
Log likelihood = -507.33524                     Pseudo R2         =     0.0146

------------------------------------------------------------------------------
        inlf |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         hsp |   .6504074   .1704773     3.82   0.000     .3162781    .9845368
       _cons |   .0998982    .086094     1.16   0.246     -.068843    .2686393
------------------------------------------------------------------------------
```

- The log-odds of entering the labor force is 0.65 higher for those with more than high school education compared to those with high-school completed or less than high-school

Odds ratios

- Let's do our usual math to make sense of coefficients. We just estimated the model $log(\frac{inlf_i}{1-inlf_i}) = \beta_0 + \beta_1 hsp_i$
- For those with $hsp = 1$, the model is $log(\frac{inlf_{hsp}}{1-inlf_{hsp}}) = \beta_0 + \beta_1$
- For those with $hsp = 0$, the model is $log(\frac{inlf_{nohsp}}{1-inlf_{nohsp}}) = \beta_0$
- The difference of the two is $log(\frac{inlf_{hsp}}{1-inlf_{hsp}}) - log(\frac{inlf_{nohsp}}{1-inlf_{nohsp}}) = \beta_1$
- Applying the rules of logs: $log(\frac{\frac{inlf_{hsp}}{1-inlf_{hsp}}}{\frac{inlf_{nohsp}}{1-inlf_{nohsp}}}) = \beta_1$
- Taking $e^{()}$: $\frac{\frac{inlf_{hsp}}{1-inlf_{hsp}}}{\frac{inlf_{nohsp}}{1-inlf_{nohsp}}} = e^{\beta_1}$

Odds ratios

$$\frac{\frac{inlf_{hsp}}{1-inlf_{hsp}}}{\frac{inlf_{nohsp}}{1-inlf_{nohsp}}} = e^{\beta_1}$$

- And that's the **(in)famous odds-ratio**
- In our example, $e^{.6504074} = 1.92$. So the odds of entering the labor force is almost twice as high for those with more than high school education compare to those without
- That's the way careful reporters would report this finding. **And it's correct**. The problem is that we would then interpret this as saying that the **probability** of entering the labor force is twice as high for those with more than high school
- **That interpretation is wrong**. A ratio of odds is more often than not far away from the ratio of probabilities

Predictions in the probability scale - logit and probit

- We can easily calculate predictions in the **probability scale**
- Remember, we are modeling $log(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \cdots + \beta_j X_j$
- But we can solve for p:

  $p(\mathbf{X}, \beta) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_j X_j}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_j X_j}} = \frac{1}{1+e^{-}\beta_0+\beta_1 X_1+\cdots+\beta_j X_j}$
- That's what Stata does by default when using the postestimation command predict
- Predictions after probit can be a bit confusing, but remember that the coefficients are shifts in the cumulative standard normal (they are z scores), so to calculate $p(x, \beta)$ we need to calculate a probability given the z score

Calculating predictions after logit and probit models

- Make sure you understanding this. Think of predictions after probit as z scores that need to be converted into a probability

```
logit inlf hsp, nolog

* Predictions for logit manually
gen phat_manu = 1/(1+exp(-(_b[_cons] +_b[hsp]*hsp)))
*Same as using the inverse logit function
gen phat_invl = invlogit(_b[_cons] +_b[hsp]*hsp)
* Same as default of predict command
predict phat_pred

sum phat_manu phat_invl phat_pred
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
   phat_manu |        753    .5683931    .0694388   .5249538   .6792453
   phat_invl |        753    .5683931    .0694388   .5249538   .6792453
   phat_pred |        753    .5683931    .0694388   .5249538   .6792453

*probit
qui probit inlf hsp, nolog

* use inverse normal
gen phat1_norm = normal(_b[_cons] + _b[hsp]*hsp)
predict  phat1_predprob
predict zscore, xb
sum phat1* zscore
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
  phat1_norm |        753    .5683931    .0694388   .5249538   .6792453
phat1_pred~b |        753    .5683931    .0694388   .5249538   .6792453
      zscore |        753    .1760512    .1813693   .0625907   .4655894

di normal(0.1760512)
* .5683
```