Maximum Likelihood Estimation and Binary Dependent Variables

1. Starting with a Simple Example: Bernoulli Trials

Lets start with a simple example:

Teams A and B play one another 10 times; A wins 4 of the games. The games are played under similar circumstances, and the outcomes of each game are independent. You are interested in estimating the (unknown) probability, p, that A wins a game.

You don't know the value of p, but you do know that whatever it is, it is in some sense consistent with the outcomes you observe. So the challenge is to use the outcomes information to come up with your best estimate of the value of p that is behind all this.

Here's the data:

Game	**Outcome for A**	**Prob of that outcome: *likelihood***	***lnlikelihood***
1	Win	p	ln(p)
2	Loss	1-p	ln(1-p)
3	Loss	1-p	ln(1-p)
4	Loss	1-p	ln(1-p)
5	Win	p	ln(p)
6	Win	p	ln(p)
7	Loss	1-p	ln(1-p)
8	Loss	1-p	ln(1-p)
9	Win	p	ln(p)
10	Loss	1-p	ln(1-p)
		Prob Outcomes:	***Sum of lnlikelihoods:***
		$p^4(1-p)^6$	$4\ln(p)+6\ln(1-p)$

Since the outcomes of the games are independent in the probabilistic sense, the probability of seeing this particular sequence of wins and losses for A is:
$p(1-p)(1-p)(1-p)pp(1-p)(1-p)p(1-p) = p^4(1-p)^6$.

This is the ***likelihood*** of observing the sequence of wins and losses, conditional on the unknown parameter *p*.

With ***Maximum Likelihood Estimation*** (**MLE**) we estimate *p* by answering the question: Which parameter value p generates the greatest likelihood of the observed outcomes… or more formally:

$$\max_p likelihood(p) \equiv \max_p p^4(1-p)^6 \rightarrow p^*.$$

To solve for p^*, it is usually easier to first take the ln(.) of the likelihood. Since the ln(.) function is strictly increasing if p^* maximizes the likelihood, it will also maximize the ***lnlikelihood***. So we can restate the problem:

$$\max_p \ln\left[likelihood(p)\right] \equiv \max_p \ln\left[p^4(1-p)^6\right] = \max_p 4\ln(p) + 6\ln(1-p) \rightarrow p^*.$$

We can use the ***First Order Condition*** (***FOC***) to solve for p^*:

FOC: $\frac{4}{p^*} - \frac{6}{1-p^*} = 0 \leftrightarrow p^* = \frac{4}{10} = .40 = \%wins$

The ***Second Order Condition*** is satisfied since:

SOC: $\frac{d^2}{dp^2}\ln\left[likelihood(p)\right] = -\frac{4}{p^2} - \frac{6}{(1-p)^2} < 0 \; for \; p > 0$.

So we have: A's %wins is an MLE of the probability that A will beat B.

%Wins as an MLE

More generally: Suppose that teams A and B play one another n times; A wins k of the games. (Unlike the above, you don't know the sequence of wins and losses, you just know that there were k wins in n games.).

As above, the games are played under similar circumstances, and the outcomes of each game are independent. And you want to estimate the probability, p, that A wins a game.

Since each game has two outcomes, A wins (with probability *p*) or B wins, with probability *1-p*, the probability that A wins k out of n games is given by the Binomial distribution:

$$P(k \; wins \; in \; n \; games) = \binom{n}{k} p^k (1-p)^{n-k}$$

This is also the likelihood that A wins k out of n games, and so the likelihood function is:

$$L(k,n \mid p) == \binom{n}{k} p^k (1-p)^{n-k}.$$

MLE and Binary Dependent Variables

As discussed above. with Maximum Likelihood Estimation (MLE), the goal is to find the value of the parameter value *p* that makes most likely what was observed, namely k wins in n games. And so the MLE estimate of *p*, p^*, will be defined by:

$$\max_p L(k,n \mid p) \to p^* .$$

To solve this optimization problem, first take ln(.)'s as above and then use first and second order conditions (FOC and SOC) to find the p^* that maximizes the *lnlikelihood*:

$$\max \ln\left[L(k,n \mid p)\right] = \max \ln\left[\binom{n}{k}\right] + k\ln(p) + (n-k)\ln(1-p) .$$

$$FOC: \frac{k}{p^*} - \frac{n-k}{(1-p^*)} = 0 \to p^* = \frac{k}{n} = \%wins$$

$$SOC: -\frac{k}{\left(p^*\right)^2} - \frac{n-k}{(1-p^*)^2} < 0 \to \text{ maximum (since SOC is everywhere < 0)}$$

We have the same result as above, that the win percent is an MLE estimator of p, the probability of a win.

2. MLE More Generally

Suppose that

$$L(y_1, y_2, \ldots, y_n \mid \beta_1, \beta_2, \ldots, \beta_k)$$

is the probability (or likelihood) of observing the outcomes $\{y_i\}$… given parameters $\{\beta_j\}$. As the betas change, the likelihood of getting the outcomes that actually occurred will change. The MLE estimates of those unknown parameters will be the set of parameter values that would have with the greatest probability predicted the sample outcomes that were actually observed … so the estimation problem is:

$$\max_{\{\beta_j\}} L\left[\{y_i\} \mid \{\beta_j\}\right] \to \{\beta_j^*\} .$$

Typically, it is convenient to take the ln of the likelihood function to solve the optimization problem… since whatever maximizes L() will also maximize lnL():

$$\max_{\{\beta_j\}} \ln\left\{L\left[\{y_i\} \mid \{\beta_j\}\right]\right\} \to \{\beta_j^*\} .$$

In this case, the maximand is the called the *lnlikelihood* (or *log-likelihood*) function. Since the overall likelihood function is often the product of a number of individual likelihood functions (as, say, in the case of independent observations), by taking ln's, we convert a messy multiplicative function into a much easier to evaluate additive function. And so standard practice is in fact to take ln's and maximize lnlikelihoods.

3. SLR Models and Maximum Likelihood Estimation

Consider the standard Simple Linear Regression (SLR) model:

SLR: $Y_i = \beta_0 + \beta_1 x_i + U_i$

Since $U_i \sim N(0, \sigma^2)$, conditional on x_i, we have $Y_i \mid x_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2)$

The conditional density function for Y_i is defined by:

$$f(y_i \mid x_i, \beta_0, \beta_1) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2 \right)$$

And so the Likelihood function is:

$$L\left(\{y_i\} \mid \{x_i\}, \beta_0, \beta_1\right) = \prod f(y_i \mid x_i, \beta_0, \beta_1) \;= \prod \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-1}{2\sigma^2}(y_i - \beta_0 - \beta_1 x_i)^2 \right)$$

and Log Likelihood: $\ln\left(L(.)\right) = n \ln\left(\frac{1}{\sigma\sqrt{2\pi}} \right) - \frac{1}{2\sigma^2} \sum (y_i - \beta_0 - \beta_1 x_i)^2$

Since $n \ln\left(\frac{1}{\sigma\sqrt{2\pi}} \right)$ is independent of the parameters to be estimated, β_0 *and* β_1, and since $-\frac{1}{2\sigma^2}$ is a negative scalar,

$$\max_{\beta_0, \beta_1} \; \left\{ n \ln\left(\frac{1}{\sigma\sqrt{2\pi}} \right) - \frac{1}{2\sigma^2} \sum (y_i - \beta_0 - \beta_1 x_i)^2 \right\} \Leftrightarrow \min_{\beta_0, \beta_1} \; \left\{ \sum (y_i - \beta_0 - \beta_1 x_i)^2 \right\}$$

Note that to maximize the *lnlikelihood* expression is to minimize
$SSR = \sum (y_i - \beta_0 - \beta_1 x_i)^2$.

So the MLE estimates of β_0 *and* β_1 $(\hat{\beta}_0$ *and* $\hat{\beta}_1)$ will also be the parameter values that minimize SSRs. Or put differently: For the standard SLR model, the OLS estimates and the MLE estimates of the intercept and slope are one and the same.

4. An Example: Predicting NFL Game Winners

In this example, we'll build some models that predict NFL game winners as a function of various game stats, including rushing yards, passing yards, and time of possession.

The data come from Warren Repole's website, www.repole.com, and cover every regular season NFL game from 2002 through the 2013 season. Here are the summary stats for the numeric variables in the dataset (the Home team is "off" and the Visitors are "def"):

Variable	Obs	Mean	Std. Dev.	Min	Max
season	3081	2007.509	3.452648	2002	2013
scoreoff	3081	23.04122	10.34521	0	62
firstdownoff	3076	19.36671	4.933257	3	40
rushattoff	3076	28.25683	7.965681	7	60
rushydsoff	3076	119.5966	52.61298	-3	378
passattoff	3076	32.99675	8.418214	9	65
passcompoff	3076	20.06762	5.941666	5	43
passydsoff	3076	226.3495	75.88289	22	510
passintoff	3076	.9518856	1.029465	0	6
fumblesoff	3076	.6862809	.8268386	-1	4
sacknumoff	1289	2.252133	1.669757	-1	11
sackydsoff	3076	13.95026	11.96639	-13	71
penydsoff	3076	50.39792	25.40716	0	175
puntavgoff	3069	43.25506	6.131968	0	69
scoredef	3081	20.42486	10.23999	0	59
firstdowndef	3076	18.41255	5.060527	3	37
rushattdef	3076	26.96912	8.023747	6	57
rushydsdef	3076	110.975	51.63911	-18	351
passattdef	3076	33.48505	8.583257	6	67
passcompdef	3076	20.01918	6.156492	1	43
passydsdef	3076	222.0878	78.37646	-7	516
passintdef	3076	1.044213	1.065648	0	6
fumblesdef	3076	.6989597	.8445399	0	5
sacknumdef	1289	2.349884	1.717919	-1	10
sackydsdef	3076	15.06762	12.55878	-7	79
penydsdef	3076	53.3407	26.62058	0	177
line	3081	2.457157	6.004107	-18.5	26.5
totalline	3081	42.48588	4.754155	30	60

For each game, and for both teams, we have the following data: game date, teams, final score, 1st down yardage, 3rd down conversion rates, rushing and passing attempts and yards, interceptions, fumbles, sacks and sack yards, penalty yards, average punt yards, and time of possession. In addition, we have venue information and point spreads and total (OU) lines.

Let's start by predicting the probability of a home team victory as a function of the difference in total rushing and passing yards: **netyds** = **netrush** + **netpass**, where netrush = HomeRushYds – VisRushYds = rushydsoff – rushydsdef, and likewise for netpass.

MLE and Binary Dependent Variables

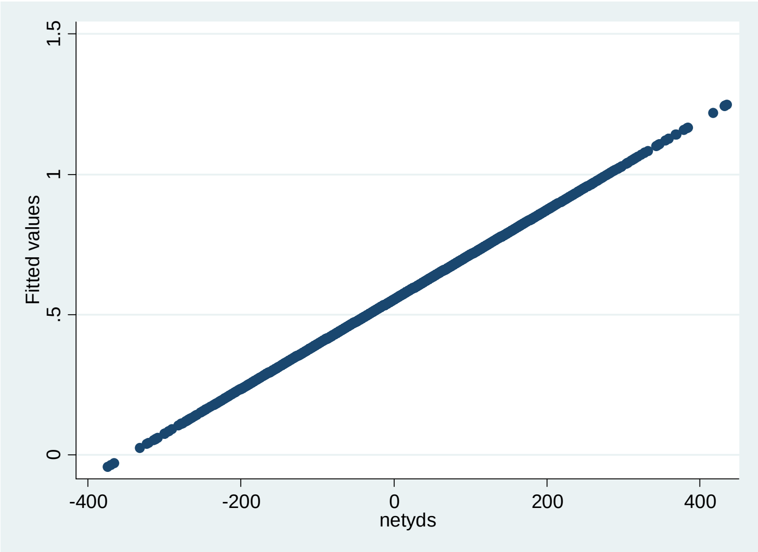
Take I: OLS - Linear Probability Models

The first estimated model is a standard Linear Probability Model… named because the estimated probabilities are linear in the explanatory variables:

```
. reg hwins netyds

      Source |       SS       df       MS              Number of obs =    3076
-------------+------------------------------           F(  1,  3074) =  540.19
       Model |  112.367696     1  112.367696           Prob > F      =  0.0000
    Residual |  639.434644  3074  .208013873           R-squared     =  0.1495
-------------+------------------------------           Adj R-squared =  0.1492
       Total |  751.802341  3075  .244488566           Root MSE      =  .45609

------------------------------------------------------------------------------
       hwins |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      netyds |   .0015947   .0000686    23.24   0.000     .0014602    .0017292
       _cons |   .5542275   .0082708    67.01   0.000     .5380106    .5704443
------------------------------------------------------------------------------
```

netyds is highly statistically significant. The models estimates that on average, and controlling for nothing other than home field advantage, an increase in net yards of 100 increases the chance of a victory by 16% points. An attractive feature of the LPM is that you can read the estimated effects right off of the regression output.

Here are the predicted values from the model:

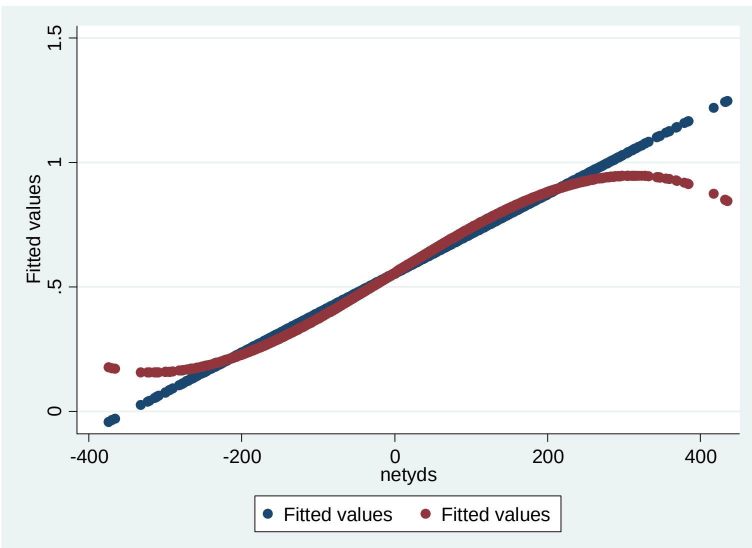


Right away you can see a problem. Some of the probabilities are below zero while others are well above 1. That’s a problem.

One way around this problem is to introduce second and third order terms:

```
. reg hwins netyds*

      Source |       SS       df       MS              Number of obs =    3076
-------------+------------------------------           F(  3,  3072) =  185.05
       Model |  115.068208     3  38.3560694           Prob > F      =  0.0000
    Residual |  636.734132  3072  .207270225           R-squared     =  0.1531
-------------+------------------------------           Adj R-squared =  0.1522
       Total |  751.802341  3075  .244488566           Root MSE      =  .45527

------------------------------------------------------------------------------
       hwins |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      netyds |    .001887   .0001073    17.59   0.000     .0016767    .0020974
     netyds2 |  -4.44e-08   4.24e-07    -0.10   0.917    -8.75e-07    7.87e-07
     netyds3 |  -6.35e-09   1.91e-09    -3.33   0.001    -1.01e-08   -2.61e-09
       _cons |   .5560969    .010094    55.09   0.000     .5363053    .5758885
------------------------------------------------------------------------------
```

Here are the predicted values from the two models:



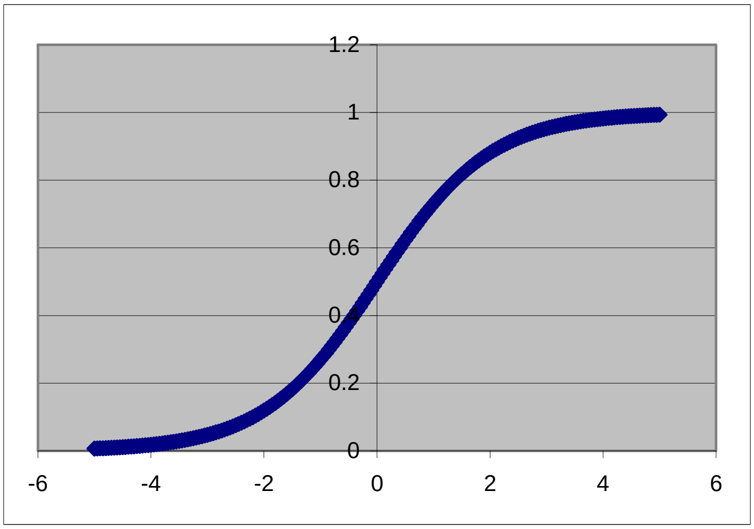
While the predicted probabilities are no longer outside the [0,1] interval, we now have a new issue: in the tails of the distribution, the predicted probabilities move counter to what is expected (increases in net yardage predict decreases in win probabilities).

Notice, however, that for the overwhelming majority of the observations, the predicted win probabilities for the two models are virtually the same.

MLE and Binary Dependent Variables

Take II: MLE - Logit

The logit model is based on the logistic function which is defined by $f(x) = \dfrac{1}{1+e^{-x}}$ and graphed below:[1]



It is a symmetric function ($f(-x) = 1 - f(x)$), bounded below by zero, increasing everywhere, and bounded above by one. In the logit specification,

$$P(Hwins \mid netyds) = \frac{1}{1+e^{-(\beta_0+\beta_1 netyds)}} .$$

Since $P(Hloses \mid netyds) = 1 - P(Hwins \mid netyds) = 1 - \dfrac{1}{1+e^{-(\beta_0+\beta_1 netyds)}} = \dfrac{e^{-(\beta_0+\beta_1 netyds)}}{1+e^{-(\beta_0+\beta_1 netyds)}}$, the ratio of the two probabilities is:

$$OddsRatio = \frac{P(Hwins \mid netyds)}{P(Hloses \mid netyds)} = e^{(\beta_0+\beta_1 netyds)} .$$

And so $\ln\left[OddsRatio\right] = \beta_0 + \beta_1 netyds$ … linear in the explanatory variable!

---

[1] The logistic and logit functions are inverses of one another. The logit function is defined by:

$$\text{logit}(p) = -\ln(\frac{1}{p} - 1) .$$

And so one way to think about the Logit Model is that it assumes that the lnOddsRatio is linear in the RHS variables… and in that way, the model looks a bit like the LPM.

To illustrate how MLE/logit estimation works, suppose that the Home team wins the first game, loses the second and wins the third, and that these results are independent, so that the probability of the three occurring is just the product of the probabilities of each one happening. Then the likelihood of these three outcomes (conditional on the different netyds) is:

$P(Hwins_1 \mid netyds_1) * P(Hloses_2 \mid netyds_2) * P(Hwins_3 \mid netyds_3)$, or

$$L\left(Hwins_1, Hloses_2, Hwins_3 \mid \beta_0, \beta_1\right) = \frac{1}{1+e^{-(\beta_0+\beta_1 netyds_1)}} \frac{e^{-(\beta_0+\beta_1 netyds_2)}}{1+e^{-(\beta_0+\beta_1 netyds_2)}} \frac{1}{1+e^{-(\beta_0+\beta_1 netyds_3)}}.$$

The MLE method will estimate the coefficient values by maximizing this expression with respect to β_0 & β_1. If we take the ln( ) of this expression, and use the fact that ln(1)=0, we get the log-likelihood function

$$-\ln(1+e^{-(\beta_0+\beta_1 netyds_1)}) - (\beta_0 + \beta_1 netyds_2) - \ln(1+e^{-(\beta_0+\beta_1 netyds_2)}) - \ln(1+e^{-(\beta_0+\beta_1 netyds_3)})$$

which is much easier to evaluate

```
. logit hwins netyds

Iteration 0:   log likelihood =  -2097.596
Iteration 1:   log likelihood = -1848.0407
Iteration 2:   log likelihood =  -1846.376
Iteration 3:   log likelihood = -1846.3746
Iteration 4:   log likelihood = -1846.3746

Logistic regression                               Number of obs   =       3076
                                                  LR chi2(1)      =     502.44
                                                  Prob > chi2     =     0.0000
Log likelihood = -1846.3746                       Pseudo R2       =     0.1198

------------------------------------------------------------------------------
       hwins |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      netyds |   .0078005   .0003924    19.88   0.000     .0070313    .0085697
       _cons |   .2618768   .0396232     6.61   0.000     .1842168    .3395369
------------------------------------------------------------------------------
```

Given these parameter estimates, the predicted probability that the home team wins is:

MLE and Binary Dependent Variables

$$P(Hwins \mid netyds) = \frac{1}{1+e^{-0.2619-.0078\,netyds)}} \;\ldots.$$

the predicted values are plotted below:

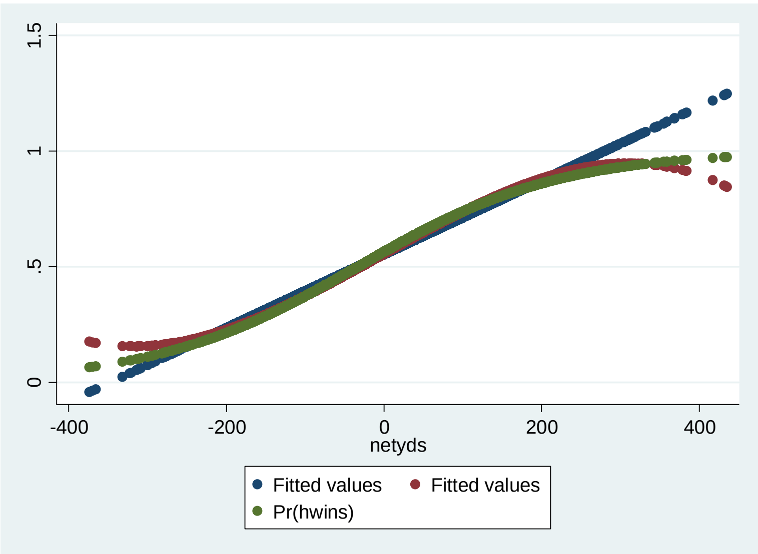
In contrast to the LPM, it is difficult to read marginal effects directly from the estimated logit coefficients. However, the signs of the logit coefficients are interpretable: if the netyds coefficient is positive (as it is above), then increases in netyds will lead to increases in the predicted probability of wins… and had the coefficient been negative, then the direction of the effect would have been negative as well.

More formally: Since $\hat{p} = \left(1+\exp\left[-(\hat{\beta}_0 + \sum \hat{\beta}_i x_i)\right]\right)^{-1}$,

$\frac{\partial \hat{p}}{\partial x_i} = \hat{\beta}_i \exp\left[-(\hat{\beta}_0 + \sum \hat{\beta}_i x_i)\right]\hat{p}^2$, and since $e\,xp\left[-(\hat{\beta}_0 + \sum \hat{\beta}_i x_i)\right] > 0$ and $\hat{p}^2 > 0$,

$$sign\left\{\frac{\partial \hat{p}}{\partial x_i}\right\} = sign\left\{\hat{\beta}_i\right\}.$$

So the signs of the logit coefficients tell you direction of effects, but coefficients need to be processed before the magnitudes of the effects can be determined.



So all three models generate about the same probability estimates for about -200 < netyds < 200, where most of the data reside … and the Logit approach avoids the problems associated with the two LPMs.

We can use Stata's margins command to estimate the marginal impact of changes in netyds on the probability that home wins, estimated at the means:[2]

```
. margins, dydx(_all) atmeans

Conditional marginal effects                      Number of obs   =       3076
Model VCE    : OIM

Expression   : Pr(hwins), predict()
dy/dx w.r.t. : netyds
at           : netyds          =    12.88329 (mean)

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      netyds |   .0018875    .000094    20.07   0.000     .0017032    .0020718
------------------------------------------------------------------------------
```

Note that the estimated impact coefficient is 0.0019, slightly greater than the .0016 estimated using the LPM above.

Just to confirm the lnlikelihood, we can use Solve in Excel to maximize the lnlikelihood function. Here are the results (I've included the LPM results as well):

		LPM		Logit			
		b0	0.55430			b0	0.26215
		b1	0.00159			b1	0.00780
			SSR				sumlnLike
			640.65				(1,849.75)
hwins	**netyds**	**prob**	**resid**	**probwin**	**probloss**	**likelihood**	**lnLike**
0	103	0.719	(0.719)	0.744	**0.256**	0.256	(1.362)
0	130	0.762	(0.762)	0.782	**0.218**	0.218	(1.522)
1	-14	0.532	0.468	**0.538**	0.462	0.538	(0.620)
1	7	0.565	0.435	**0.579**	0.421	0.579	(0.547)
0	-167	0.288	(0.288)	0.261	**0.739**	0.739	(0.303)
0	-52	0.471	(0.471)	0.464	**0.536**	0.536	(0.624)
1	-41	0.489	0.511	**0.486**	0.514	0.486	(0.722)
1	106	0.723	0.277	**0.748**	0.252	0.748	(0.290)
1	-63	0.454	0.546	**0.443**	0.557	0.443	(0.814)
0	31	0.604	(0.604)	0.623	**0.377**	0.377	(0.977)

Notice that in the Logit implementation, and consistent with the MLE methodology, the likelihood function grabs probwin when the home team wins and probloss when they lose.

[2] Note the dydx in the syntax. This gives us an estimate of the marginal impact, as opposed to an elasticity, which would require eyex in the syntax.

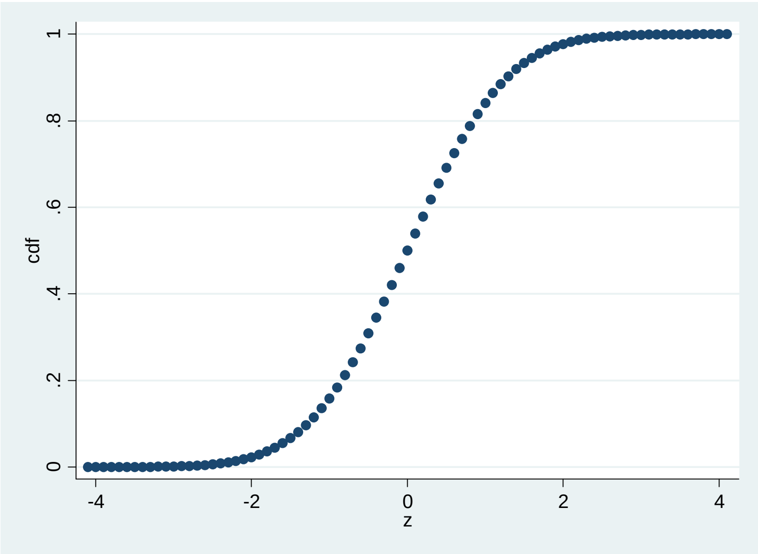
If you look closely, you'll see small differences between the estimated Logit coefficients here and in the Stata output above. You'll also see that the reported lnlikelihood in the Stata output is greater (less negative) than what Excel's Solver routine produced. This should not surprise as we have seen before that Solver's algorithm sometimes stops short of the mark.

Take II: MLE - Probit

In the probit specification,

$$P(Hwins \mid netyds) = \Phi\left[\beta_0 + \beta_1 netyds\right] = prob\left[Z \le \beta_0 + \beta_1 netyds\right],$$

where Z is the standard Normal distribution and $\Phi(.)$ is the Cumulative Distribution Function (CDF) for Z. Here's what the probit function looks like:

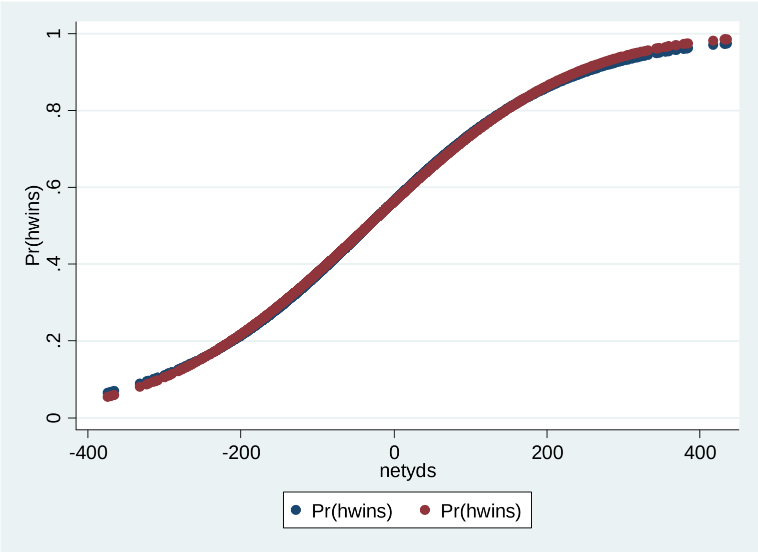


Let's see is using the probit specification of the likelihood function matters:

```
. probit hwins netyds

Iteration 0:   log likelihood =  -2097.596
Iteration 1:   log likelihood = -1847.0646
Iteration 2:   log likelihood = -1846.6966
Iteration 3:   log likelihood = -1846.6965

Probit regression                                 Number of obs   =       3076
                                                  LR chi2(1)      =     501.80
                                                  Prob > chi2     =     0.0000
Log likelihood = -1846.6965                       Pseudo R2       =     0.1196

------------------------------------------------------------------------------
       hwins |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      netyds |   .0046991    .000224    20.98   0.000     .0042601    .0051381
       _cons |   .1594993     .02399     6.65   0.000     .1124797    .2065188
------------------------------------------------------------------------------
```

As in the case of the logit function, the signs of the coefficients in the probit tell you something directionally, but to estimate impacts, you have to do a bit of a calculation.

But we can look at predicted values for the two MLE approaches:



Almost identical predicted values (probit v. logit)! And as before, we can use the margins command to estimate the marginal impact of changes in netyds on the probability of a home team victory:

```
. margins, dydx(_all) atmeans

Conditional marginal effects                      Number of obs   =       3076
Model VCE    : OIM

Expression   : Pr(hwins), predict()
dy/dx w.r.t. : netyds
at           : netyds          =    12.88329 (mean)

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      netyds |   .0018298   .0000868    21.08   0.000     .0016597        .002
------------------------------------------------------------------------------
```

As you can see, and consistent with the figure above, the estimated probit and logit marginal impacts are virtually identical.

And for fun: And since we have the data, here are the results from a few logit models of interest (nettposs is net time of possession). The neutral variable reflects the small number of games played on neutral fields (not that those games have officially designated home and away teams, despite that fact that neither team is at home):

```
. esttab, compress

----------------------------------------------------------------------------
                (1)        (2)        (3)        (4)        (5)        (6)
              hwins      hwins      hwins      hwins      hwins      hwins
----------------------------------------------------------------------------
neutral      -0.141     0.0188     -0.211     0.0106     -0.145    0.00177
            (-0.29)     (0.04)    (-0.48)     (0.02)    (-0.30)     (0.00)

netyds      0.00780***
            (19.87)

netrush                 0.0154***             0.0166***             0.0135***
                       (23.22)               (23.90)               (16.38)

netpass                           0.00148*** 0.00413***            0.00228***
                                   (4.05)       (9.29)              (4.35)

nettposs                                                 0.110***  0.0457***
                                                        (21.00)     (6.50)

_cons         0.263***   0.260***   0.298***   0.249***   0.295***   0.257***
             (6.61)     (6.22)     (8.12)     (5.88)     (7.33)     (6.00)
----------------------------------------------------------------------------
N             3076       3076       3076       3076       3076       3076
----------------------------------------------------------------------------
statistics in parentheses
* p<0.05, ** p<0.01, *** p<0.001

. corr hwins netyds netrush netpass nettposs
(obs=3076)

             |    hwins   netyds  netrush  netpass nettposs
-------------+---------------------------------------------
```

```
       hwins |   1.0000
      netyds |   0.3866   1.0000
     netrush |   0.4718   0.5564   1.0000
     netpass |   0.0733   0.7331  -0.1572   1.0000
    nettposs |   0.4114   0.6911   0.5613   0.3619   1.0000. esttab, compress
```

For the final model we can use the margins command to calculate the elasticities at the mean to get some sense of *economic significance*:

```
. margins, eyex(_all) atmeans

Conditional marginal effects                      Number of obs   =       3076
Model VCE    : OIM

Expression   : Pr(hwins), predict()
ey/ex w.r.t. : neutral netrush netpass nettposs
at           : neutral         =     .006827 (mean)
               netrush         =    8.621586 (mean)
               netpass         =    4.261704 (mean)
               nettposs        =    .6085555 (mean)

------------------------------------------------------------------------------
             |            Delta-method
             |      ey/ex   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     neutral |   4.82e-06    .001437     0.00   0.997    -.0028117    .0028214
     netrush |   .0463929   .0029388    15.79   0.000     .0406329    .0521529
     netpass |   .0038756   .0008918     4.35   0.000     .0021277    .0056234
    nettposs |   .0110883    .001723     6.44   0.000     .0077113    .0144653
------------------------------------------------------------------------------
```

Reading Elasticities from Logit Models

You may recall that for the OLS model with Sample Regression Function $\hat{p} = \hat{\beta}_0 + \sum \hat{\beta}_i x_i$, the elasticity of the predicted value wrt x_i is $\varepsilon_i = \frac{x_i}{\hat{p}} \frac{\partial \hat{p}}{\partial x_i} = \frac{\hat{\beta}_i x_i}{\hat{p}}$, and so the ratio of elasticities for two explanatory variables is $\frac{\varepsilon_i}{\varepsilon_j} = \frac{\hat{\beta}_i x_i}{\hat{\beta}_j x_j}$. And so if these elasticities are being calculated *at the means*, the relative magnitudes will be the product of the ratio of the estimated coefficients and the ratio of the means: $\frac{\varepsilon_i}{\varepsilon_j} = \frac{\hat{\beta}_i}{\hat{\beta}_j} \frac{\overline{x}_i}{\overline{x}_j}$. It turns out that this is also the case for the logit model:

For the logit model, the predicted probability will be defined by $\hat{p} = \left(1 + \exp\left[-(\hat{\beta}_0 + \sum \hat{\beta}_i x_i)\right]\right)^{-1}$, and so

$\varepsilon_i = \frac{x_i}{\hat{p}} \frac{\partial \hat{p}}{\partial x_i} = \frac{x_i}{\hat{p}} \hat{\beta}_i \exp\left[-(\hat{\beta}_0 + \sum \hat{\beta}_i x_i)\right] \hat{p}^2 = \hat{\beta}_i x_i \exp\left[-(\hat{\beta}_0 + \sum \hat{\beta}_i x_i)\right] \hat{p}$. But then

$\frac{\varepsilon_i}{\varepsilon_j} = \frac{\hat{\beta}_i x_i}{\hat{\beta}_j x_j} \frac{\exp\left[-(\hat{\beta}_0 + \sum \hat{\beta}_i x_i)\right] \hat{p}}{\exp\left[-(\hat{\beta}_0 + \sum \hat{\beta}_i x_i)\right] \hat{p}} = \frac{\hat{\beta}_i x_i}{\hat{\beta}_j x_j}$, which is the same as above.

So in both cases, if you know the means of the explanatory variables, you can easily assess relative elasticities by just looking at the product of the estimated coefficients and their respective means.

Just to check to see if this is correct, here are some calculations of relative elasticities in the LPM and logit models estimated above:

		coeffs			***βi xbari***			***elasticities***	
	means	**logit**	**LPM**		**logit**	**LPM**		**logit**	**LPM**
home	0.993	(0.002)	0.0084		(0.002)	0.008		(0.001)	0.015
netrush	8.622	0.013	0.0023		0.116	0.020		0.046	0.035
netpass	4.262	0.002	0.0004		0.010	0.002		0.004	0.003
nettposs	0.609	0.046	0.0090		0.028	0.005		0.011	0.010

	Ratios wrt home				
home	1.000	1.000		1.000	1.000
netrush	(66.150)	2.419		(66.153)	2.419
netpass	(5.526)	0.188		(5.526)	0.188
nettposs	(15.811)	0.655		(15.811)	0.655

	Ratios wrt nettposs				
home	(0.063)	1.526		(0.063)	1.526
netrush	4.184	3.693		4.184	3.693
netpass	0.350	0.287		0.350	0.287
nettposs	1.000	1.000		1.000	1.000

And the same is true for probit models: $\frac{\varepsilon_i}{\varepsilon_j} = \frac{\hat{\beta}_i x_i}{\hat{\beta}_j x_j}$.

Relative Elasticities: Who Knew?

More generally… in fact, this result really is quite general (who knew?):

Suppose that the Sample Regression Function is defined by: $\hat{y} = f\left(\hat{\beta}_0 + \sum \hat{\beta}_i x_i\right)$, for some function $f(.)$, which is increasing in its argument so that $f'(.) > 0$. Note that OLS, logit and probit are all of this type.

Then we have:

- Since $\frac{\partial}{\partial x_i}\hat{y} = \hat{\beta}_i f'\left(\hat{\beta}_0 + \sum \hat{\beta}_i x_i\right)$ and since $f'(.) > 0$, $sign\left(\frac{\partial}{\partial x_i}\hat{y}\right) = sign\left(\hat{\beta}_i\right)$. So the sign of the coefficient tells you the direction of the marginal effect: if $sign\left(\hat{\beta}_i\right) > [<] 0$, then increases in x_i will lead to increases [decreases] in $\hat{y}$.
- $\dfrac{\frac{\partial}{\partial x_i}\hat{y}}{\frac{\partial}{\partial x_j}\hat{y}} = \dfrac{\hat{\beta}_i}{\hat{\beta}_j}$. So the ratio of the estimated coefficients tells you the ratio of the relative marginal effects, the predicted impacts of changes in the two RHS variables
- $\dfrac{\varepsilon_i}{\varepsilon_j} = \dfrac{\hat{\beta}_i x_i}{\hat{\beta}_j x_j}$. So the ratio of the elasticities can be determined by the product of the ratio of the estimated coefficients and the ratio of the respective RHS values at which those elasticities have been calculated. (Since $\varepsilon_i = \frac{x_i}{\hat{y}}\frac{\partial \hat{y}}{\partial x_i} = \frac{\hat{\beta}_i x_i}{\hat{y}} f'()$, $\frac{\varepsilon_i}{\varepsilon_j} = \frac{\hat{\beta}_i x_i}{\hat{\beta}_j x_j}$.)

So while we typically need to evaluate things to determine estimated impacts of changes in the RHS variables, we can often say a fair amount about signs of effects and relative magnitudes by just looking at the estimated parameters.