What is the process for conducting Negative Binomial Regression in SPSS?

Authored by

stats writer

June 29, 2024

The Negative Binomial Regression is a statistical method used to analyze count data with overdispersion, i.e., when the variance is greater than the mean. It is commonly used in social science, epidemiology, and business research. The process for conducting Negative Binomial Regression in SPSS involves several steps. First, the researcher must import the data into SPSS and ensure that the response variable is in the appropriate format. Then, the researcher needs to specify the model by selecting the Negative Binomial Regression option from the Regression menu. This will bring up a dialog box where the response and predictor variables can be selected. After specifying the model, the researcher needs to assess the model fit by examining the significance of the model and the goodness-of-fit statistics. If the model is a good fit, the researcher can then interpret the results and draw conclusions about the relationship between the variables. It is important to note that proper interpretation of the results requires an understanding of the underlying assumptions and limitations of the Negative Binomial Regression. Overall, conducting Negative Binomial Regression in SPSS involves careful data management, model specification, and interpretation to ensure accurate and meaningful results.

Negative Binomial Regression | SPSS Data Analysis Examples

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.

This page was updated using SPSS 19.

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 https://stats.idre.ucla.edu/wp-content/uploads/2016/02/nb_data.sav. 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.

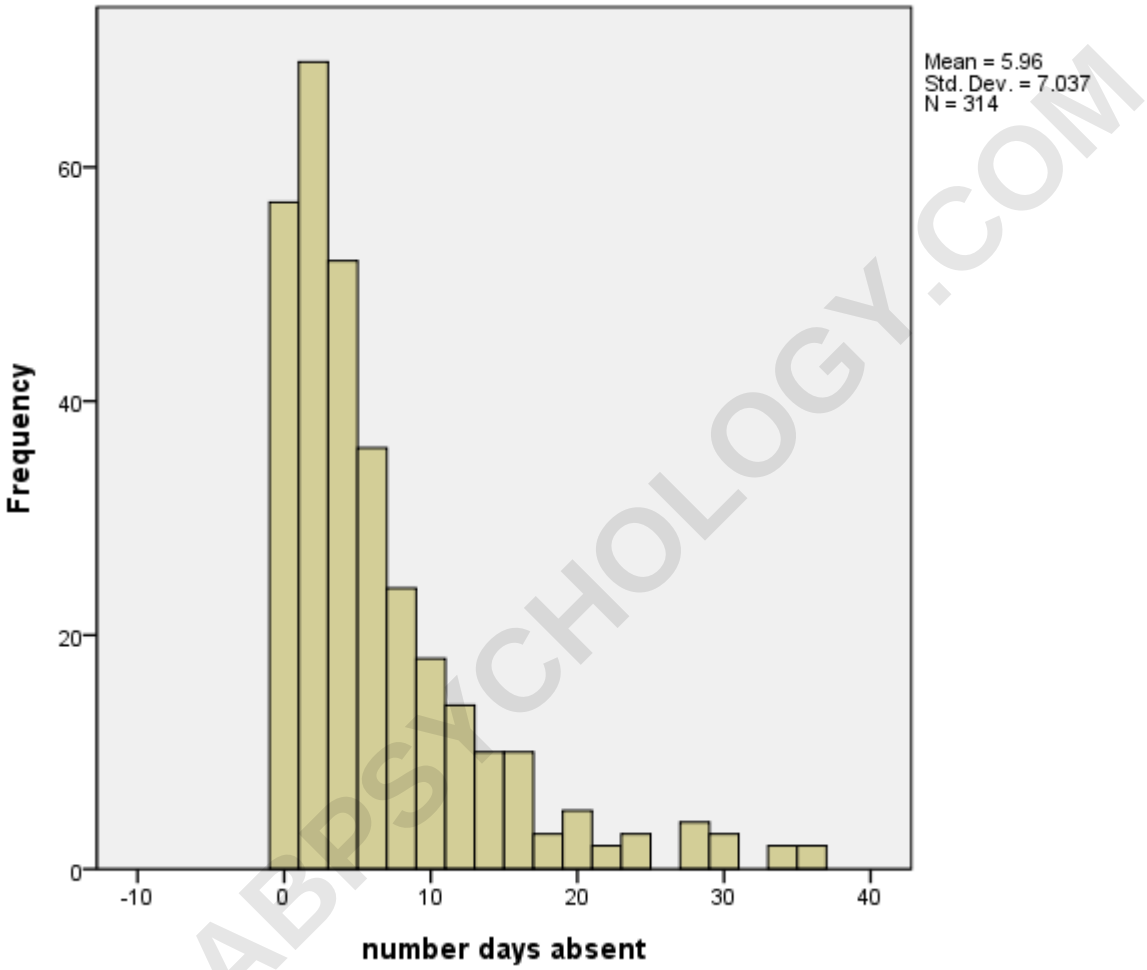
```
get file "D:datanb_data.sav".

descriptives variables = daysabs math.
```


Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
number days absent	314	0	35	5.96	7.037
ctbs math pct rank	314	1	99	48.27	25.362
Valid N (listwise)	314				

graph

/histogram daysabs.



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.

```
means tables=daysabs by prog
/cells mean count var.
```

Report

number days absent

prog	Mean	N	Variance
1	10.65	40	67.259
2	6.93	167	55.447
3	2.67	107	13.939
Total	5.96	314	49.519

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 genlin command to estimate a negative binomial regression model. We use the SPSS keyword by to indicate that the variable that follows is a categorical predictor, and we use the SPSS keyword with to indicate that the variable that follow is a continuous predictor. We use the (order = descending) option to use the first level of the variable prog as the reference group. On the model subcommand, we again list the predictor variables. We also indicate that the distribution to be used is negbin (negative binomial) and the link is a log link. By default, SPSS will not estimate the

dispersion parameter.

Because we wish for this to be estimated, we specified (MLE) after our

distribution.

SPSS provides many output tables, so we will interrupt the output to

explain a few tables at a time.

```
genlin daysabs by prog (order = descending) with math
/model prog math distribution = negbin(MLE) link=log.
```

Model Information

Dependent Variable	number days absent
Probability Distribution	Negative binomial (MLE)
Link Function	Log

Case Processing Summary

	N	Percent
Included	314	100.0%
Excluded	0	0.0%
Total	314	100.0%

In the first two tables above, we see that the probability distribution used was negative binomial, the link function was log, and

that all 314 cases were used in the analysis. We then see information on the distribution of the categorical predictor variable, as well as information on the distribution of the dependent variable and the continuous predictor variable.

Goodness of Fit[b]

	Value	df	Value/df
Deviance	358.519	309	1.160
Scaled Deviance	358.519	309	
Pearson Chi-Square	339.877	309	1.100
Scaled Pearson Chi-Square	339.877	309	
Log Likelihood[a]	-865.629		
Akaike's Information Criterion (AIC)	1741.258		
Finite Sample Corrected AIC (AICC)	1741.453		
Bayesian Information Criterion (BIC)	1760.005		
Consistent AIC (CAIC)	1765.005		

Dependent Variable: number days absent
Model: (Intercept), prog, math

a. The full log likelihood function is displayed and used in computing information criteria.
b. Information criteria are in small-is-better form.

The table above provides several indices of the goodness of fit of the model. These measures can be used to compare models.

Omnibus Test[a]

Likelihood Ratio Chi-Square	df	Sig.
61.687	3	.000

Dependent Variable: number days absent
Model: (Intercept), prog, math

a. Compares the fitted model against the intercept-only model.

Tests of Model Effects

Source	Type III		
	Wald Chi-Square	df	Sig.
(Intercept)	212.477	1	.000
prog	49.214	2	.000
math	5.714	1	.017

Dependent Variable: number days absent
Model: (Intercept), prog, math

The tables above provide tests of the model as a whole (Omnibus Test). The likelihood ratio chi-square provides a test of the overall model comparing this model to a model without any predictors (a "null" model). We can see that our model is a significant improvement over such a model by looking at the p-value of this test.

In the Tests of Model Effects table, we see that each of the predictors

is statistically significant. The table includes the two degree of
freedom test of prog, which indicates that as a whole, the variable
prog is a significant predictor of dayabs.

Parameter Estimates

Parameter	B	Std. Error	95% Wald Confidence Interval		Hypothesis Test		
			Lower	Upper	Wald Chi-Square	df	Sig.
(Intercept)	2.615	.1964	2.230	3.000	177.403	1	.000
[prog=3]	-1.279	.2020	-1.675	-.883	40.076	1	.000
[prog=2]	-.441	.1826	-.799	-.083	5.828	1	.016
[prog=1]	0[a]	.	.	.	.	.	.
math	-.006	.0025	-.011	-.001	5.714	1	.017
(Scale)	1[b]						
(Negative binomial)	.968	.0995	.792	1.184			

Dependent Variable: number days absent
Model: (Intercept), prog, math

a. Set to zero because this parameter is redundant.
b. Fixed at the displayed value.

The table Parameter Estimates contains the negative binomial
regression coefficients for each of the predictor variables along with their standard errors,
Wald chi-square values, p-values and 95% confidence intervals for the coefficients.
Both of the dummy variables for the variable
prog are statistically significant. Compared to level 1 of prog, the expected log count for level 2 decreases by

0.44.
Compared to level 1 of prog, the expected log count of 3.prog
decreases by 1.28. The variable math has a coefficient of -0.006, which
is statistically significant. This means that for each one-unit
increase on math, the expected log count of the number of days absent
decreases by 0.006 day.
Additionally, there is an estimate of the dispersion coefficient, (Negative
binomial). A Poisson model is one in which this value is constrained to
zero. In this example, the parameter's 95% confidence interval does not include
zero, suggesting that the negative binomial model form is more appropriate than
the Poisson. An estimate greater than zero suggests over-dispersion (variance
greater than mean). An estimate less than zero suggests under-dispersion, which
is very rare.

If you would like the results displayed as incident rate ratios, you can use the
(exponentiated) option on the print subcommand after solution.

genlin daysabs by prog (order = descending) with math
/model prog math distribution = negbin(MLE) link=log
/print solution (exponentiated).

Parameter Estimates

Parameter	B	Std. Error	95% Wald Confidence Interval		Hypothesis Test			Exp(B)	95% Wald Confidence Interval for Exp(B)	
			Lower	Upper	Wald Chi-Square	df	Sig.		Lower	Upper
(Intercept)	2.615	.1964	2.230	3.000	177.403	1	.000	13.671	9.304	20.088
[prog=3]	-1.279	.2020	-1.675	-.883	40.076	1	.000	.278	.187	.414
[prog=2]	-.441	.1826	-.799	-.083	5.828	1	.016	.644	.450	.920
[prog=1]	0[a]	.	.	.	.	.	.	1	.	.
math	-.006	.0025	-.011	-.001	5.714	1	.017	.994	.989	.999
(Scale)	1[b]									
(Negative binomial)	.968	.0995	.792	1.184						

Dependent Variable: number days absent
Model: (Intercept), prog, math

a. Set to zero because this parameter is redundant.
b. Fixed at the displayed value.

Looking at the Exp(B) column in the table above indicates that the
incident rate for prog=2 is 0.64 times the incident rate for the
reference group (prog=1). Likewise, the incident rate for prog=3
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 overdispersion
parameter alpha in negative binomial regression does

not effect the
expected counts, but it does effect the estimated variance of the expected
counts.

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).

For assistance in further understanding the model, we can use the emmeans
subcommand. Below we use the emmeans subcommand to calculate the predicted number of events at each level of
prog, holding all other variables (in this example, math) in the
model at their means.

```
genlin daysabs by prog (order = descending) with math
/model prog math distribution = negbin(MLE) link=log
```

/emmeans tables = prog scale = original.

Estimates

prog	Mean	Std. Error	95% Wald Confidence Interval	
			Lower	Upper
3	2.85	.330	2.20	3.50
2	6.59	.551	5.51	7.67
1	10.24	1.674	6.96	13.52

Covariates appearing in the model are fixed at the following values: math=48.27

In the output above, we see that the predicted number of events (e.g., days
absent) 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.

Below we will obtain the predicted number of events while holding math
at 20, then 40.

genlin daysabs by prog (order = descending) with math
/model prog math distribution = negbin(MLE) link=log

/emmeans control = math (20)

/emmeans control = math (40).

< some output omitted >

Estimated Marginal Means 1: Grand Mean

Estimates

Mean	Std. Error	95% Wald Confidence Interval	
		Lower	Upper
6.84	.685	5.49	8.18

Covariates appearing in the model are fixed at the following values: math=20.00

Estimated Marginal Means 2: Grand Mean

Estimates

Mean	Std. Error	95% Wald Confidence Interval	
		Lower	Upper
6.06	.450	5.18	6.95

Covariates appearing in the model are fixed at the following values: math=40.00

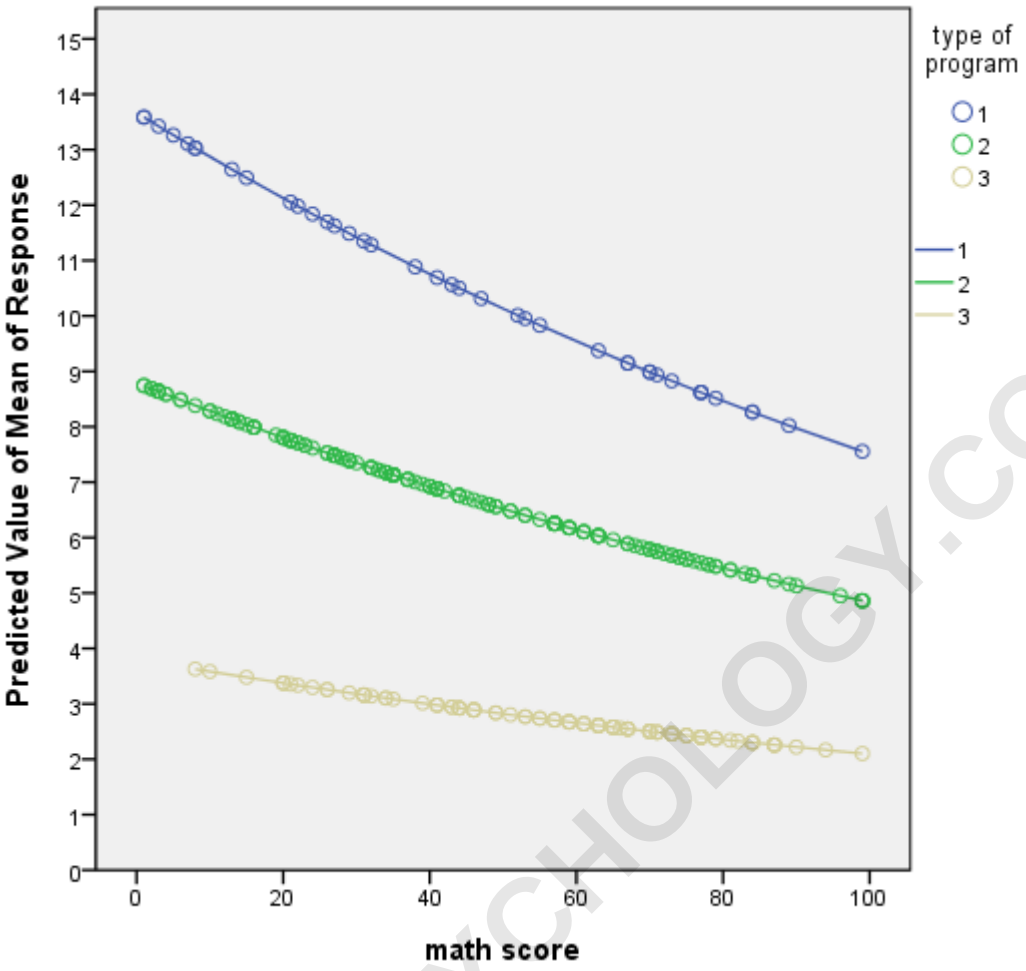
The tables above show that with prog at its observed values and math held at 20 for all observations, the average predicted count (or average number of days absent) is about 6.84; when math = 40, the average predicted count is about 6.06. If we compare the predicted counts at any two

levels of math, like math =
20 and math = 40, we can see that the ratio is (6.06/6.84) = 0.89. This matches
the IRR of 0.994 for a 20 unit change: 0.994^20 = 0.89.

You can graph the predicted number of events with the commands below.
The graph indicates that the most awards are predicted for those in program
type 1, especially if the student has a high math score. The lowest number
of predicted awards is for those students in program type 3.

```
genlin daysabs by prog (order = descending) with math
/model prog math distribution = negbin(MLE) link=log
/save meanpred (mean_values).

GGRAPH
/GRAPHDATASET NAME="graphdataset"
VARIABLES=math mean_values prog
/GRAPHSPEC SOURCE=INLINE.
BEGIN GPL
SOURCE: s=userSource(id("graphdataset"))
DATA: math=col(source(s), name("math"))
```

```
DATA: mean_values=col(source(s), name("mean_values"))
DATA: prog=col(source(s), name("prog"), unit.category())
GUIDE: axis(dim(1), label("math score"))
GUIDE: axis(dim(2), label("Predicted Value of Mean of Response"), delta(1))
GUIDE: legend(aesthetic(aesthetic.color.exterior), label("type of program"))
SCALE: linear(dim(1), min(0), max(100))
SCALE: linear(dim(2), min(0), max(14))
SCALE: cat(aesthetic(aesthetic.color.exterior), include("1.00", "2.00", "3.00"))
ELEMENT: point(position(math*mean_values), color.exterior(prog))
ELEMENT: line(position(math*mean_values), color(prog))
END GPL.
```



Things to consider

See also

References

<!-webbot bot="PurpleText" PREVIEW="Don't change anything below this line."