Can binary logistic models be overdispersed?

Joseph M Hilbe
Jet Propulsion Laboratory, California Institute of Technology
and
Arizona State University
29June, 2013; hilbe@asu.edu

The problem I consider here is whether binary models such as logistic and probit regression can be overdispersed. Probably, the better question is whether binary (0,1) response models can be extra-dispersed? Extra-dispersion provides for both over and under-dispersion. I shall argue that the problem is based on two issues, 1) the meaning of overdispersion, and 2) the meaning of a binary model. I will focus on overdispersion rather than extra-dispersion since overdispersed models are much more common than under-dispersed ones.

I recall when I first read that binary response models cannot be overdispersed that I wondered how that could be the case. It seemed to me that I was modeling binary response data that was clustered or correlated in a variety of ways, and that this added variability, or dispersion, to the data. I researched the literature on the topic, and found that statisticians had differing views on the subject. There are a number of statisticians who do talk about overdispersion in logistic models, and others who claim that only grouped logistic models, or binomial logistic models, can be overdispersed.

Binary logistic regression is based on the canonical or natural form of the Bernoulli probability distribution function (PDF), which is a subset of the binomial PDF. Both the Bernoulli and binomial PDFs are members of the one parameter exponential family of distributions, which underlie the statistical method known as *generalized linear models*, or GLM. Binary logistic regression is based on the Bernoulli distribution; grouped logistic regression is based on the binomial PDF.

Let me give an overview regarding the relationship of binary to grouped data. This will help make what follows more understandable. Consider an example model consisting 10 observations, a response variable, *y*, and two predictors, *x*1 and *x*2. The data appears as:

TABLE A

	y	x1	x2	covariate pattern
1:	1	1	0	1
2:	1	1	1	2
3:	0	1	0	1
4:	1	0	0	3
5:	0	0	1	4
6:	1	1	0	1
7:	1	0	1	4
8:	0	0	0	3
9:	0	1	0	1
10:	1	1	0	1

I combine the covariate patterns to more easily see their relationships.

TABLE B

	y	x1	x2	covariate pattern
1:	1	1	0	1
3:	0	1	0	1
6:	1	1	0	1
9:	0	1	0	1
10:	1	1	0	1
2:	1	1	1	2
4:	1	0	0	3
8:	0	0	0	3
5:	0	0	1	4
7:	1	0	1	4

*x*1 and *x*2 must have the same values for each observation in a covariate pattern. Therefore the first five cases in Table B share the same pattern of 1-0. The value of *y* indicates the count of successes, or 1's, for a given covariate pattern. Restructuring the above gives:

TABLE C

y	cases	x1	x2
3	5	1	0
1	1	1	1
1	2	0	0
1	2	0	1

Again, *cases* is the number of observations having the same pattern of covariates. *y* is the number of successes (1's) having the same covariate pattern. As I will emphasize later, the information contained in the observation-based data in Table A is exactly the same as the data in the above grouped data in Table C. An analyst could model the data in Table A using a binary logistic regression, and the data in Table C with a grouped logistic model.

One of the requirements of a probability distribution is that its elements are independent of one another. This is the case for Bernoulli and binomial PDFs underlying logistic regression as well. The respective probability distributions are given as

BINOMIAL PDF

$$f(y|\pi,m)=\prod_{i=1}^{n}\binom{m}{y}\pi_i^{\,y_i}(1-\pi_i)^{m-y_i}$$

BERNOULLI PDF

$$f(y|\pi)=\prod_{i=1}^{n}\pi_i^{\,y_i}(1-\pi_i)^{1-y_i}$$

Normally the product sign is dropped from the above equations as being understood, since it is common to all PDFs. I displayed it to provide complete formulae. Note that the Bernoulli PDF is

the same as the binomial PDF except that the value or n (cases) is 1 for the Bernoulli PDF. As such the leading combination term in the binomial PDF is not needed. The value reduces to 1.

Most analysts think of dispersion as variability. Greater dispersion in the data means greater variability. There are a variety of causes for extra variability, or dispersion, in this type of data. The foremost reason or cause of dispersion occurs when the data are clustered, e.g. when measuring the behavior of dogs from different litters, or even different breeds if we lump all dogs together as part of a larger study of species. Domestic dogs are of the same species, but with a lot of variability in physical appearance and behavioral propensities. If we are modeling clustered data as a binary logistic regression, there is in fact dispersion, or variability, in the data. The data is not independent, as required by the Bernoulli PDF. That's the key to this problem. For a true binary logistic model, there is no overdispersion, or under-dispersion. Each observation is independent of one another. But what happens when we model a logistic regression where the data is clustered, ie. where the observations are not independent? Some analysts then claim as a consequence that the model is not a true Bernoulli model - it's a quasi-likelihood model. It's clear that such a model is not a binary model with independent observations. However, 99.9% of most analysts will say that they are in fact estimating a logistic model - albeit one that may violate the distributional assumptions of the model, but it's still a logistic regression. You see the problem here? If the data is not independent, and hence is extra-dispersed -- it's not a true Bernoulli model. One may adjust the model standard errors (SE's) by scaling, or by using a robust or sandwich variance estimator. I have argued in both Hilbe (2011) and Hilbe (2013) that one should use robust SEs by default when modeling binomial and count models since few models are equi-dispersed, ie. have no variability not consistent with their underlying PDFs. If the data is equi-dispersed, then the robust SE's reduce to the original model SE's.

I should like to provide evidence demonstrating that it makes sense to say that binary logistic models can be overdispersed. I have used a larger version of this model in Hilbe (2009). The ***heartxsmall*** data has 500 observations and consists of heart attack patient data from the Canadian National Registry for Cardiovascular Disease, FASTRAK. The model variables are:

```
Response  :  death 1=died within 48 hours of admission; 0 = not die
Predictors:  anterior  1=anterior site infarction; 0 = other site
             killip    1=normal  2=angina  3=blockage  4= MI
```

Each higher *Killip* level is a higher risk for death. *Anterior* is a risk factor for death as well. I use a logistic regression model of *death* on *anterior* and *killip* level with *killip1* as the reference level. Only the table of estimates is displayed for the binary logistic model. The top model is a standard model; the second is modeled with robust or sandwich standard errors. If the values of the model and robust standard errors differ much, this indicates correlation in the data, or extra-disperson. If the model is extra-dispersed, some statisticians will label the second model as quasi-likelihood, not true logistic model.

I am using the Stata **glm** command to estimate the models, with the option *nolog* being used to inhibit the display of the iteration log, and *nohead* used to inhibit the display of the header statistics.

```
. glm death anterior i.killip , fam(bin) nolog nohead

------------------------------------------------------------------------------
             |                 OIM
       death |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    anterior |   .9805594   .5557307     1.76   0.078    -.1086529    2.069772
             |
      killip |
          2  |   1.311553   .5514751     2.38   0.017     .2306818    2.392425
          3  |    .747291   1.095394     0.68   0.495    -1.399643    2.894225
          4  |   2.632142    1.20385     2.19   0.029     .2726404    4.991644
       _cons |   -4.48001   .5241869    -8.55   0.000    -5.507397   -3.452622
------------------------------------------------------------------------------
```

The sandwich variance estimator, or robust estimator, is applied to the model. It is a post hoc method where the standard errors are adjusted after estimation. One more iteration is given with the adjusted variance producing new standard errors.

```
. glm death anterior i.killip , fam(bin) nolog nohead vce(robust)

------------------------------------------------------------------------------
             |               Robust
       death |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    anterior |   .9805594    .559947     1.75   0.080    -.1169166    2.078035
             |
      killip |
          2  |   1.311553   .5416441     2.42   0.015     .2499502    2.373156
          3  |    .747291   1.134506     0.66   0.510    -1.476301    2.970883
          4  |   2.632142   1.120983     2.35   0.019      .435056    4.829229
       _cons |   -4.48001   .5551493    -8.07   0.000    -5.568082   -3.391937
------------------------------------------------------------------------------
```

There is some difference in the standard errors, but not much. Killip levels 3 and 4 differ the most. In any case, it appears that the data may have extra dispersion in the data over what is expected based on the assumptions of the Bernoulli distribution. But we're not certain.

I now will convert the data to a grouped format, which should allow us to determine if the data and model is overdispersed. First, create a variable (*cases*) that will contain a count of the number of observations having the same covariate pattern for each of the 8 patterns. This is the binomial denominator. *death* is changed to be the number of deaths occurring for each covariate pattern. For example, the first line tells us that there were 291 observations in the 500 observation data set above that have a 0 for *anterior* and 1 for *killip*. Of those 291 covariate patterns, only 3 patients died.

```
. gen byte cases=1
. collapse (sum)  cases (sum) death, by(anterior killip)
. l, nolab clean

     anterior   killip   cases   death
  1.        0        1     219       3
  2.        0        2      46       1
  3.        0        3      12       1
  4.        0        4       3       0
  5.        1        1     155       4
  6.        1        2      51       6
  7.        1        3      12       0
  8.        1        4       2       1
```

I now model the grouped data, checking if there are any differences in coefficients.

```
. glm death anterior i.killip , fam(bin cases) nolog

Generalized linear models                         No. of obs      =         8
Optimization     : ML                             Residual df     =         3
                                                  Scale parameter =         1
Deviance         =  4.705019956                   (1/df) Deviance =   1.56834
Pearson          =  4.290418562                   (1/df) Pearson  =   1.43014 <=

Variance function: V(u) = u*(1-u/cases)           [Binomial]
Link function    : g(u) = ln(u/(cases-u))         [Logit]

                                                  AIC             =  3.716753
Log likelihood   = -9.867013003                   BIC             = -1.533305
------------------------------------------------------------------------------
             |                 OIM
       death |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    anterior |   .9805593   .5557307     1.76   0.078    -.1086529    2.069772
             |
      killip |
          2  |   1.311553   .5514751     2.38   0.017     .2306817    2.392424
          3  |   .7472907   1.095395     0.68   0.495    -1.399643    2.894225
          4  |   2.632142    1.20385     2.19   0.029     .2726404    4.991644
             |
       _cons |   -4.48001   .5241869    -8.55   0.000    -5.507397   -3.452622
------------------------------------------------------------------------------
```

The coefficients and standard errors of the observation and grouped models are the same, which tells us that there is no difference in information we have between the two models. The data is the same, just expressed in a different manner. Look at the Pearson dispersion statistics though. At 1.43 this indicates a rather high amount of overdispersion in the data. The robust estimator on this data will differ from the observation-based model since model size is included in the formula for calculating it.

```
. glm death anterior i.killip , fam(bin cases) nolog nohead vce(robust)

------------------------------------------------------------------------------
             |               Robust
       death |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    anterior |   .9805593   .3285661     2.98   0.003     .3365816    1.624537
             |
      killip |
          2  |   1.311553   .2411769     5.44   0.000     .8388551    1.784251
          3  |   .7472907   1.155795     0.65   0.518    -1.518026    3.012607
          4  |   2.632142   .8119378     3.24   0.001     1.040774    4.223511
             |
       _cons |   -4.48001   .2554531   -17.54   0.000    -4.980689   -3.979331
------------------------------------------------------------------------------
```

The standard errors do differ. *anterior* was not significant using model based standard errors, but is using robust standard errors. Only *killip* level 2 is not significant.

The data in the grouped model shows moderate to high overdispersion. So, if we say that the grouped model can be extra-dispersed, or overdispersed in this case, but the ungrouped model cannot be overdispersed, it makes little sense. The exact same information is being modeled.

One cannot take the Pearson or deviance statistics as a measure of overdispersion for the observation-based model, as one can for the grouped model. However, given that the same information is inherent in both the observation and grouped formats, if the grouped data is overdispersed, the observation is as well. I called this ***implicit*** **overdispersion** (Hilbe, 2011). It is not overdispersion in exactly the same sense as in the grouped model, but it is clear that binary data can be correlated, or overdispersed, and that it has identical information to its associated grouped format, which can be overdispersed.

Two points can be made. For some authors a binary response model, as such, requires that data be independent. If there is correlation or excess variability in the data, the model is NOT a true binary model; it is a quasi-likelihood model. This is a basis of the restriction for claiming that Bernoulli models cannot be overdispersed The second point is that it can be shown that a binary model contains the exact same information as its corresponding grouped model. To say therefore that one model is overdispersed and the other cannot is not consistent.

I think many analysts read that binary models cannot be overdispersed and just do not question It. This happened with the deviance dispersion being the appropriate statistic to measure count model extra-dispersion. Some analysts simply took this on faith, so to speak. But they were mistaken. I proved using simulation that the Pearson *Chi2*-based dispersion is the appropriate measure. R's ***glm*** command provides the deviance and residual DOF for calculating a dispersion statistic, as if the deviance-based dispersion were the correct measure, The ***quasipoisson*** model is supposed to be used for overdispersed count data though. But the ***quasipoisson*** is a Poisson model with standard errors scaled by the Pearson *Chi*2 based dispersion, which is as it should be. Scaling is done my multiplying the model standard error by the square root of the dispersion. Why does R not provide the Pearson *Chi*2 and a dispersion statistic? Perhaps they never noticed the conflict.

On the same topic, R's *quasibinomial* model is also a regular observation-based logistic model where the standard errors have been scaled by the Pearson dispersion statistic. One uses this model when there is evidence of excess variability caused by clustering effects. This extra variability is dispersion, and is adjusted by the dispersion statistic.

I hope that my position is clear. In Hilbe (209) I use the term "*implicitly overdispersed*" for binary logit models which, when converted to grouped format, were proven to be overdispersed. The thing is, with the binary model you can't immediately tell that it is extra-dispersed using the Pearson or deviance dispersion statistics. Most of the times you know if the data is in panel format, or appears to be clustered. It is at times clear that a model has more variability than allowed by the Bernoulli distributional assumptions. Technically such a model is not a Bernoulli model, but rather a Bernoulli *quasi-likelihood* model, or a *quasi-binomial* model. If we define the models that way, then a binary logit model cannot be overdispersed, but a *quasi-binomial*, or better, *quasi-logit* model can be. I have therefore used the concept implicit overdispersion, but it is still overdispersion in the sense of having more variability in the model than is.

References

Hardin, J.W. and J.M Hilbe (2012), *Generalized Linear Models and Extensions, 3rd edition*, Stata Press- CRC Press

Hilbe, J.M. (2009), *Logistic Regression Models*, Chapman & Hall/CRC
Hilbe, J.M. and R.P Robinson (2013), *Methods of Statistical Model Estimation*, Chapman & Hall/CRC
Zuur, A.F., J.M Hilbe, and E Ieno (2013), *A Beginner's Guide to GLM and GLMM with R: a frequentist and Bayesian perspective for ecologists*, Highlands Statistics

Consider an example model consisting 10 observations, a response variable and two predictors, x1 and x2. The data appears as:

	y	x1	x2	covariate pattern
1:	1	1	0	1
2:	1	1	1	2
3:	0	1	0	1
4:	1	0	0	3
5:	0	0	1	4
6:	1	1	0	1
7:	1	0	1	4
8:	0	0	0	3
9:	0	1	0	1
10:	1	1	0	1

I combine the covariate patterns to more easily see their relationships.

	y	x1	x2	covariate pattern
1:	1	1	0	1
3:	0	1	0	1
6:	1	1	0	1
9:	0	1	0	1
10:	1	1	0	1
2:	1	1	1	2
4:	1	0	0	3
8:	0	0	0	3

```
  5:     0   0   1     4
  7:     1   0   1     4
-----------------------------
```

*x*1 and *x*2 must have the same values for each observation in a covariate pattern. Therefore the first five cases directly above share the same pattern of 1-0. The value of *y* indicates the count of successes for a given covariate pattern. Restructuring the above gives:

```
------------------
y   cases   x1   x2
------------------
3       5    1    0
1       1    1    1
1       2    0    0
1       2    0    1
------------------
```