What is Negative Binomial Regression and how is it used in Stata data analysis?

Authored by

stats writer

June 29, 2024

Negative Binomial Regression is a statistical method used in Stata data analysis to model the relationship between a count response variable and one or more explanatory variables. It is a type of generalized linear model that is specifically designed for count data that exhibits overdispersion, meaning there is more variability in the data than can be explained by the model. This method allows for the estimation of the effect of explanatory variables on the count outcome, while taking into account the overdispersion in the data. It is commonly used in fields such as epidemiology, public health, and social sciences to analyze data with count outcomes, such as number of accidents, number of hospitalizations, or number of crimes. Overall, Negative Binomial Regression is a valuable tool in Stata for analyzing count data and understanding the relationship between variables.

Negative Binomial Regression | Stata Data Analysis Examples

Version info: Code for this page was tested in Stata 12.

Negative binomial regression is for modeling count variables, usually for over-dispersed count outcome variables.

Please note: The purpose of this page is to show how to use various data analysis commands. It does not cover all aspects of the research process which researchers are expected to do. In particular, it does not cover data cleaning and checking, verification of assumptions, model diagnostics or

potential follow-up analyses.

Examples of negative binomial regression

Example 1. School administrators study the attendance behavior of high school juniors at two schools. Predictors of the number of days of absence include the type of program in which
the student is enrolled and a standardized
test in math.

Example 2. A health-related researcher is studying the number of
hospital visits in past 12 months by senior citizens in a community based on the
characteristics of the individuals and the types of health plans under which
each one is covered.

Description of the data

Let's pursue Example 1 from above.

We have attendance data on 314 high school juniors from two urban high schools in
the file nb_data.dta. The response variable of interest is

days absent, daysabs. The variable math
is the standardized math score for
each student. The variable prog is a three-level nominal variable indicating the
type of instructional program in which the student is enrolled.

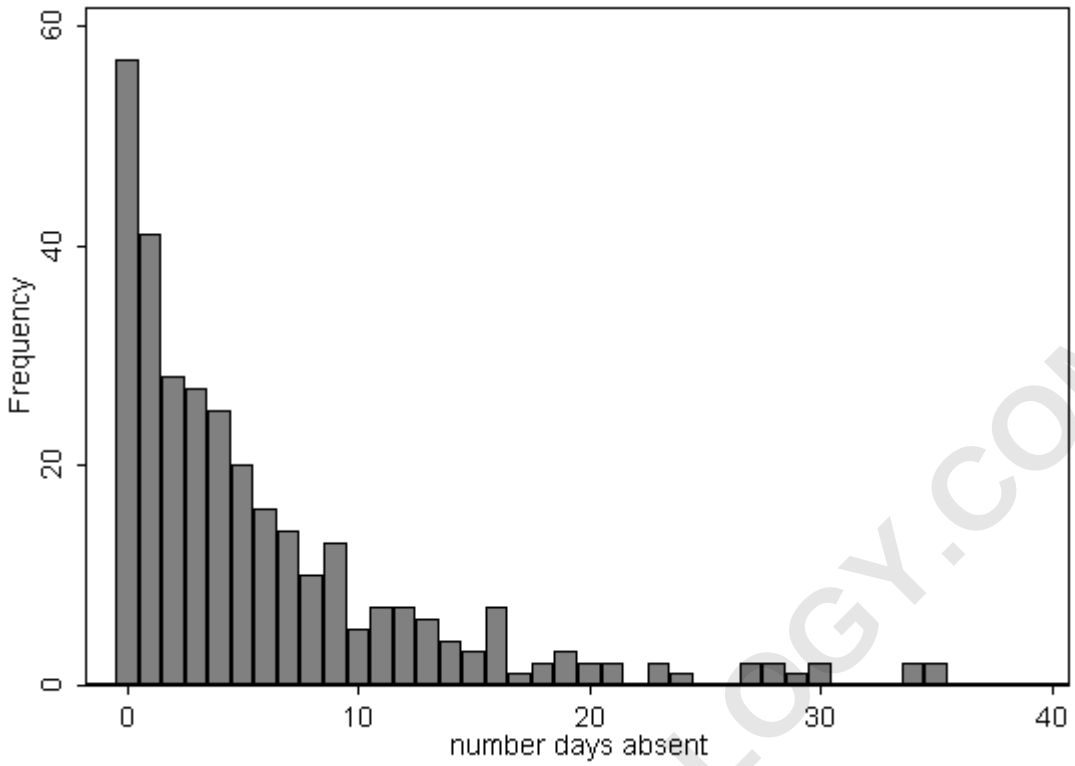
Let's look at the data. It is always a good idea to start with descriptive
statistics and plots.

```
use https://stats.idre.ucla.edu/stat/stata/dae/nb_data, clear

summarize daysabs math

Variable | Obs Mean Std. Dev. Min Max
-------------+--------------------------------------------------------
daysabs | 314 5.955414 7.036958 0 35
math | 314 48.26752 25.36239 1 99

histogram daysabs, discrete freq scheme(s1mono)
```



Each variable has 314 valid observations and their distributions seem quite reasonable. The unconditional mean
of our outcome variable is much lower than its variance.

Let's continue with our description of the variables in this dataset. The table below shows the average numbers of
days absent by program type and seems to suggest that program type is a good candidate for predicting the number of

days absent, our outcome variable, because the mean value of the outcome appears to vary by prog. The variances within each level of prog are higher than the means within each level. These are the conditional means and variances. These differences suggest that over-dispersion is present and that a Negative Binomial model would be appropriate.

```
tabstat daysabs, by(prog) stats(mean v n)
Summary for variables: daysabs
by categories of: prog

prog | mean variance N
---------+----------------------------
1 | 10.65 67.25897 40
2 | 6.934132 55.44744 167
3 | 2.672897 13.93916 107
---------+----------------------------
Total | 5.955414 49.51877 314
----------------------------------------
```

Analysis methods you might consider

Below is a list of some analysis methods you may have encountered. Some of the methods listed are quite reasonable, while others have either fallen out of favor or have limitations.

Negative binomial regression analysis

Below we use the nbreg command to estimate a negative binomial regression model. The i. before prog indicates that it is a factor variable (i.e., categorical variable), and that it should be included in the model as a series of indicator variables.

```
nbreg daysabs math i.prog

Fitting Poisson model:

Iteration 0: log likelihood = -1328.6751
Iteration 1: log likelihood = -1328.6425
Iteration 2: log likelihood = -1328.6425

Fitting constant-only model:

Iteration 0: log likelihood = -899.27009
Iteration 1: log likelihood = -896.47264
```

Iteration 2: log likelihood = -896.47237
Iteration 3: log likelihood = -896.47237

Fitting full model:

Iteration 0: log likelihood = -870.49809
Iteration 1: log likelihood = -865.90381
Iteration 2: log likelihood = -865.62942
Iteration 3: log likelihood = -865.6289
Iteration 4: log likelihood = -865.6289

Negative binomial regression Number of obs = 314
LR chi2(3) = 61.69
Dispersion = mean Prob > chi2 = 0.0000
Log likelihood = -865.6289 Pseudo R2 = 0.0344

---

daysabs | Coef. Std. Err. z P>|z|

---+---

math | -.005993 .0025072 -2.39 0.017 -.010907 -.001079
|
prog |
2 | -.44076 .182576 -2.41 0.016 -.7986025 -.0829175
3 | -1.278651 .2019811 -6.33 0.000 -1.674526 -.882775
|

```
_cons | 2.615265 .1963519 13.32 0.000 2.230423 3.000108
-------------+----------------------------------------------------------------
/lnalpha | -.0321895 .1027882 -.2336506 .1692717
-------------+----------------------------------------------------------------
alpha | .9683231 .0995322 .7916384 1.184442
------------------------------------------------------------------------------
Likelihood-ratio test of alpha=0: chibar2(01) = 926.03
Prob>=chibar2 = 0.000
```

```
test 2.prog 3.prog

( 1) 2.prog = 0
( 2) 3.prog = 0

chi2( 2) = 49.21
Prob > chi2 = 0.0000
```

We can also see the results as incident rate ratios by using the
irr option.

```
nbreg, irr
```

Negative binomial regression Number of obs = 314
LR chi2(3) = 61.69
Dispersion = mean Prob > chi2 = 0.0000
Log likelihood = -865.6289 Pseudo R2 = 0.0344

--
daysabs | IRR Std. Err. z P>|z| [95% Conf. Interval]
-------------+--
math | .9940249 .0024922 -2.39 0.017 .9891523 .9989216
|
prog |
2 | .6435471 .1174963 -2.41 0.016 .4499573 .920427
3 | .2784127 .0562341 -6.33 0.000 .1873969 .4136335
-------------+--
/lnalpha | -.0321895 .1027882 -.2336506 .1692717
-------------+--
alpha | .9683231 .0995322 .7916384 1.184442
--
Likelihood-ratio test of alpha=0: chibar2(01) = 926.03
Prob>=chibar2 = 0.000

The output above indicates that the incident rate for 2.prog is 0.64
times the incident rate for the reference group (1.prog).

Likewise, the
incident rate for 3.prog is 0.28 times the incident rate for the reference group
holding the other variables constant. The percent change in the incident rate of
daysabs is a 1% decrease for every unit increase in math.

The form of the model equation for negative binomial regression is the same as
that for Poisson regression. The log of the outcome is predicted with a linear
combination of the predictors:

log(daysabs) = Intercept + b1(prog=2) + b2(prog=3)
+ b3math.

This implies:

daysabs = exp(Intercept + b1(prog=2) + b2(prog=3)+
b3math) = exp(Intercept) * exp(b1(prog=2)) * exp(b2(prog=3))
*** exp(b3math)**

The coefficients have an additive effect in the log(y)

scale and the
IRR have a multiplicative effect in the y scale. The dispersion
parameter alpha in negative binomial regression does not effect the
expected counts, but it does effect the estimated variance of the expected counts. More
details can be found in the Stata documentation.

For additional information on the various metrics in which the results can be
presented, and the interpretation of such, please see Regression Models for
Categorical Dependent Variables Using Stata, Second Edition by J. Scott Long
and Jeremy Freese (2006).

To understand the model better, we can use the margins command. Below we use the
margins command to calculate the predicted counts at each level of prog, holding all other variables (in this example,
math) in the model at their means.

```
margins prog, atmeans
```

Adjusted predictions Number of obs = 314
Model VCE : OIM

Expression : Predicted number of events, predict()
at : math = 48.26752 (mean)
1.prog = .1273885 (mean)
2.prog = .5318471 (mean)
3.prog = .3407643 (mean)

--
| Delta-method
| Margin Std. Err. z P>|z|
-------------+--
prog |
1 | 10.2369 1.674445 6.11 0.000 6.955048 13.51875
2 | 6.587927 .5511718 11.95 0.000 5.50765 7.668204
3 | 2.850083 .3296496 8.65 0.000 2.203981 3.496184
--

In the output above, we see that the predicted number of events for level 1
of prog is about 10.24, holding math at its mean. The predicted
number of events for level 2 of prog is lower at 6.59, and

**the predicted
number of events for level 3 of prog is about 2.85. Note that the
predicted count of level 2 of prog is (6.587927/10.2369) = 0.64 times the
predicted count for level 1 of prog. This matches what we saw in the IRR
output table.**

**Below we will obtain the predicted number of events for values of math
that range from 0 to 100 in increments of 20.**

```
margins, at(math=(0(20)100)) vsquish
```

```
Predictive margins Number of obs = 314
Model VCE : OIM

Expression : Predicted number of events, predict()
1._at : math = 0
2._at : math = 20
3._at : math = 40
4._at : math = 60
5._at : math = 80
6._at : math = 100
```

```
------------------------------------------------------------------------------
| Delta-method
| Margin Std. Err. z P>|z|
-------------+----------------------------------------------------------------
_at |
1 | 7.717607 .9993707 7.72 0.000 5.758876 9.676337
2 | 6.845863 .6132453 11.16 0.000 5.643924 8.047802
3 | 6.072587 .3986397 15.23 0.000 5.291268 6.853907
4 | 5.386657 .4039343 13.34 0.000 4.59496 6.178354
5 | 4.778206 .5226812 9.14 0.000 3.75377 5.802643
6 | 4.238483 .6474951 6.55 0.000 2.969416 5.50755
------------------------------------------------------------------------------
```

The table above shows that with prog at its observed values and math held at 0 for all observations, the average predicted count (or average number of days absent) is about 7.72; when math = 100, the average predicted count is about 4.24. If we compare the predicted counts at any two levels of math, like math = 20 and math = 40, we can see that the ratio is (6.072587/6.845863) = 0.887. This matches the IRR of 0.994 for a 20 unit change: 0.994^20

= 0.887.

The user-written fitstat command (as well as Stata's estat
commands) can be used to obtain additional model fit information that may be helpful if
you want to compare models. You can type search fitstat to download
this program (see
How can I use the search command to search for programs and get additional help?
for more information about using search).

fitstat

Measures of Fit for nbreg of daysabs

Log-Lik Intercept Only: -896.472 Log-Lik Full Model: -865.629
D(308): 1731.258 LR(3): 61.687
Prob > LR: 0.000
McFadden's R2: 0.034 McFadden's Adj R2: 0.028
ML (Cox-Snell) R2: 0.178 Cragg-Uhler(Nagelkerke) R2: 0.179
AIC: 5.552 AIC*n: 1743.258

BIC: -39.555 BIC': -44.439

BIC used by Stata: 1760.005 AIC used by Stata: 1741.258

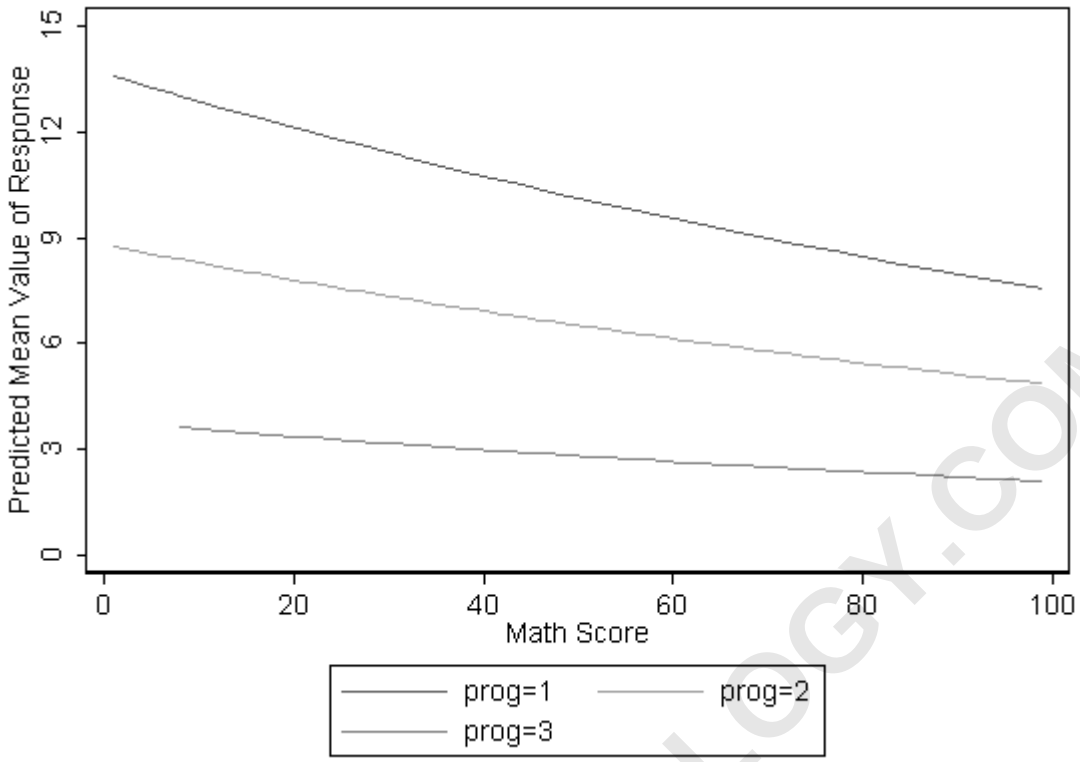
You can graph the predicted number of events with the commands below.

The graph indicates that the most days absent are predicted for those in the

academic program 1, especially if the student has a low math score. The

lowest number of predicted days absent is for those students in program 3.

```
predict c
sort math
twoway (line c math if prog==1) ///
(line c math if prog==2) ///
(line c math if prog==3), ///
ytitle("Predicted Mean Value of Response") ylabel(0(3)15 ,nogrid) ///
xtitle("Math Score") legend(order(1 "prog=1" 2 "prog=2" 3 "prog=3")) scheme(s1mono)
```



Things to consider

See also

References