An Overview of Logistic Regression

Joseph M. Hilbe
hilbe@asu.edu : 14 Mar, 2014

From *Logistic Regression Models,* 2nd edition

1: INTRODUCTION

Regression is a statistical method by which one variable is explained or understood on the basis of one or more other variables. The variable that is being explained is called the dependent, or response, variable; the other variables used to explain or predict the response are called independent variables. Many times independent variables are simply referred to as predictors. I shall use the terms *response* and *predictors* throughout this monograph.

Linear regression is the standard or basic regression model in which the mean of the response is predicted or explained on the basis of a single predictor. The basic model is easily extendable such that it becomes a multivariate linear model; i.e. a linear regression having more than one predictor.

Since linear regression is used to predict the mean value of the response, there will be error involved. Models do not – or at least should not – perfectly predict the mean response. In the case of linear regression the error is normally distributed --- normal meaning as a bell curve. The response is likewise, at least theoretically, normal. That is why the linear model is many times referred to as Gaussian regression.

The short of this is that normal or Gaussian regression assumes that the response is normally distributed. Normal distribution theoretically includes both positive and negative numbers, with zero as the theoretically ideal mean. The linear regression model, however, is extremely robust to violations of the strict adherence to these requirements. What is required though is that the response is continuous and allows the possibility of negative values. It is also required that the variance is constant and that the error terms are normal.

What if the response is binary, taking the values of 1 (success) or 0 (failure)? The error terms cannot be other than 1 or 0 as well. Additionally, the variance is nonconstant. These violations make the Gaussian or linear regression model inappropriate for the analysis of binary responses. The linear probability model, which is simply normal regression used to model binary responses, was once fairly popular when no other means were available. But now all major commercial statistical software applications have the capability to appropriately model binary responses. The model for this situation is termed **logistic regression**.

Logistic regression is the method commonly used to model binary responses. Other models capable of handling binary responses exist as well; e.g. probit, complimentary loglog, and loglog are the most noteworthy examples. However, only logistic regression can be used to determine the odds ratios for its predictors. This is an important feature that plays a vital role in areas such as medical research statistics. We shall discuss this aspect of the logistic model in more detail as we progress in this monograph.

I mentioned that logistic regression is used to model binary responses. This implies that the response is distributed as binary. We don't call this a binary distribution, but rather it is known as the Bernoulli distribution, named after James (or Jacob) Bernoulli (1654-1705). The Bernoulli distribution is a variety of binomial distribution, one in which the binomial denominator has a value of 1. As such, the logistic model is a member of the binomial family of generalized linear models (**GLM**). We shall later touch on this relationship in more depth.

2: CONCEPTS RELATED TO THE LOGISTIC MODEL

The goal of a logistic regression model is to understand a binary or proportional response (dependent variable) on the basis of one or more predictors. For our purposes now, I'll exclude discussion of proportional responses or grouped logistic regression models, concentrating instead on binary responses. We shall discuss them later though since grouped models play an important role in the modeling of health data.

The response variable, which I will many times refer to as simply “y”, is binary, formatted in terms of 1/0. 1 indicates a success; 0 a failure or lack of success. Success is to be thought of in a very wide sense – it can be in terms of “yes”/”no”, “present”/”not present”, “dead”/”alive”, and the like. Most software require the response format to be in terms of 1/0, but software like SPSS will accept 0 vs. any positive integer (which the algorithm internally reformats to a 1). SAS can be confusing because it reverses the relationship. In SAS one predicts by default 0, which can be thought of as a success. Nearly all other applications predict 1 as the success, so keep this in mind when using PROC LOGISTIC. SAS/STAT/GENMOD, SAS’s generalized linear models procedure, of which the binomial family with logit link is a member, maintains the traditional 1/0 relationship, with a 1 indicating success (etc). I suggest that GENMOD be used for all logistic regression modeling endeavors when using SAS, unless one wishes to apply Hosmer & Lemeshow fit and classification statistics to the fitted model, which are only available with PROC LOGISTIC. My caveat is simply to be careful when interpreting your model, and when comparing with other software applications.

There are two major uses to which statisticians employ a logistic model – which I’ll shorten to *LR* model at many forthcoming places. One foremost use is in the interpretation of parameter estimates as odds ratios; the other relates to the calculation of the fitted value, which can be understood as the probability of 1 (success, death, etc). Both of these uses play very important roles in domains such as health and medical research, credit scoring, social science research, and the like.

I’ll use the **heart01** data set for an example of how to interpret parameter estimates, using the Stata statistical package. I favor Stata due to its ease of use and host of ancillary residual and fit assessment options. I also show R code as well.

Stata has three “official” logistic regression commands; **logit**, **logistic**, and **glm**. **Logit** and **logistic** employ a full maximum likelihood method of estimation for calculating parameter estimates (ceefficients) and associated statistics. Model coefficients are displayed by default when using the **logit** command, which is hard-coded in C into the Stata executable. The **logistic** command is an *ado* command based on **logit**. By default it displays odds ratios and associated statistics. The *logit* link of the binomial family of distributions is an optional **glm** command regression procedure, providing full logistic regression estimation and fit statistics. **glm** is an acronym for “Generalized Linear Models”, and displays a number of additional fit statistics not available when using other commands. The SAS **Genmod** and **Logistic** procedures produce similar logistic regression output to Stata. **Genmod** can also be used for GEE estimation and for modeling data using Bayesian GLM capabilities. The SPSS **GLZ** and **Logistic** procedures yield similar default output to Stata and SAS, but have fewer ancillary capabililties.

We return from our digression regarding software implementations to the data we wish to evaluate. The **heart01** data set includes, among other variables, two that we shall use. First is the response variable, *death* [1=die; 0=not die, within 48 hours of initial symptoms]. The other variable is the predictor, *anterior* [having the primary heart injury located in the anterior versus the inferior part of the heart – 1=anterior; 0=inferior (all other variables are dropped from the data in memory)]

We first bring the **heart01** data into memory, keeping only *death* and *anterior*. We then tabulate to observe the relationship between the variables. Thereupon we calculate the odds ratio of *death* on *anterior*. Note that 187 out of nearly 4700 patients died of a heart attack, or myocardial infarction (MI). 120 of these deaths resulted from the patient having an anterior MI, 67 due to an inferior MI.

```
. use heart01, clear
. keep death anterior
. tab death anterior

      | 0=inferior;1=anterior
death | Inferior Anterior |   Total
------+-------------------+--------
    0 |    2,504    2,005 |   4,509
    1 |       67      120 |     187
------+-------------------+--------
Total | 2,571       2,125 |   4,696
```

R

```
=========================================================================
load("c://rfiles/heart.1.Rdata")  # or wherever the data is saved
attach(heart.1); library(gmodels)
CrossTable(death, anterior, expected=FALSE,
                              prop.t=FALSE, prop.r=FALSE, prop.c=FALSE,
                              pop.chisq=FALSE)
=========================================================================
```

The odds ratio of a 2x2 table is best calculated by dividing the odds of predictor, X, when its value is 1 by the odds of X when it has a value of 0. The meaning of odds and odds ratio will be examined shortly. For now, suffice it to say that for the table above, the response or dependent variable is *death*; the predictor is *anterio*r, with *anterior*=1 labeled *Anterior* and *anterior*=0 is labeled *Inferior*. The odds of death among patients having an anterior infarct or heart attack (*Anterior)* is 120/2005 while the odds of death among those having an inferior site infarct (*Inferior)* is 67/2504. The odds ratio is calculated by dividing these two odds; ie by dividing the odds of death of anterior site infarct patients (120/2005=0.05985) by the odds of death among inferior site infarct patients (67/2504=0.02676). The result is OR=0.05985/0.02676=2.23655. The exact value of the odds ratio may be obtained by dividing the values of the cells themselves in place of the intermediate rounded decimals.

```
. di (120/2005) / (67/2504)
2.2367961
```

The odds ratio of a 2x2 table may also be calculated by cross multiplying the diagonals, with a resulting odds ratio of 2.2368.

```
. di (2504*120)/(2005*67) /* calculating odds ratio from tab */
2.2367961
```

Next I subject the same variables to a logistic regression, with *death* as the response. Check the odds ratio, which the algorithm calculates by exponentiating the parameter estimate. The first command uses Stata's *logistic* command, which is estimated using a full maximum likelihood algorithm.

```
MAXIMUM LIKELIHOOD LOGISTIC COMMAND

. logistic death anterior

Logistic regression                               Number of obs   =       4696
                                                  LR chi2(1)      =      28.14
                                                  Prob > chi2     =     0.0000
Log likelihood = -771.92263                       Pseudo R2       =     0.0179
------------------------------------------------------------------------------
       death | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    anterior |   2.236797   .3476534     5.18   0.000     1.649411    3.033362
       _cons |   .0267572   .0033124   -29.25   0.000     .0209927    .0341046
------------------------------------------------------------------------------
```

R

```
=====================================================================
summary(mylogit <- glm(died ~ factor(as.numeric(anterior)),
                       family=binomial, data=heart.1))
OR <- exp(coef(mylogit))
stderr <- sqrt(diag(vcov(mylogit)))
ORsea <- OR*stderr
ci <- exp(confint.default(mylogit))
tab <- data.frame(OR, ORsea, ci)
tab
=====================================================================

> tab
                                      OR       ORsea    X2.5..    X97.5..
(Intercept)                   0.02675719 0.003312331 0.0209927 0.03410457
factor(as.numeric(anterior))2 2.23679607 0.347651628 1.6494126 3.03335666
```

The estimated odds ratio is identical to that calculated by hand using both a 'ratio of odds' method and a 'cross multiplication' method. I'll next use Stata's generalized linear models, or *glm*, command. Note how the binomial family is supplied to the algorithm. The default link function for the binomial family is the *logit*, or *logistic*; we therefore need not specify it.

The *glm* command uses an **I**teratively **R**e-weighted **L**east **S**quares algorithm (IRLS) to estimate GLM parameters, standard errors, and so forth. IRLS is a simplification of maximum likelihood estimation that can be used when the log-likelihood, which is a re-parameterization of the probability function (in this case the binomial PDF), is a member of the exponential family of distributions. The binomial, with Bernoulli as a subset of the binomial, is such an exponential family form member.

```
GLM COMMAND: BINOMIAL FAMILY; LOGIT LINK DEFAULT
<BINOMIAL DENOMINATOR DEFAULT TO 1 -- BERNOULLI DISTRIBUTION>

. glm death anterior, nolog fam(bin) ef

Generalized linear models                          No. of obs      =      4696
Optimization :                                     ML Residual df  =      4694
                                                   Scale parameter =         1
Deviance = 1543.845261                            (1/df) Deviance  =  .3288976
Pearson = 4696                                    (1/df) Pearson   =  1.000426

Variance function: V(u) = u*(1-u)                 [Bernoulli]
Link function : g(u) = ln(u/(1-u))                [Logit]
                                                  AIC              =  .3296093
Log likelihood = -771.9226305                     BIC              = -38141.42
------------------------------------------------------------------------------
          |                  OIM
    death |  Odds Ratio   Std. Err.      z    P>|z|    [95% Conf. Interval]
----------+-------------------------------------------------------------------
 anterior |   2.236796   .3476532     5.18   0.000    1.64941      3.033361
    _cons |   .0267572   .0033124   -29.25   0.000   .0209927      .0341046
------------------------------------------------------------------------------
```

Note that the odds ratio is the same in all four derivations – hand calculation from an odds ratio method and a cross tabulation method, by maximum likelihood (ML), and by GLM. I can interpret the odds ratio here, which as we'll see is simply the exponentiated coefficient, as: "*A patient having an anterior MI has an approximate two and a quarter times greater odds of death within 48 hours of the onset of symptoms than is a patient sustaining an inferior MI*." Of course, a true life model would include a number of other explanatory predictors, often called *confounders*, which would likely alter the odds. However, it is highly doubtful that added predictors would result in the opposite conclusion. If is well known that anterior infarcts have a higher mortality than do inferior or other myocardial infarcts.

We can also determine the **probability of death** based on having an anterior, as well as an inferior, MI. We model the data in a normal fashion, but without exponentiating the coefficients. For this we can use Stata's *logit* command, but can equally use the *glm* command as well.

```
. logit death anterior, nolog

Logistic regression                                Number of obs =   4696
                                                   LR chi2(1)    =  28.14
                                                   Prob > chi2   = 0.0000
Log likelihood = -771.92263                        Pseudo R2     = 0.0179
----------------------------------------------------------------------
   death |     Coef.  Std. Err.      z   P>|z|    [95% Conf. Interval]
---------+------------------------------------------------------------
anterior |  .8050445   .1554244   5.18   0.000    .5004182   1.109671
   _cons | -3.620952   .1237929 -29.25   0.000   -3.863582  -3.378323
----------------------------------------------------------------------
```

We can either use the built-in predict command to obtain the linear predictor, also referred to here as xb or η. [Note: *xb* is also written as $x\beta$, or as βx – all meaning the same]. η is the Greek letter, *eta*, which is traditionally used to refer to the linear predictor when employing GLM methods of estimation. First we shall calculate the linear predictor by hand, then the fitted value, which is a predicted probability.

Using the definition of linear predictor,

$$x_i b = \beta_0 + \beta_j X_{ij} + \cdots + \beta_k X_{nk}$$

where β are the the coefficient estimates for model predictors, from j=1-k, and X are the predictors associated with coefficients. There are i=1-n observations for each predictor. It is important to remember that in a logistic model, as with all regression models, each predictor has the same number of observations, and each observation the same predictors. The model to be estimated by a logistic regression estimation algorithm is an *n x k* matrix with no missing values. At the start of the estimation process Stata automatically drops any observation in a model in which any predictor in the observation has a missing value. In R the user must drop observations with missing values themselves using functions designed for that purpose.

For the example logit model above, the linear predictor, xb, for an anterior MI is calculated as:

```
xb = .805*1 + (-3.621)

. di .805-3.621
-2.816
```

The linear predictor for a patient dying (within 48 hours of symptoms) is -2.816.

The probability -- alternatively symbolozed as *p*, π, μ, or *mu* -- of a patient dying given that they have an anterior site MI is

$$\mu = \frac{1}{1 + exp(-xb)} = \frac{exp(xb)}{1 + exp(xb)}$$

```
mu = 1/(1+exp(-xb)) = 1/(1+exp(2.816)) = .05646567
```

The linear predictor for a patient dying (withnin 48 hours of symptoms) given that they have an inferior site MI is

```
xb = .805*0 + (-3.621)

. di 0 − 3.621
-3.621
```

The probability of an inferior MI is therefore:

```
p = 1/(1+exp(-xb)) = 1/(1+exp(3.621)) = .02605868
```

Now let's use the predict command following **glm**.

```
<model using glm>

. predict xb, xb /* calculates the linear predictor, xb */
. predict mu, mu /* calculates the fitted value, p or · */
. tab xb

   linear |
predictor |   Freq.    Percent     Cum.
----------+---------------------------
-3.620952 |   2,571      54.75    54.75
-2.815907 |   2,125      45.25   100.00
------------+-------------------------
Total | 4,696 100.00
```

```
. tab mu

  predicted |
 mean death |  Freq.   Percent     Cum.
------------+-------------------------
   .0260599 |  2,571     54.75    54.75
   .0564706 |  2,125     45.25   100.00
------------+-------------------------
      Total |  4,696 100.00
```

R

```
===========================================================
eta <- predict(mylogit)
mu <- 1/(1+exp(-eta))
tab2 <- data.frame(eta, mu)
table(tab2)
========================================================

> table(tab2)
                   mu
eta                 0.026059898874871 0.0564705882352649
  -3.62095211271725              2571                  0
  -2.81590759695917                 0               2125
```

The values calculated by hand and produced by using Stata's built-in post-estimation commands are nearly identical. Any differences are a result of rounding error.

To summarize then, following a logistic modeling estimation, we calculate the linear predictor by summing each parameter estimate times its respective data element across all terms, plus the constant.

STATA `predict eta, xb`
R `eta <- predict(mylogit)`

We then calculate the fitted value, μ, by the formula mu = 1/(1+exp(-eta)), ot

STATA `predict mu, mu`
R `mu <- 1/(1+exp(-eta))`

Logit is another concept that commonly associated with logistic models. It is used with respect to both GLM and full maximum likelihood implementations of the logistic model. The natural log of the odds, or logit, is the link function that linearizes the relationship of the response to the predictors. In GLM procedures, it can be determined by calculating *xβ* on the basis of μ.

The odds is the relationship of μ, or *mu*, the probability of success (*y*=1), to 1-μ, or the probability of non-success (*y*=0). You may also see the symbols π or p used to represent μ as well. We symbolize this relationship as

$$\text{odds} = \frac{\mu}{1 - \mu} = \frac{\pi}{1 - \pi}$$

Suppose that we have a probability value of 0.5. The odds, given the above formula, is .5/(1-.5) = .5/.5 = 1. This makes perfect sense when thought about. If the probability of *x* is .5, it means in colloquial terms, 50:50. Neither binary alternative is favored. The odds are 1. A µ of .2 gives an odds of .2/(1-.2) = .2/.8 = ¼ = .25.

The log odds, or *logit*, is simply the natural log of the odds.

$$\text{Logodds} = \text{logit} = \ln(\mu/(1-\mu)) = \ln(\pi/(1-\pi))$$

We can then calculate the fit, µ, as:

```
 exp[ ln (µ /(1- µ)) ]   =  exp(xb)
            (µ /(1 µ -))  =  exp(xb)
(1- µ)/ µ = 1/ exp(xb)   =  exp(-xb)
                1/ µ - 1  =  exp(-xb)
                    1/ µ  =  1 + exp(-xb)
                       µ  =  1 / (1 + exp(-xb))
```

which is the value we indicated earlier for µ in terms of xb.

Technically, we show the relationship of the log-odds or *logit* to the fit and linear predictor for a given observation as:

$$x_i\beta = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \ldots + \beta_n x_{ni}$$

$$\ln(\mu_i/(1-\mu_i)) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \ldots + \beta_n x_{ni}$$

$$\mu_i = 1/(1+\exp(-x_i\beta))$$

We next turn to the nature of the estimation algorithms that are used to determine logistic parameters and associated statistics. I admit that most statisticians learning about the logistic model would rather skip this section, but it is necessary to understand if one is truly interested in understanding how estimation works, and how and why algorithms must be amended to affect changes necessary to better model a particular data situation.