Regression Models for Categorical Dependent Variables Using Stata

Third Edition

J. SCOTT LONG
Departments of Sociology and Statistics
Indiana University
Bloomington, Indiana

JEREMY FREESE
Department of Sociology and Institute for Policy Research
Northwestern University
Evanston, Illinois

A Stata Press Publication
StataCorp LP
College Station, Texas

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in LaTeX 2ε
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN-10: 1-59718-111-0
ISBN-13: 978-1-59718-111-2

Library of Congress Control Number: 2014948009

(Pages omitted)

Contents

(Pages omitted)

Preface

As with previous editions, our goal in writing this book is to make it routine to carry out the complex calculations necessary to fully interpret regression models for categorical outcomes. Interpreting these models is complex because the models are nonlinear. Software packages that fit these models often do not provide options that make it simple to compute the quantities that are useful for interpretation; when they do provide these options, there is usually little guidance as to how to use them. In this book, we briefly describe the statistical issues involved in interpretation and then show how you can use Stata to make these computations.

While our purpose remains the same, this third edition is an almost complete rewrite of the second edition—almost every line of code in our SPost commands has been rewritten. Advances in computing and the addition of new features to Stata has expanded the possibilities for routinely applying more sophisticated methods of interpretation. As a result, ideas we noted in previous editions as good in principle are now much more straightforward to implement in practice. For example, while you could compute average marginal effects using commands discussed in previous editions, it was difficult and few people did so (ourselves included). Likewise, in previous editions, we relegated methods for dealing with nonlinearities and interactions on the right-hand side of the model to the last chapter, and our impression was that few readers took advantage of these ideas because they were comparatively difficult and error-prone to use.[1]

These limitations changed with the addition of factor variables and the `margins` command in Stata 11. It took us personally quite a while to fully absorb the potential of these powerful enhancements and decide how best to take advantage of them. Plus, Stata 13 added several features that were essential for what we wanted to do.

This third edition considers the same models as the second edition of the book. We still find these to be the most valuable models for categorical outcomes. And, as in previous editions, our discussion is limited to models for cross-sectional data. While we would like to consider models for panel data and other hierarchical data structures, doing so would at least double the size of an already long book.

1. Those who have read previous editions will note that this last chapter has been dropped entirely. In addition to covering linked variables of the right-hand side, that chapter also discussed adapting our commands to other estimation commands; however, this is now obsolete because `margins` works with most estimation commands. We also dropped the section on working effectively in Stata because Long's (2009) *Workflow of Data Analysis Using Stata* covers these topics in detail.

We note, however, that many of our SPost commands—such as `mtable`, `mgen`, and `mchange` (hereafter referred to as the `m*` commands)—are based on `margins` and can be used with any model that is supported by `margins`. This is a substantial change from our earlier `prchange`, `prgen`, `prtab`, and `prvalue` commands, which only worked with the models discussed in the book. A second major improvement is that our `m*` commands work with weights and survey estimation, because these are supported by `margins`.

SPost was originally developed using Stata 4 and Stata 5. Since then, our commands have often been enhanced to use new features in Stata. Sometimes these enhancements have led to code that was not as efficient, robust, or elegant as we would have liked. In SPost13, we rewrote much of the code, incorporated better returns, improved output, and removed obscure or obsolete features.

How to cite

Our commands are not officially part of Stata. We have written them in an effort to contribute to the community of researchers whose work involves extensive use of the models we cover. If you use our commands or other materials in published work, we ask that you cite our work in the same way that you cite other useful sources. We ask that you simply cite the book rather than providing different citations to different commands:

> Long, J. S., and J. Freese. 2014. *Regression Models for Categorical Dependent Variables in Stata*. 3rd ed. College Station, TX: Stata Press.

Thanks

Hundreds of people have contributed to our work since 2001. We cannot possibly mention them all here, but we gratefully acknowledge them for taking the time to give us their ideas. Thousands of students have taken classes using previous editions of the books, and many have given us suggestions to make the book or the commands more effective.

In writing this third edition, several people deserve to be mentioned. Ian Anson and Trent Mize tested commands and provided comments on draft chapters. Tom VanHeuvelen ran labs in two classes that used early versions of the commands, helped students work around bugs, made valuable suggestions on how to improve the commands, provided detailed comments on each chapter, was a sounding board for ideas, and was an exceptional research assistant. Rich Williams gave us many suggestions that improved the book and our commands. He has a (sometimes) valuable gift for finding bugs as well. Scott Long gratefully acknowledges the support provided by the College of Arts and Sciences at Indiana University.

People at StataCorp provided their expertise in many ways. More than this, though, we are grateful for their engagement and support of our project. Jeff Pitblado was enor-

mously helpful as we incorporated factor variables and `margins` into our commands. His advice made our code far more compact and reliable. Vince Wiggins provided valuable advice on our graphing commands and helped us understand `margins` better. Lisa Gilmore, as always, did a great job moving the book from draft to print. Most importantly, discussions with David Drukker stimulated our thinking about a new edition and, as always, asked challenging questions that made our ideas better.

Illinois and Indiana
August 2014

Jeremy Freese
Scott Long

(Pages omitted)

ual for more information on these types of models. Likewise, we do not consider any models for panel or other multilevel data, even though Stata contains commands for fitting these models. For additional information, see Rabe-Hesketh and Skrondal (2012), Cameron and Trivedi (2010), and the *Stata Longitudinal-Data/Panel-Data Reference Manual*.

1.3 Whom is this book for?

We expect that readers of this book will vary considerably in their knowledge of both statistics and Stata. With this in mind, we have tried to structure the book to accommodate the diversity of our audience. Minimally, however, we assume that you have a solid familiarity with the linear regression model and that you are comfortable using the basic features of the operating system of your computer. Although we have provided sufficient information about each model so that you can read each chapter without prior exposure to the models discussed, we strongly recommend that you do not use this book as your sole source of information on the models (see section 1.8 for reading recommendations). Our book will be most useful if you have already studied or are studying the models considered herein in conjunction with reading our book.

Ideally, you are running Stata 13 or later. Most of our examples will, however, run in Stata 11 and 12. If you are using a version of Stata earlier than Stata 11, we suggest that you instead use the second edition of our book (Long and Freese 2006). However, with the powerful new features in Stata 13 and the new methods of interpretation in this third edition, we hope you decide instead to upgrade your software. To make the most out of the book, you will need access to the Internet to download our commands, datasets, and sample programs (see section 1.5 for details). For information about obtaining Stata, see the StataCorp website at http://www.stata.com.

1.4 How is the book organized?

Chapters 2–4 introduce materials that are essential for working with the models we present in the later chapters:

Chapter 2: Introduction to Stata reviews the basic features of Stata that are necessary to get new or inexperienced users up and running with the program. New users should work through the brief tutorial that we provide in section 2.18. This introduction is by no means comprehensive, so we include information on how to learn more about using Stata. Those who are familiar with Stata can skip this chapter, although even these readers might benefit from scanning it.

Chapter 3: Estimation, testing, and fit reviews Stata commands for fitting models, testing hypotheses, and computing measures of model fit. Those who regularly use Stata for regression modeling might be familiar with much of this material; however, we suggest at least a quick review of the material. Most importantly,

you should read our detailed discussion of factor-variable notation, which was introduced in Stata 11. Understanding how to use factor variables is essential for the methods of interpretation presented in the later chapters.

Chapter 4: Methods of interpretation is an overview of various approaches to interpreting regression models. We introduce the `margins` command that is part of official Stata and the `mtable`, `mgen`, and `mchange` commands that are part of SPost13. This chapter is essential background before proceeding to part II. Study this chapter carefully, even if you are an advanced user. Readers new to Stata are likely to find that this chapter has more detail than initially needed; therefore, throughout the chapter, we suggest which sections you may wish to only skim on first reading.

Part II covers regression models for different types of outcomes.

Chapters 5 and 6: Models for binary outcomes begin with an overview of how the binary logit and probit models are derived and how they can be fit. After the model has been fit, we show how to test hypotheses, compute residuals and influence statistics, and calculate scalar measures of model fit. Chapter 6 uses postestimation commands that assist in interpretation using predicted probabilities, discrete and marginal change in the predicted probabilities, and for the logit model, odds ratios. Because binary models provide a foundation on which many models for other kinds of outcomes are derived, and because these two chapters provide more detailed explanations of common tasks than later chapters do, we recommend reading these chapters carefully even if you are interested mainly in another type of outcome.

Chapter 7: Models for ordinal outcomes presents the ordered logit and ordered probit models. We show how these models are fit and how to test hypotheses about coefficients. We also consider tests of a key assumption of both models, known as the parallel regression assumption. For interpreting results, we discuss methods similar to those described in chapter 6, and we also discuss interpretation in terms of a latent dependent variable. Methods of interpretation using predicted probabilities apply directly to models for nominal outcomes, so it is useful to familiarize yourself with these methods before proceeding to chapter 8. This chapter also details the implications of assuming that an ordinal model is appropriate for your outcome and recommends that you use models for nominal outcomes as part of your evaluation of ordinal models.

Chapter 8: Models for nominal outcomes focuses on the multinomial logit model. We show how to test hypotheses that involve multiple coefficients and discuss tests of a key assumption known as the independence of irrelevant alternatives assumption. Methods of interpretation using predictions are identical to those for ordinal models. Interpretation using odds ratios is a simple extension of the methods introduced in chapter 6, although the multinomial logit model's many parameters make the process of interpretation much more complicated. To deal with

this complexity, we present a graphical method for summarizing the parameters. The multinomial probit model without correlated errors is discussed briefly, and then the multinomial logit model is used to explain the stereotype logit model. This model, which is often used with ordinal outcomes, also has applications with nominal outcomes. These models assume case-specific independent variables (each independent variable has one value for each observation). We end the chapter with a short review of models that also include alternative-specific data, in which some variables vary over the alternatives for each individual, such as an individual's similarity to each candidate in an election. We consider the conditional logit model and the alternative-specific multinomial probit model, the latter of which allows correlations between alternative-specific error terms. Lastly, we present the rank-ordered logistic regression model, which can be used when you have information about the ranking of outcomes as opposed to information about only the selected or most preferred outcome.

Chapter 9: Models for count outcomes begins with the Poisson and negative binomial regression models, including a test to determine which model is appropriate for your data. We also show how to incorporate differences in exposure time into parameter estimation. Next, we consider interpretation for changes in the predicted rate and changes in the predicted probability of observing a given count. The rest of the chapter deals with models that address problems associated with having too many zeros relative to what the model predicts or having no zeros at all. We start with zero-truncated models for which zeros are missing from the outcome variable, perhaps because of the way the data were collected. We then merge a binary model and a zero-truncated model to create the hurdle model. We also consider fitting and interpreting zero-inflated count models, which are designed to account for the many zero counts often found in count outcomes.

1.5 The SPost software

From our point of view, one of the best things about Stata is how easy it is to add your own commands. If Stata does not have a command you need or some command does not work the way you like, you can program a new command yourself, and it will work as if it were part of official Stata. We have created a suite of programs, referred to collectively as SPost13 (Stata postestimation commands for version 13), for the postestimation interpretation of regression models. These commands must be installed before you can try the examples in later chapters.

If you have used SPost before, read this! For this book, we completely rewrote our earlier SPost commands, which we will refer to as SPost9. If you have the `spost9_ado` package installed on your computer, you should uninstall it (details below) before you install the `spost13_ado` package.

(Pages omitted)

9 Models for count outcomes

Count variables record how many times something has happened. Examples include the number of patients, hospitalizations, daily homicides, theater visits, international conflicts, beverages consumed, industrial injuries, soccer goals scored, new companies, and arrests by police. Although the linear regression model has often been applied to count outcomes, these estimates can be inconsistent or inefficient. In some cases, the linear regression model can provide reasonable results; however, it is much safer to use models specifically designed for count outcomes.

In this chapter, we consider seven regression models for count outcomes, all based on the Poisson distribution. We begin with the Poisson regression model (PRM), which is the foundation for other count models. We then consider the negative binomial regression model (NBRM), which adds unobserved, continuous heterogeneity to the PRM and often provides a much better fit to the data. To deal with outcomes where observations with zero counts are missing, we consider the zero-truncated Poisson and zero-truncated negative binomial models. By combining a zero-truncated model with a binary model, we develop the hurdle regression model, which models zero and nonzero counts in separate equations. Finally, we consider the zero-inflated Poisson and the zero-inflated negative binomial models, which assume that there are two sources of zero counts.

As with earlier chapters, we review the statistical models, consider issues of testing and fit, and then discuss methods of interpretation. These discussions are intended as a review for those who are familiar with the models. See Long (1997) for a more technical introduction to count models and Cameron and Trivedi (2013) for a definitive review. You can obtain sample do-files and datasets as explained in chapter 2.

9.1 The Poisson distribution

Because the univariate Poisson distribution is fundamental to understanding regression models for counts, we start by exploring this distribution. Let μ be the rate of occurrence or the expected number of times an event will occur during a given period of time. Let y be a random variable indicating the actual number of times an event did occur. Sometimes the event will occur fewer times than expected, even not at all, and other times it will occur more often.

The relationship between the expected count μ and the probability of observing a given count y is specified by the Poisson distribution

$$\Pr(y \mid \mu) = \frac{e^{-\mu}\mu^{y}}{y!} \qquad \text{for } y = 0, 1, 2, \ldots$$

where $\mu > 0$ is the sole parameter defining the distribution. $y!$ is the factorial operator; for example, $4! = 4 \times 3 \times 2 \times 1$. The easiest way to get a sense of the Poisson distribution is to compare plots of predicted probabilities for different values of the rate μ, as shown in figure 9.1.

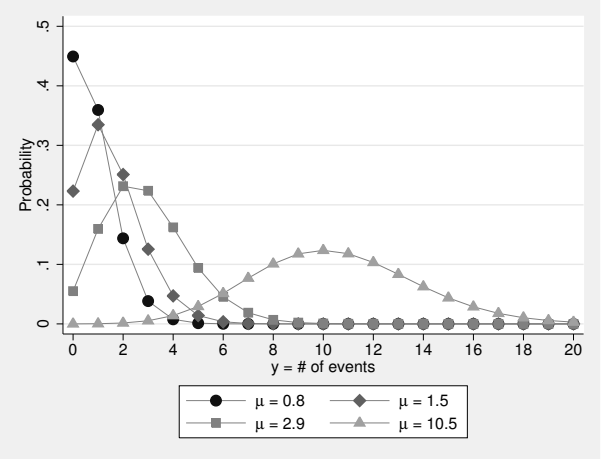


Figure 9.1. The Poisson probability density function (PDF) for different rates

The figure illustrates four characteristics of the Poisson distribution that are important for understanding regression models for counts:

1. As the mean of the distribution μ increases, the mass of the distribution shifts to the right.
2. The mean μ is also the variance. Thus $\text{Var}(y) = \mu$, which is known as equidispersion. In real data, count variables often have a variance greater than the mean, which is called overdispersion. It is possible for counts to be underdispersed, but this is rarer.
3. As μ increases, the probability of a zero count decreases rapidly. For many count variables, there are more observed 0s than predicted by the Poisson distribution.
4. As μ increases, the Poisson distribution approximates a normal distribution. This is shown by the distribution for $\mu = 10.5$.

These ideas are used as we develop regression models for count outcomes in the rest of the chapter.

Aside: Plotting the Poisson PDF. The commands below were used to create figure 9.1. The first `generate` creates variable `k`, which contains the values 0 to 20 that are the counts for which we want to compute probabilities. This is done by subtracting 1 from `_n`, where `_n` is how Stata refers to the row number of an observation. The probability of outcome k from a Poisson distribution with mean mu is computed with the function `poissonp(`mu`, `k`)` for each of four values of mu.[1]

```
clear all
set obs 21
gen k = _n - 1
label var k "y = # of events"
gen psn1 = poissonp(0.8, k)
label var psn1 "&mu = 0.8"
gen psn2 = poissonp(1.5, k)
label var psn2 "&mu = 1.5"
gen psn3 = poissonp(2.9, k)
label var psn3 "&mu = 2.9"
gen psn4 = poissonp(10.5, k)
label var psn4 "&mu = 10.5"
graph twoway connected psn1 psn2 psn3 psn4 k, ///
    ytitle("Probability") ylabel(0(.1).5) xlabel(0(2)20)
    lwidth(thin thin thin thin) msymbol(O D S T)
```

9.1.1 Fitting the Poisson distribution with the poisson command

To illustrate count models, we use data from Long (1990) on the number of articles written by biochemists in the 3 years prior to receiving their doctorate. The variables considered are

```
. use couart4, clear
(couart4.dta | Long data on Ph.D. biochemists | 2013-11-13)
. codebook art female married kid5 mentor phd, compact
```

Variable	Obs	Unique	Mean	Min	Max	Label
art	915	15	1.692896	0	19	Articles in last 3 yrs of PhD
female	915	2	.4601093	0	1	Gender: 1=female 0=male
married	915	2	.6622951	0	1	Married: 1=yes 0=no
kid5	915	4	.495082	0	3	# of kids < 6
mentor	915	49	8.767213	0	77	Mentor´s # of articles
phd	915	83	3.103109	.755	4.62	PhD prestige

1. We use the Stata Markup and Control Language code `&mu` to add the Greek letter μ to the labels.

The count outcome is the number of articles a scientist has published, with a distribution that is highly skewed with 30% of the cases being 0:

```
. tabulate art, missing
```

Articles in last 3 yrs of PhD	Freq.	Percent	Cum.
0	275	30.05	30.05
1	246	26.89	56.94
2	178	19.45	76.39
3	84	9.18	85.57
4	67	7.32	92.90
5	27	2.95	95.85
6	17	1.86	97.70
7	12	1.31	99.02
8	1	0.11	99.13
9	2	0.22	99.34
10	1	0.11	99.45
11	1	0.11	99.56
12	2	0.22	99.78
16	1	0.11	99.89
19	1	0.11	100.00
Total	915	100.00	

Often, the first step in analyzing a count variable is to compare the mean with the variance to determine whether there is overdispersion. By default, `summarize` does not report the variance, so we use the `detail` option:

```
. sum art, detail
```

Articles in last 3 yrs of PhD

Percentiles		Smallest		
1%	0	0		
5%	0	0		
10%	0	0	Obs	915
25%	0	0	Sum of Wgt.	915
50%	1		Mean	1.692896
		Largest	Std. Dev.	1.926069
75%	2	12		
90%	4	12	Variance	3.709742
95%	5	16	Skewness	2.51892
99%	7	19	Kurtosis	15.66293

The variance is more than twice as large as the mean, providing clear evidence of overdispersion.

9.1.2 Comparing observed and predicted counts with mgen

We can visually inspect the overdispersion in `art` by comparing the observed probabilities with those predicted from the Poisson distribution. Later, we will use the same method as a first assessment of the specification of count regression models (sec-

tion 9.2.5). We begin by using the `poisson` command to fit a model with a constant but no independent variables. When there are no independent variables, `poisson` fits a univariate Poisson distribution, where $\exp(\beta_0)$ equals the mean μ. Using `art` as the outcome,

```
. poisson art, nolog
Poisson regression                              Number of obs   =        915
                                                LR chi2(0)      =       0.00
                                                Prob > chi2     =          .
Log likelihood = -1742.5735                     Pseudo R2       =     0.0000

------------------------------------------------------------------------------
         art |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |   .5264408   .0254082    20.72   0.000     .4766416      .57624
------------------------------------------------------------------------------
```

Because $\widehat{\beta}_0 = 0.526$, the estimated rate is $\widehat{\mu} = \exp(0.526) = 1.693$, which matches the mean of `art` obtained with `summarize` earlier.

In earlier chapters, we used `mgen` to compute predictions as an independent variable changed, holding other variables constant. Although this can be done with count models, as illustrated below, the `mgen` option `meanpred` creates variables with observed and average predicted probabilities, where the rows correspond to values of the outcome.[2] The syntax for `mgen` used in this way is

`mgen, meanpred stub(`*stub*`) pr(`*min*`/`*max*`)` [*options*]

where option `pr(`*min*`/`*max*`)` specifies that we want to create variables with predictions for each of the counts from *min* to *max*. The new variables are

Variable name	Content
stub`val`	Value k of the dependent variable y ranging from *min* to *max*. The first row contains *min*; the second row *min*+1; etc.
stub`obeq`	Observed proportion or probability that $y = k$. These values correspond to the percentages from `tabulate`.
stub`oble`	Observed cumulative probability that $y \leq k$.
stub`preq`	Average predicted probability $\Pr(y = k)$.
stub`prle`	Average predicted probability $\Pr(y \leq k)$.
stub`obpr`	Difference between observed and predicted probabilities.

In a regression model with no independent variables, the predicted probability of $y = k$ is the same for all observations. Accordingly, `mgen` is simply computing $\Pr(y = k)$ from a Poisson distribution with a mean equal to the mean of the outcome. In our example,

2. SPost9 uses the `prcounts` command to do this.

```
. poisson art, nolog
  (output omitted)
. mgen, pr(0/9) meanpred stub(psn)
Predictions from:
```

Variable	Obs	Unique	Mean	Min	Max	Label
psnval	10	10	4.5	0	9	Articles in last 3 yrs...
psnobeq	10	10	.0993443	.0010929	.3005464	Observed proportion
psnoble	10	10	.8328962	.3005464	.9934427	Observed cum. proportion
psnpreq	10	10	.0999988	.0000579	.311469	Avg predicted Pr(y=#)
psnprle	10	10	.8307106	.1839859	.9999884	Avg predicted cum. Pr(...
psnob_pr	10	10	-.0006546	-.0691068	.1165605	Observed - Avg Pr(y=#)

`mgen` created six variables with 10 observations that correspond to the counts 0–9. In the list below, `psnval` contains the count values, `psnobeq` contains observed proportions or probabilities, and `psnpreq` has predicted probabilities from a Poisson distribution with mean 1.69:

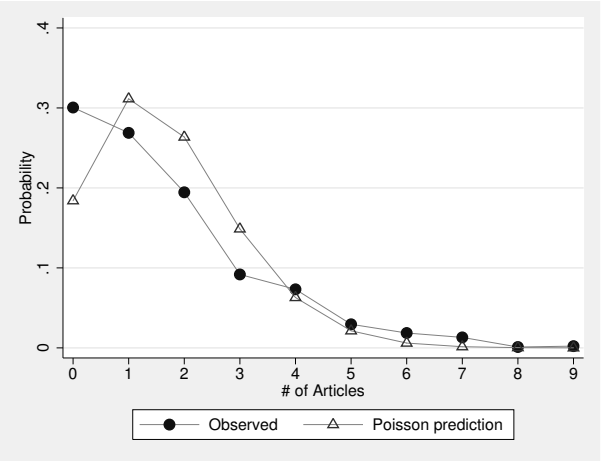
```
. list psnval psnobeq psnpreq in 1/10
```

	psnval	psnobeq	psnpreq
1.	0	.3005464	.1839859
2.	1	.2688525	.311469
3.	2	.1945355	.2636424
4.	3	.0918033	.148773
5.	4	.073224	.0629643
6.	5	.0295082	.0213184
7.	6	.0185792	.006015
8.	7	.0131148	.0014547
9.	8	.0010929	.0003078
10.	9	.0021858	.0000579

The values of `psnobeq` match those from `tabulate art` above, except that `tabulate` shows percentages while `mgen` generates probabilities, which equal the percentages divided by 100. The first row shows that the observed probability of no publications is 0.301, while the predicted probability from the Poisson distribution is only 0.184.

Using these variables, we can create a graph that compares the observed probabilities with the predicted probabilities from the Poisson distribution:

```
. label var psnobeq "Observed"
. label var psnpreq "Poisson prediction"
. label var psnval "# of Articles"
. graph twoway connected psnobeq psnpreq psnval,
>     ytitle("Probability") ylabel(0(.1).4, gmax) xlabel(0/9) msym(O Th)
```



The graph clearly shows that the fitted Poisson distribution underpredicts 0s; overpredicts counts 1, 2, and 3; and has smaller underpredictions of larger counts. This pattern of overprediction and underprediction is characteristic of count models that do not adequately account for heterogeneity among observations in their rate μ. Because the univariate Poisson distribution assumes that all scientists have exactly the same rate of productivity, which is clearly unrealistic, our next step is to allow heterogeneity in μ based on observed characteristics of the scientists.

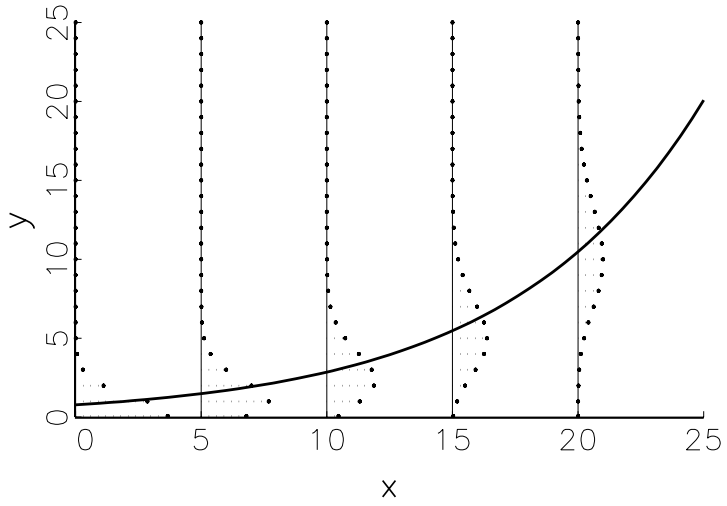
9.2 The Poisson regression model

The PRM extends the Poisson distribution by allowing each observation i to have a different rate μ_i. More formally, the PRM assumes that the observed count for observation i is drawn from a Poisson distribution with mean μ_i, where μ_i is estimated from the independent variables in the model. This is sometimes referred to as incorporating observed heterogeneity and leads to the structural equation

$$\mu_i = E\left(y_i \mid \mathbf{x}_i\right) = \exp\left(\mathbf{x}_i\boldsymbol{\beta}\right)$$

Taking the exponential of $\mathbf{x}_i\boldsymbol{\beta}$ forces μ_i to be positive, which is necessary because counts can be only 0 or positive.

To see how this works, consider the PRM with one independent variable:



The mean $\mu = \exp(\alpha + \beta x)$ is shown by the solid, curved line that increases as x increases. For each value of μ, the Poisson distribution around the mean is shown by the dots, which should be thought of as coming out of the page to represent the probability of each count. Interpretation of the model involves assessing how changes in the independent variables affect the conditional mean and the probabilities of each count. Details on interpretation are given after we consider estimation.

9.2.1 Estimation using poisson

The PRM is fit with the command

`poisson` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [`, noconstant`
`exposure(`*varname*`) vce(`*vcetype*`) irr`]

In our experience, `poisson` converges quickly and difficulties are rare.

Variable lists

depvar is the dependent variable. `poisson` does not require this to be an integer; however, if you have noninteger values, you obtain the following warning:

```
note: you are responsible for interpretation of noncount dep. variable.
```

indepvars is a list of independent variables. If *indepvars* is not included, a model with only an intercept is estimated that fits a univariate Poisson distribution, as shown in the previous section.

Specifying the estimation sample

if and in qualifiers. `if` and `in` qualifiers can be used to restrict the estimation sample. For example, if you want to fit a model for only women, you could specify `poisson art i.mar kid5 phd ment if female==1`.

Listwise deletion. Stata excludes observations with missing values for any of the variables in the model. Accordingly, if two models are estimated using the same data but have different independent variables, it is possible to have different samples. As discussed in chapter 3, we recommend that you explicitly remove observations with missing data.

Weights and complex samples

`poisson` can be used with `fweights`, `pweights`, and `iweights`. Survey estimation for complex samples is possible using `svy`. See chapter 3 for details.

Options

`noconstant` suppresses the constant term or intercept in the model.

`exposure(`*varname*`)` specifies a variable indicating the amount of time during which an observation was "at risk" of the event occurring. Details are given in section 9.2.6.

`vce(`*vcetype*`)` specifies the type of standard errors to be computed. `vce(robust)` requests that robust variance estimates be used. See sections 3.1.9 and 9.3.5 for details.

`irr` reports estimated coefficients that are transformed to incidence-rate ratios defined as $\exp(\beta_k)$. These are discussed in section 9.2.2.

Example of the PRM

If scientists who differ in their rates of productivity are combined, the univariate distribution of articles will be overdispersed, with the variance greater than the mean. Differences among scientists in their rates of productivity could be due to factors such as quality of their graduate program, gender, marital status, number of young children, and the number of articles written by a scientist's mentor. To account for these differences, we add these variables as independent variables:

```
. use couart4, clear
(couart4.dta | Long data on Ph.D. biochemists | 2014-04-24)
. poisson art i.female i.married kid5 phd mentor, nolog
Poisson regression                              Number of obs   =        915
                                                LR chi2(5)      =     183.03
                                                Prob > chi2     =     0.0000
Log likelihood = -1651.0563                     Pseudo R2       =     0.0525
```

art	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
female						
Female	-.2245942	.0546138	-4.11	0.000	-.3316352	-.1175532
married						
Married	.1552434	.0613747	2.53	0.011	.0349512	.2755356
kid5	-.1848827	.0401272	-4.61	0.000	-.2635305	-.1062349
phd	.0128226	.0263972	0.49	0.627	-.038915	.0645601
mentor	.0255427	.0020061	12.73	0.000	.0216109	.0294746
_cons	.3046168	.1029822	2.96	0.003	.1027755	.5064581

How you interpret a count model depends on whether you are interested in 1) the expected value or rate of the count outcome or 2) the distribution of counts. If your interest is in the rate of occurrence, several methods can be used to compute the change in the rate for a change in an independent variable, holding other variables constant. If your interest is in the distribution of counts or perhaps the probability of a specific count, such as not publishing, the probability of a count for a given level of the independent variables can be computed. We begin with interpretation using rates.

9.2.2 Factor and percentage changes in E(y | x)

In the PRM,

$$\mu = E(y \mid \mathbf{x}) = \exp(\mathbf{x}\beta)$$

The changes in μ as an independent variable changes can be presented in several ways. Factor change and percentage change coefficients are counterparts to odds ratios that were discussed in previous chapters. In those chapters, we expressed reservations about the usefulness of interpreting coefficients that indicated changes in the odds. As we will explain shortly, the interpretation of factor and percentage change coefficients in count models is much clearer and more useful.

Perhaps the most common method of interpretation is the factor change in the rate. Let $E(y \mid \mathbf{x}, x_k)$ be the expected count for a given $\mathbf{x}$, where we explicitly note the value of x_k, and let $E(y \mid \mathbf{x}, x_k + 1)$ be the expected count after increasing x_k by 1. Simple algebra shows that the ratio is

$$\frac{E(y \mid \mathbf{x}, x_k + 1)}{E(y \mid \mathbf{x}, x_k)} = e^{\beta_k} \tag{9.1}$$