STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **8**, September 2020, pp 773–789.
Published online in International Academic Press (www.IAPress.org)

overdisp: A Stata (and Mata) Package for Direct Detection of Overdispersion in Poisson and Negative Binomial Regression Models

L. P. Fávero [1,*], P. Belfiore [2], M. A. Santos [1], R. Freitas Souza [1]

[1] *School of Economics, Business and Accounting, University of São Paulo, Brazil*
[2] *Center for Engineering, Modeling and Applied Social Sciences, Federal University of ABC, Brazil*

Abstract Stata has several procedures that can be used in analyzing count-data regression models and, more specifically, in studying the behavior of the dependent variable, conditional on explanatory variables. Identifying overdispersion in count-data models is one of the most important procedures that allow researchers to correctly choose estimations such as Poisson or negative binomial, given the distribution of the dependent variable. The main purpose of this paper is to present a new command for the identification of overdispersion in the data as an alternative to the procedure presented by Cameron and Trivedi [5], since it directly identifies overdispersion in the data, without the need to previously estimate a specific type of count-data model. When estimating Poisson or negative binomial regression models in which the dependent variable is quantitative, with discrete and non-negative values, the new Stata package `overdisp` helps researchers to directly propose more consistent and adequate models. As a second contribution, we also present a simulation to show the consistency of the overdispersion test using the `overdisp` command. Findings show that, if the test indicates equidispersion in the data, there are consistent evidence that the distribution of the dependent variable is, in fact, Poisson. If, on the other hand, the test indicates overdispersion in the data, researchers should investigate more deeply whether the dependent variable actually exhibits better adherence to the Poisson-Gamma distribution or not.

Keywords overdisp, Overdispersion, Count-Data Models, Stata

AMS 2010 subject classifications 62J02, 62-07

DOI: 10.19139/soic-2310-5070-557

1. Introduction

Many situations have as an outcome of interest a nonnegative integer, or a count, denoted by y, $y \in \mathbb{N}_0 = 0, 1, 2, \ldots$. The benchmark model for the analysis of integer count-data is the Poisson regression model, which restricts the variance of the data to be equal to the mean, conditional on explanatory variables [5, 6, 7]. Failures of this restriction can allow researchers to estimate parameters considering more general distributions, such as the negative binomial.

Many commonly used count-data models are implemented in a variety of software packages, such as the `poisson` and `nbreg` packages in Stata [19], `glm` and `glm.nb` packages in R [16] and `GENMOD` Procedure in SAS [18], and many applications of these models can be found in economics, finance, actuary, ecology, demography, sociology psychology, and health, among other relevant fields of knowledge [7, 11, 14, 20, 21].

Following the test implemented in Stata through a sequence of four commands proposed by Cameron and Trivedi [5], we present the new package `overdisp` to directly identify overdispersion in Stata. There are notable advantages to running `overdisp` in this way. First, it provides a simple, intuitive, fast and easy command which allows users to choose between Poisson and negative binomial estimations in the presence of count-data. Second, prior to fitting a particular model, users can take advantage of Statas excellent statistics and data

[*] Correspondence to: Luiz Paulo Fávero (Email: lpfavero@usp.br). School of Economics, Business and Accounting - University of São Paulo. Av. Prof. Luciano Gualberto, 908 - FEA 1 - r. G173 - Cidade Universitária. São Paulo - SP - Brazil / Zip Code: 05508-900.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

management commands to prepare and descriptively analyze their data [13]. Last, all analyses can be reproduced and documented for publication and review by typing the command into a file and running it directly.

One of the key advantages of such a package for count-data regression is that, for the development of new models, overdispersion can be identified in one line of code rather than having to program the test.

The remainder of the article is structured as follows. Section 2 briefly reviews count-data models. Section 3 formally presents the overdispersion test. Section 4 describes how to install `overdisp` in Stata, presents the `overdisp` command syntax and describes the estimation options. Section 5 illustrates `overdisp` by replicating examples presented in Cameron and Trivedi [5] and Fávero and Belfiore [7]. Section 6 performs a simulation study using overdisp to demonstrate an important research finding. Section 7 concludes.

2. Review of count-data models

According to Rabe-Hesketh and Skrondal [17], a general count-data regression model can be written as follows:

$$\ln(\widehat{u}_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_k X_{ki} \tag{1}$$

where β_0 represents the constant, $\beta_j(1, 2, ..., k)$ are the estimated parameters for each X_j explanatory variable, $\widehat{u}$ is the expected number of occurrences or the estimated incidence rate ratio for the phenomenon under study for a given exposition (period, area, region, among other examples) and for a determined observation $i(i = 1, 2, ..., n)$, and n is the sample size.

2.1. Poisson regression models

The beginning point for the study of count-data regression models is the presentation of the Poisson distribution that, for a determined observation i, has the following occurrence probability for a count m in a given exposition:

$$p(\widehat{y}_i = m) = \frac{e^{-\widehat{u}_i}.\widehat{u}_i^m}{m!} \tag{2}$$

where $m = 0, 1, 2, ...$

As discussed by Cameron and Trivedi [5], Avci [2] and Fávero and Belfiore [7], in the Poisson distribution, the mean and the variance for the variable under study should be equal to $\widehat{u}$, as can be shown as follows:

- Mean:

$$E(y|X) = \sum_{m=0}^{\infty} m \frac{e^{-\widehat{u}}.\widehat{u}^m}{m!} = \widehat{u} \sum_{m=1}^{\infty} \frac{e^{-\widehat{u}}.\widehat{u}^{m-1}}{(m-1)!} = \widehat{u} \tag{3}$$

- Variance:

$$\begin{aligned} Var(y|X) = \sum_{m=0}^{\infty} \frac{e^{-\widehat{u}}.\widehat{u}^m}{m!}(m - \widehat{u})^2 = \sum_{m=0}^{\infty} \frac{e^{-\widehat{u}}.\widehat{u}^m}{m!}(m^2 - 2.m.\widehat{u} + \widehat{u}^2) = \\ \widehat{u}^2 \sum_{m=2}^{\infty} \frac{e^{-\widehat{u}}.\widehat{u}^{m-2}}{(m-2)!} + \widehat{u} \sum_{m=1}^{\infty} \frac{e^{-\widehat{u}}.\widehat{u}^{m-1}}{(m-1)!} - \widehat{u}^2 = \widehat{u} \end{aligned} \tag{4}$$

Poisson regression model parameters can be estimated by maximum likelihood, where the dependent variable follows a Poisson distribution [9]. Being the probability of occurrence for a specific count m in a determined exposition, we can define the log likelihood function for Poisson regression models as being

$$LL = \sum_{i=1}^{n} [-\widehat{u}_i + (y_i)\ln(\widehat{u}_i) - \ln(y_i!)] \tag{5}$$

2.2. *Negative binomial regression models*

Negative binomial regression models are also part of the regression models for count-data, but the estimation takes into account the existence of overdispersion in the dependent variable, conditional on explanatory variables [1, 10]. The probability function for a negative binomial distribution (also known as Poisson-Gamma distribution), which would permit us to calculate the occurrence probability for a count m, given a determined exposition, can be written as:

$$p(y_i = m) = \begin{pmatrix} m + \alpha^{-1} - 1 \\ \psi - 1 \end{pmatrix} \cdot \left(\frac{\alpha^{-1}}{\widehat{u}_i + \alpha^{-1}}\right)^{\alpha^{-1}} \cdot \left(\frac{\widehat{u}_i}{\widehat{u}_i + \alpha^{-1}}\right)^{m} \tag{6}$$

where α is the inverse of the form parameter of the Gamma distribution $(\alpha > 0)$. So, in the presence of overdispersion, we have:

- Mean:

$$E(y|X) = \widehat{u} \tag{7}$$

- Variance:

$$Var(y|X) = \widehat{u} + \alpha.\widehat{u}^2 \tag{8}$$

The second term of the negative binomial distribution expression variance (equation (8)) represents overdispersion and, if $\alpha \to 0$, this phenomenon will not be present in the data, favoring the estimations of the Poisson regression model. However, if α is statistically greater than zero, the existence of overdispersion in the dependent variable, conditional on explanatory variables, causes that the negative binomial regression should be estimated.

Overdispersion is often encountered when fitting parametric models. The Poisson distribution has one free parameter and does not allow for the variance to be adjusted independently of the mean (equation (4)). If overdispersion is a feature, an alternative model with additional free parameters may provide a better fit. In the case of count data, a negative binomial regression model can be proposed instead, in which the mean of the Poisson distribution can itself be thought of as a random variable drawn, in this case, from the Gamma distribution, thereby introducing an additional free parameter (term $\alpha.\widehat{u}^2$ in equation (8)).

In this sense, overdispersion can be defined as a great variability (statistical dispersion) that occurs in a variable in comparison with its mean. According to Fávero and Belfiore [7], high variation in the data is due to heterogeneous or non-uniform samples, due to the presence of relevant outliers and/or, in the specific case of count data, due to high levels of expositions for a quantitative variable with discrete and non-negative values For instance: counts per month instead of counts per day, or counts per square kilometer instead of square meter, can generate overdispersion in the data.

The estimation of the parameters of equation (1), in the presence of overdispersion, can also be estimated by maximum likelihood [3, 12], and the log likelihood function for negative binomial regression models is

$$LL = \sum_{i=1}^{n} \left[y_i \ln\left(\frac{\alpha.\widehat{u}_i}{1+\alpha.\widehat{u}_i}\right) - \frac{\ln(1+\alpha.\widehat{u}_i)}{\alpha} + \ln\Gamma(y_i+\alpha^{-1}) - \ln\Gamma(y_i+1) - \ln\Gamma(\alpha^{-1}) \right] \quad (9)$$

Therefore, the estimation of the parameters passes through the previous definition of the statistical significance of the α term. For this, Cameron and Trivedi [5] propose a test to verify the existence of overdispersion in the dependent variable, conditional on explanatory variables, in which there is a need for a previous estimation of a Poisson regression model.

In the following sections, we will formally discuss this test, as well as introduce a new Stata command (`overdisp`), which can be implementd without the necessity of previously estimating Poisson models and, thus, contributes to the identification of overdispersion in the data since it detects the phenomenon faster and easier.

3. Overdispersion test

The formal test of the null hypothesis of equidispersion, $Var(y|X) = E(y|X)$, against the alternative of overdispersion, was firstly introduced by Cameron and Trivedi [4], and is based on the following equation:

$$Var(y|X) = E(y|X) + \alpha.[E(y|X)]^2 \quad (10)$$

which is the variance function for the negative binomial distribution, as shown in equation (8). So, we have to test the significance of the parameter α ($H_1 : \alpha > 0$) against the null hypothesis ($H_0 : \alpha = 0$).

According to the authors, to implement the test, firstly a new variable y^* needs to be generated, as follows:

$$y^* = [(y - \widehat{u})^2 - y]/\widehat{u} \quad (11)$$

where $\widehat{u} = \exp(X'\widehat{\beta})$; $\widehat{\beta}$ represents the vector of parameters to be estimated through the model presented by equation (1).

The test can be implemented by a regression of y^* on $\widehat{u}$, without an intercept term. The t test of the coefficient of $\widehat{u}$ indicates the presence of significant overdispersion ($H_0 : P > |t| >$ significance level → equidispersion, i.e, no significant overdispersion, favoring Poisson estimation; $H_1 : P > |t| \leq$ significance level → significant overdispersion, favoring negative binomial estimation).

4. The `overdisp` Stata command

Cameron and Trivedi [5] presented the overdispersion test in Stata through the application of four commands in sequence, following the logic discussed in the previous section. These commands are as follows:

```
poisson depvar [indepvar]
```

which estimates a Poisson regression model. The term `depvar` denotes the dependent or response variable and `indepvar` denotes the list of covariates appearing in the model. The following command generates the predicted number of events ($\widehat{u}$, called `uhat` in Stata):

```
predict uhat
```

Based on equation (11), next command creates the a variable (y^*, called `ystar` in Stata), as follows:

```
generate ystar = ((depvar-uhat)^2 - depvar)/uhat
```

Finally, following the authors, an auxiliary simple regression of y^* (ystar) on $\widehat{u}$ (uhat), without an intercept, can be estimated:

```
regress ystar uhat, noconstant
```

The overdisp command consolidates these four commands, and was developed through a Mata code, the matrix programming language available in Stata since its version 9. Appendix 1 offers the Stata and Mata ado code for the overdisp command.

4.1. Instalation

overdisp is available from Stata 15 and can be installed from the Statistical Software Components (SSC) archive [8] by typing the following command within a net-aware version of Stata:

```
ssc install overdisp
```

Two files will be installed on your computer: overdisp.ado, a Stata ado file which defines the command; and overdisp.sthlp, a Stata help file which documents the command. Note that these files will be installed onto your adopath, the path where Stata searches for the files it needs. If you have already installed overdisp from the SSC, you can check that you are using the latest version by typing the following command:

```
adoupdate overdisp
```

The overdisp command follows standard Stata syntax for estimation commands. We restrict our discussion here to the most common options. A complete description is provided in the overdisp help file. You may type the following at any point to view this help file:

```
help overdisp
```

4.2. Syntax

The overdisp command has the following syntax:

```
overdisp depvar [indepvar] [, level(#)]
```

where overdisp is the name of the command, depvar denotes the dependent or response variable, indepvar denotes the list of covariates appearing in the model, the square brackets indicate optional arguments and the comma separates the specification of the model from the specification of any modeling or estimation options listed in options.

4.3. Option

The options are defined as follows: level(#) sets confidence level and the default is level(95). H0 indicates equidispersion.

5. Examples

We illustrate the application of the overdisp command using the two examples below, in which we show the use of the new command and the interpretation of its output in different settings. Given the illustrative purpose of this section, we closely follow the sources of the examples [5, 7] when describing the data and discussing the possible overdispersion in the dependent variable, conditional on explanatory variables, of the proposed models. We contribute by proposing an alternative, direct way to test overdispersion.

5.1. Example 1: Determinants of annual number of doctor visits

In the first example, we use the dataset `mus17data.dta` when describing the data and discussing the existence of overdispersion in the dependent variable, conditional on explanatory variables, replicating part of results from chapter 17 of Cameron and Trivedi [5].

The application we consider here is the analysis of the determinants of annual number of doctor visits (`docvis`) of the Medicare population aged 65 and higher from the sample of the U.S. Medical Expediture Panel Survey for 2003. The covariates include age (`age`), squared age (`age2`), years of education (`educyr`), presence of activity limitation (`actlim`), number of chronic conditions (`totchr`), having private insurance that supplements Medicare (`private`), and having public Medicaid insurance for low-income individuals that supplements Medicare (`medicaid`).

Thus, the proposed count-data model here is

$$\ln(\widehat{docvis_i}) = \beta_0 + \beta_1.age_i + \beta_2.age2_i + \beta_3.educyr_i + \beta_4.actlim_i + \beta_5.totchr_i + \beta_6.private_i + \beta_7.medicaid_i \tag{12}$$

and our objective is verify the existence of overdispersion in the dependent variable `docvis`, conditional on the seven explanatory variables.

We start by loading the data. We do this using the use command where we specify the `clear` option to replace any data should they currently exist in memory.

```
use http://www.stata-press.com/data/mus/mus17data, clear
```

We then use the `codebook` command with the `compact` option to compactly describe the data contents [13].

```
codebook docvis age age2 educyr actlim totchr private medicaid, compact
```

```
Variable     Obs Unique      Mean   Min    Max  Label
-----------------------------------------------------------------------------------
docvis      3677     53  6.822682     0    144  # doctor visits
age         3677     26  74.24476    65     90  Age
age2        3677     26  5552.936  4225   8100  Age-squared
educyr      3677     18  11.18031     0     17  Years of education
actlim      3677      2   .333152     0      1  =1 if activity limitation
totchr      3677      9  1.843351     0      8  # chronic conditions
private     3677      2  .4966005     0      1  =1 if has private supplementary insurance
medicaid    3677      2   .166712     0      1  =1 if has Medicaid public insurance
-----------------------------------------------------------------------------------
```

The dependent variable *docvis* is quantitative, discrete and has non-negative values. As such, the `tab` command, which is frequently used to obtain the distribution frequencies for a qualitative variable, can be used in this case, given that the dependent variable presents nonnegative integer values.

```
tab docvis
```

(output omitted)

Next command, `hist`, offers the opportunity to see the histogram for the dependent variable, presented in Figure 1. The term `discrete` informs that the dependent variable presents only integer values.

```
hist docvis, discrete freq
```

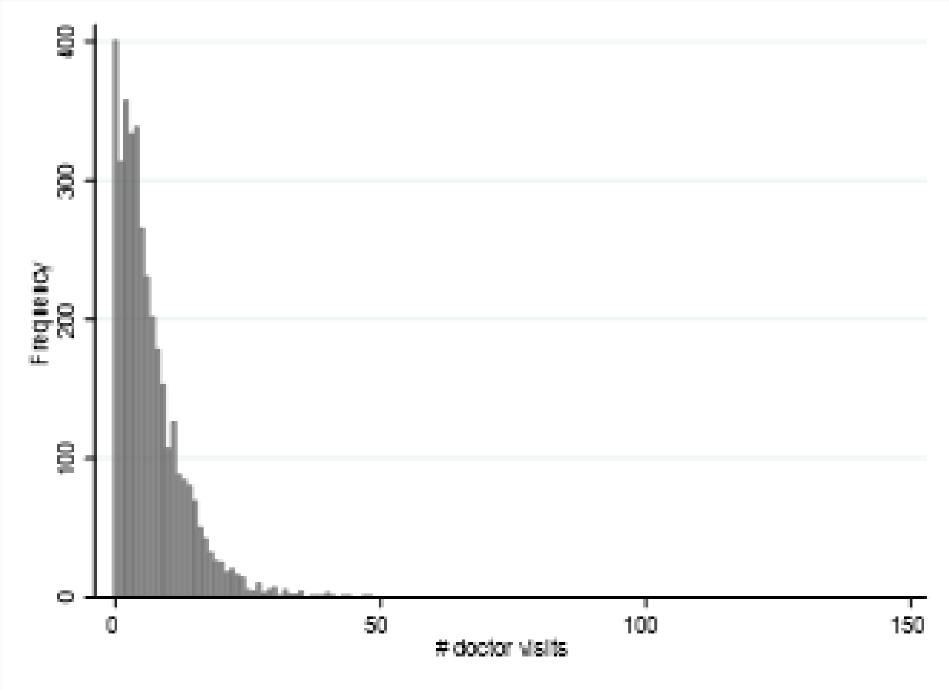

Figure 1. Histogram of dependent variable *docvis*.

Before preparing any regression model for count-data, it is interesting for the user to evaluate if the mean and variance of the dependent variable are equal, or at least close. This will give an idea as to if the Poisson regression model estimation is adequate, or if it will be necessary to estimate a negative binomial regression model. Typing in the following command will allow that this initial diagnostic be performed.

```
tabstat docvis, stats(mean var)
```

```
variable |  mean      variance
----------------------------------
docvis   |  6.822682  54.68509
----------------------------------
```

As we can see, the variance of the dependent variable is about 8 times greater than its mean. Although this fact suggests the existence of overdispersion in the data, until this moment we do not have conditions to verify this phenomenon in the dependent variable, conditional on the explanatory variables.

Cameron and Trivedi [5] recommend that all modeling where the dependent variable contains count data be started by means of estimating a Poisson regression model. To do this, we will type the following commands [†]:

```
global xlist private medicaid age age2 educyr actlim totchr
```

which defines the vector of explanatory variables. To estimate the Poisson regression model, we have to type:

```
poisson docvis $xlist
```

(output omitted)

[†] The presented commands are largely based on Cameron and Trivedi [5].

The `poisson` command estimates a Poisson regression model by maximum likelihood. Just as for multiple, binary and multinomial logistic regression models, if the researcher does not inform the desired confidence level for the estimated parameter interval definition, that standard will be 95%. The following command generates the predicted number of events, $\widehat{u}$:

```
predict uhat
```

Next, based on equation (11), we now have to create a new variable in the database, which is called *ystar*, according to what follows:

```
generate ystar = ((docvis-uhat)^2 - docvis)/uhat
```

Finally, we have to estimate the auxiliary simple regression model $y_i^* = \gamma.\widehat{u}_i$, by means of typing the following command:

```
regress ystar uhat, noconstant
```

```
------------------------------------------------------------------------------
ystar |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
uhat |   .7047319   .1035926     6.80   0.000     .5016273    .9078365
------------------------------------------------------------------------------
```

The outcome indicates the presence of significant overdispersion in the dependent variable, conditional on explanatory variables ($P > |t| = 0.000$).

The new command `overdisp` consolidates the last four commands and, thus, contributes by proposing an alternative, direct way to test overdispersion. To do so, we have to type the following command after the defition of the vector of explanatory variables (command `global xlist`):

```
overdisp docvis $xlist
```

```
Overdispersion test (H0: equidispersion)        Number of obs     =      3,677
------------------------------------------------------------------------------
docvis |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
------------+-----------------------------------------------------------------
uhat   |   .7047319   .1035926     6.80   0.000     .5016273    .9078365
------------------------------------------------------------------------------
```

Although this result causes researchers to estimate a Poisson model with the `vce(robust)` option, we recommend to model this feature using the negative binomial model.

5.2. *Example 2: Determinants of annual number of durable good purchases through installment closed-end credit*

The second application we consider is Fávero and Belfiore's [7] analysis of the determinants of the quantity of durable good purchases made using installment closed-end credit in the last year per consumer (`purchases`).

As the finance department for a large appliance retailer wants to know if consumer income and age explain the use of financing when purchasing goods such as cellular telephones, tablets, laptops, televisions, videogames, DVD/Blu-ray players and etc., to develop a marketing campaign for this form of financing based on customer profile, a survey was conducted on a random sample of 200 clients. Thus, covariates include the monthly consumer income in US$ (`income`) and the consumer age in years (`age`). We use the dataset `Financing.dta` when describing the data and discussing the existence of overdispersion in the dependent variable, conditional on the covariates, replicating results from chapter 15 of Fávero and Belfiore [7].

The proposed count-data model here is

$$\ln(\widehat{purchases_i}) = \beta_0 + \beta_1.income_i + \beta_2.age_i \tag{13}$$

and now our objective is verify the existence of overdispersion in the dependent variable purchases, conditional on the two explanatory variables.

Firstly, we can use the `codebook` command with the compact option to compactly describe the data, as follows:

```
codebook purchases income age, compact

Variable    Obs Unique    Mean    Min    Max  Label
----------------------------------------------------------------
purchases 200       5   1.02      0      4  amount of durable good
                                            purchases made using
                                            installment closed-end
income    200      12   2970   1500   3600  monthly consumer income
                                            (US$)
age       200      18  45.13     36     54  consumer age (years)
----------------------------------------------------------------
```

The dependent variable is quantitative, discrete and has non-negative values. In this case, as discussed in the previous example, we can use `tab` and `hist` commands, as follows:

```
tab purchases
```

(output omitted)

```
hist purchases, discrete freq
```

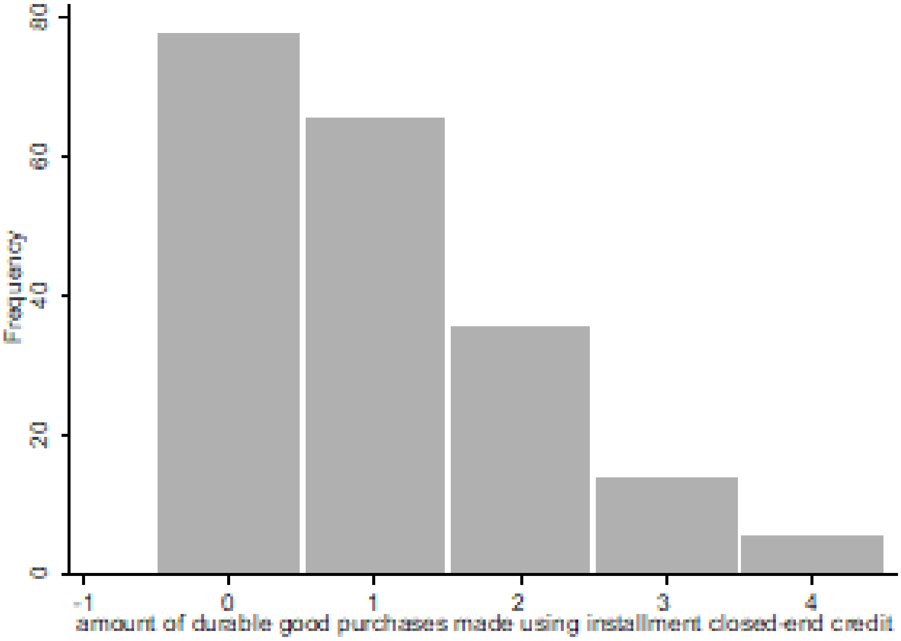

Figure 2. Histogram of dependent variable *purchases*.

The following command allows us to investigate preliminarily if the mean and variance of the dependent variable are equal, or at least close.

```
tabstat purchases, stats(mean var)
```

```
    variable |  mean  variance
-------------+--------------------
purchases    |  1.02  1.125226
----------------------------------
```

By means of analyzing the mean and variance, which are quite close, we can suppose that the Poisson regression model will be suitable in this case.

Following again the logic proposed by Cameron and Trivedi [5], lets firstly estimate the Poisson regression model and, in the sequence, investigate the existence of overdispersion in the data, typing the four following commands:

```
poisson purchases income age
```

(output omitted)

```
predict uhat
```

Based on equation (11), we can generate the new variable *ystar*, according to what follows:

```
generate ystar = ((purchases-uhat)^2 - purchases)/uhat
```

Estimating the auxiliary simple regression model $y_i^* = \gamma.\widehat{u}_i$, by means of typing the following command, we obtain the next output:

```
regress ystar uhat, noconstant
```

```
------------------------------------------------------------------------------
ystar |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
uhat  |  -.1942878   .1174778    -1.65   0.100    -.4259489    .0373734
------------------------------------------------------------------------------
```

The t test of the coefficient of $\widehat{u}$ indicates the equidispersion, since $P > |t| > 0.05$, favoring Poisson estimation.

The `prcounts` command, to be typed after the `poisson` command, allows variables corresponding to the occurrence probabilities for each possibility of the dependent variable to be generated for each observation. In the case the `prcounts` command has not been installed in Stata, the researcher should type in `findit prcounts` and install it in the statistical package. The command is:

```
prcounts prpoisson, plot
```

Variables that correspond, respectively, to the occurrence of 0 to 9 observed and predicted probabilities for the whole sample (*prpoissonobeq* and *prpoissonpreq*) are created. Finally, the *prpoissonval* variable presents the actual values of 0 to 9 (Stata default) that will be related to the observed and predicted probabilities. The following command allows the observed probabilities and occurrence predictions from 0 to 9 to be visually compared:

```
graph twoway (scatter prpoissonobeq prpoissonpreq prpoissonval, connect (1 1))
```

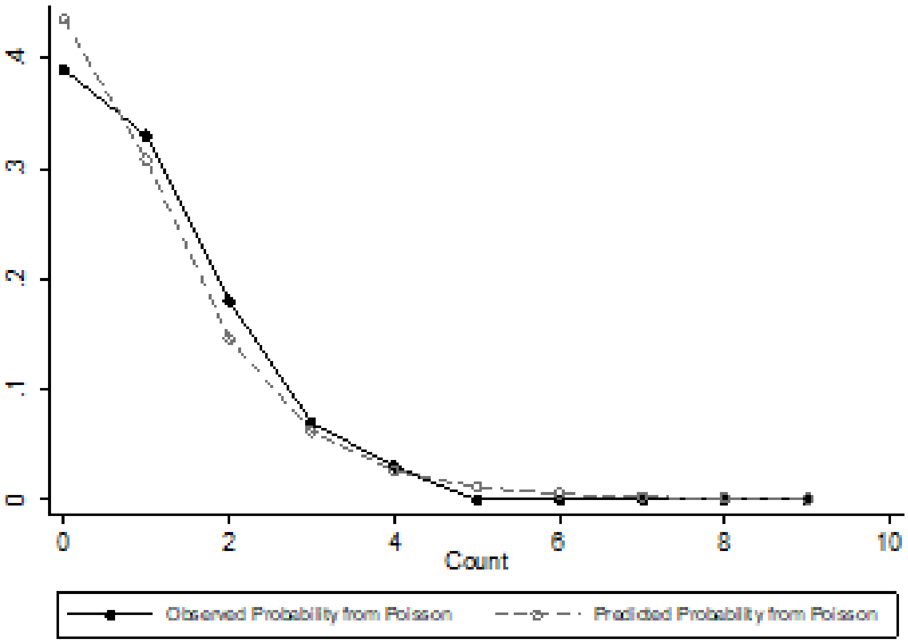

Figure 3. Distributions of observed and predicted probabilities of occurrence of 0 to 9 durable good purchases made using installment closed-end credit.

To verify the quality of the final adjusted estimated model (goodness-of-fit), we can perform an χ^2 test to compare the two curves presented in Figure 3. Thus, after estimating the Poisson model, we should type:

```
poisgof
```

```
----------------------------
Goodness-of-fit  = 166.8808

Prob > chi2(197) =   0.9416
----------------------------
```

The result shows the quality of the final Poisson regression model, or rather, that there are no statistically significant differences between the predicted and observed values for the number of durable good purchases made using installment closed-end credit.

The `overdisp` command makes unnecessary the use of the four commands proposed by Cameron and Trived [5], and even the `prcounts` and `poisgof` commands, since it directly identifies overdispersion in the data, without the need to previously estimate a specific type of count-data model, and consequently, allows researchers to propose correct models, such Poisson or negative binomial, directly.

6. Simulation study

In this section, we present a simulation study to show the consistency of the overdispersion test using the `overdisp` command, from the generation of dependent variables with Poisson or negative binomial distributions, as a function of two random explanatory variables.

We analyze 1,0000,000 replications of 25,000 observations. On each replication, we generate observations responses, y_i, and two X_i expanatory variables which are standard normal variates defined at the observation i. Appendix 2 presents the Stata codes for the proposed simulations.

We can verify that 5.24% of the simulations with the dependent variable presenting Poisson distribution indicated the existence of overdispersion in the data at 5% of significance level (8.83% of these simulations indicated existence of overdispersion in the dependent variable, conditional to the explanatory variables, at 10% of significance level - already including the previous 5.24%). On the other hand, no simulation rejected the hypothesis of overdispersion in the dependent variable when it assumes a negative binomial distribution. Table 1 summarizes the findings of our simulations.

Table 1. Simulation results - consistency of the overdispersion test for dependent variables with Poisson and negative binomial distributions.

Distribution of the Dependent Variable	**Overdispersion**	**5% of Significance Level**	**10% of Significance Level**
Poisson	No	947,593	911,747
	Yes	52,407	88,253*
	Total	1,000,000	1,000,000
Negative Binomial	No	0	0
	Yes	1,000,000	1,000,000
	Total	1,000,000	1,000,000

* Includes the 52,407 cases in which the test did not reject the hypothesis of existence of overdispersion at the significance level of 5%.

Simulation results indicate that, although the overdispersion test proposed here through the new Stata command `overdisp` is quite efficient in the occurence of such phenomenon for dependent variables with negative binomial distribution, it can still generate distortions in the interpretation of the results in cases in which the dependent variable follows a Poisson distribution.

This finding represents an important contribution, i.e., if the test indicates equidispersion in the data, there are consistent evidence that the distribution of the dependent variable is, in fact, Poisson. On the other hand, if the test indicates overdispersion in the data, the researcher should investigate more deeply whether the dependent variable actually exhibits better adherence to the Poisson-Gamma distribution or not.

7. Conclusion

We have presented the `overdisp` command in Stata, for count-data regression models. Through specication of Poisson or negative binomial models, we have illustrated how to implement directly the overdispersion test. In essence, the new command `overdisp` contributes to the identification of overdispersion in the data since it consolidates the four commands presented by Cameron and Trivedi [5], what renders the process of detecting overdispersion in count-data models faster and easier.

Compared with `chi2gof` [15], `overdisp` can be implementd without the necessity of previously estimating Poisson models, and can be used even if negative binomial models prevail. In other words, while `chi2gof` compares differences between actual frequencies of the dependent variable with its predicted frequencies obtained through the proposed model after the estimation of a Poisson or a negative binomial model, researchers can apply the `overdisp` command without the necessity of estimating Poisson or negative binomial models. The use of the `overdisp` command also makes unnecessary the use of the `prcounts` and `poisgof` commands, as shown in Section 5.2.

Finally, it is important to mention that the choice of the significance level of the overdispersion test is left to the user, but an increasing significance level may cause distortions in the interpretation of the results for dependent variables with Poisson distribution.

In future developments we aim to allow `overdisp` for zero-inflated and Cragg hurdle regression models, as weel as the extension to incorporate underlying groups with different overdispersion behaviors, such as finite mixture models. The package is available from the Statistical Software Components (SSC) archive [8] and can be installed from Stata by typing `ssc install overdisp overdisp`.

Acknowledgement

We are especially grateful to Enrique Pinzon (StataCorp) for providing us with helpful suggestions and support. All errors are our own. We also thank Professor A. Colin Cameron (University of California, Davis) and Professor Pravin K. Trivedi (Indiana University, Bloomington) for allowing us to use a dataset from the Microeconometrics Using Stata (2010).

APPENDIX 1 - Stata and Mata ado code for `overdisp` command

Below we present the ado code for the `overdisp` command:

```
program define overdisp, eclass byable(onecall)
       version 15

        if _by() {
        local BY ‘"by ‘_byvars’‘_byrc0’:"’
        }

        ‘BY’ _vce_parserun overdisp, noeqlist jkopts(eclass): ‘0’

        if "‘s(exit)’" != "" {
        ereturn local cmdline ‘"overdisp ‘0’"’
        exit
        }

        if replay() {
                if ‘"‘e(cmd)’"’ != "overdisp" {
                error 301
                }
                else if _by() {
                error 190
                }
                else {
                Display ‘0’
                }
                exit
        }
        ‘vv’ ///
        ‘BY’ Estimate ‘0’
        ereturn local cmdline ‘"overdisp ‘0’"’

end

program define Estimate, eclass byable(recall)
 syntax varlist(numeric ts min=2 fv) [if] [in], [ Level(cilevel) * ]

 // indicator for [if] and [in] conditions

 marksample touse
 // Parsing display options

 _get_diopts diopts rest, ‘options’
 qui cap Display, ‘diopts’ ‘rest’
 if _rc==198 {
                Display, ‘diopts’ ‘rest’
        }

 // Getting depvar, indepvars, and computing test

 gettoken depvar indepvars: varlist
 tempname b V df
 mata: _overdisp_work("‘depvar’", "‘indepvars’", "‘b’", "‘V’", ///
        "‘df’", "‘touse’")

 // Doing work to post results using coeftable

 quietly count if ‘touse’
 local N = r(N)
 matrix rownames ‘b’ = uhat
 matrix colnames ‘b’ = uhat
 matrix rownames ‘V’ = uhat
 matrix colnames ‘V’ = uhat

 local dff = ‘df’
 ereturn post ‘b’ ‘V’, esample(‘touse’) obs(‘N’)  ///
             depname(‘depvar’) dof(‘dff’) buildfvinfo

 ereturn local cmd "overdisp"
 Display, bmatrix(e(b)) vmatrix(e(V)) ‘rest’ ‘diopts’ level(‘level’)
end

program define Display
        syntax [, bmatrix(passthru) vmatrix(passthru) *]

        _get_diopts diopts other, ‘options’
        local myopts ‘bmatrix’ ‘vmatrix’

        if "‘other’"!=""{
        display "{err}option ‘other’ not allowed"
        exit 198
        }
 _coef_table_header, title(Overdispersion test (H0: equidispersion))
 _coef_table, ‘diopts’ ‘myopts’

 end

 mata:
 void _overdisp_work (
  string scalar yvar,///
  string scalar xvars,///
  string scalar beta,///
  string scalar Var,  ///
  string scalar df,///
  string scalar touse)
{
  real scalar n, cha, iter, k, tmle, pmle, n2, k2
  real vector y, b, u, sco, bold, bmle, ystar, b2, e2, s2
  real matrix X, hes, gradmatrix, Vmle, semle, V2, se2

  st_view(y=., ., yvar, touse)
  st_view(X=., ., tokens(xvars), touse)
  X = X, J(rows(X), 1, 1)
  b = invsym(X’X)*X’(ln(y+(y:==0)*0.1))
  n = rows(X)
  cha = 1
  iter = 1
  do {
  u = exp(X*b)
  sco = (X’(y-u))/n
  hes = -(X’(X:*u))/n
  bold = b
 b = bold + invsym(-hes)*sco
  cha = (bold-b)’(bold-b)/(b’b)
  iter = iter + 1
  } while (cha > 1e-16)
  bmle = b
  k = cols(X)
  gradmatrix = X:*(y-u)
  Vmle = invsym(-n*hes)*(gradmatrix’gradmatrix)*invsym(-n*hes)*(n/(n-k))
  semle = sqrt(diagonal(Vmle))
  tmle = bmle:/semle
  pmle = 2*ttail(n-k, abs(tmle))
  ystar=(((y-u):^2)-y):/(u)
  b2 = invsym(u’u)*u’ystar
  e2 = ystar - u*b2
  n2 = rows(u)
  k2 = cols(u)
  s2 = (e2’e2)/(n2-k2)
  V2 = s2*invsym(u’u)
  se2 = sqrt(diagonal(V2))
  st_matrix(beta, b2)
  st_matrix(Var, V2)
  st_numscalar(df, n2-k2)
}
end
```

APPENDIX 2 - Stata codes for the simulations

We start by clearing any existing data from memory and by using the `forvalues` command to execute the commands enclosed within its braces 1,000,000 times.

```
. clear
. forvalues r = 1/1000000 {
2. display "Replication = ‘r’"
3. clear
4. set obs 25000
5. generate x1 = rnormal(0, 1)
6. generate x2 = rnormal(0, 1)
7a. genpoisson y, adoonly
8. overdisp y x1 x2
9. predict uhat
10. matrix b = e(b)
11. matrix V = e(V)
12. svmat b
13. format %4.3f b
14. list, abbreviate(9) separator(0)
15. }

. forvalues r = 1/1000000 {
2. display "Replication = ‘r’"
3. clear
4. set obs 25000
5. generate x1 = rnormal(0, 1)
6. generate x2 = rnormal(0, 1)
7b. gennbreg y, adoonly
8. overdisp y x1 x2
9. predict uhat
10. matrix b = e(b)
11. matrix V = e(V)
12. svmat b
13. format %4.3f b
14. list, abbreviate(9) separator(0)
15. }
```

(output omitted)

Line 2 of the loops displays the current replication number so that the user can see how far the simulation study has progressed. Line 3 clears any existing data from memory. Line 4 specifies that the new dataset will have 25,000 observations. Lines 5 and 6 generate standard normal explanatory variables. While line 7a (first loop) generates dependent variables following theoretical Poisson distribution, line 7b (second loop) creates dependent variables with the negative binomial distribution. Line 8 applies the `overdisp` command to the simulated data. Line 9 predicts the values of $\widehat{u}$. Lines 10 and 11 create coefficient vectors and their associated variance-covariance matrices. Line 12 takes the parameter estimates and stores their values as new variables. Line 13 sets the display precision of the newly created variables to three decimal places. Line 14 lists the dataset of the results. Line 15 closes the loop.

REFERENCES

1. Z. Y. Algamal, *Variable selection in count data regression model based on firefly algorithm*, Statistics, Optimization and Information Computing , vol. 7, pp. 520–529, 2019.
2. E. Avci, *Flexiblity of using Com-Poisson regression model for count data*, Statistics, Optimization and Information Computing , vol. 6, pp. 278–285, 2018.
3. A. C. Cameron, and P. K. Trivedi, *Econometric models based on count data: comparisons and applications of some estimators and tests*, Journal of Applied Econometrics, vol. 1, no. 1, pp. 29–53, 1986.
4. A. C. Cameron, and P. K. Trivedi, *Microeconometrics: Methods and Applications*, Cambridge University Press, New York, 2005.
5. A. C. Cameron, and P. K. Trivedi, *Microeconometrics using Stata*, Stata Press, College Station, 2010.
6. A. C. Cameron, and P. K. Trivedi, *Regression Analysis of Count Data*, Cambridge University Press, Cambridge, 2013.
7. L. P. Fávero, and P. Belfiore, *Data Science for Business and Decision Making*, Academic Press Elsevier, Cambridge, 2019.
8. L. P. Fávero, and P. Belfiore, overdisp: module to detect overdispersion in count-data models using Stata, Statistical Software Components, Boston College Department of Economics. https://ideas.repec.org/c/boc/bocode/s458496.html, 2018.
9. L. P. Fávero, M. A. Santos, and R. G. Serra, *Cross-border branching in the Latin American banking sector*, International Journal of Bank Marketing , vol. 36, no. 3, pp. 496–528, 2018.
10. W. Gardner, E. P. Mulvey, and E. C. Shaw, *Regression analyses of counts and rates: Poisson, overdispersed Poisson, and negative binomial models*, Psychological Bulletin, vol. 118, no. 3, pp. 392–404, 1995.
11. S. Gurmu, *Tests for detecting overdispersion in the positive Poisson regression model*, Journal of Business & Economic Statistics, vol. 9, no. 3, pp. 215–222, 1991.
12. J. A. Hausman, B. H. Hall, and Z. Griliches, *Econometric models for count data with an application to the patents-R & D relationship*, Econometrica , vol. 52, no. 4, pp. 909–938, 1984.
13. G. Leckie, *runmixregls: a program to run the mixregls mixed-effects location scale software from within Stata*, Journal of Statistical Software, vol. 59, pp. 1–41, 2014.
14. J. S. Long, and J. Freese, *Regression Models for Categorical Dependent Variables using Stata*, Stata Press, College Station, 2006.
15. M. Manjn, and O. Martłnez, *The chi-squared goodness-of-fit test for count-data models*, Stata Journal, 14, pp. 798–816, 2014.
16. R Core Team, *R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing*, Vienna, Austria. http://www.R-project.org/, 2016.
17. S. Rabe-Hesketh, and A. Skrondal, *Multilevel and Longitudinal Modeling using Stata: Categorical Responses, Counts, and Survival* Stata Press, College Station, 2012.
18. SAS Institute Inc, *SAS/STAT Software, Version 9.22*, Cary, NC, http://www.sas.com/, 2018.
19. StataCorp, *Stata Data Analysis Statistical Software, Release 15*, College Station, TX, http://www.stata.com/, 2018.
20. H. Zhang, Y. Liu, and B. Li, *Notes on discrete compound Poisson model with applications to risk theory*, Insurance: Mathematics and Economics, vol. 59, p. 325–336, 2014.
21. M. L. Zwilling, *Negative binomial regression*, The Mathematica Journal, vol. 15, pp. 1–18, 2013.