Poisson Regression

James H. Steiger

Department of Psychology and Human Development
Vanderbilt University

Poisson Regression

Introduction

- In this lecture we discuss the Poisson regression model and some applications.
- Poisson regression deals with situations in which the dependent variable is a count.
- We'll start by quickly reviewing properties of the Poisson.

Introduction

The Poisson Distribution

- When events arrive without any systematic "clustering," i.e., they arrive with a known average rate in a fixed time period but each event arrives at a time independent of the time since the last event, the exact integer number of events can be modeled with the Poisson distribution
- The Poisson is a single parameter family, the parameter being λ, the expected number of events in the interval of interest
- For a Poisson random variable X, the probability of exactly r events is

$$Pr(X = r) = \frac{\lambda^r e^{-\lambda}}{r!}$$

Introduction

The Poisson Distribution

- The Poisson is used widely to model occurrences of low probability events.
- A random variable X having a Poisson distribution with parameter λ has mean and variance given by

$$E(X) = \lambda$$
$$Var(X) = \lambda$$

Introduction

The Poisson Distribution

- The Poisson distribution is a limiting case of the binomial distribution when the number of trials becomes large while the expectation remains stable, i.e., the probability of success is very small.
- An important additional property of the Poisson distribution is that sums of independent Poisson variates are themselves Poisson variates, i.e., if Y_1 and Y_2 are independent with Y_i having a $P(\mu_i)$ distribution, then

$$Y_1 + Y_2 \sim P(\mu_1 + \mu_2) \tag{1}$$

- As we shall see, the key implication of this result is that individual and grouped data can both be analyzed with the Poisson distribution.

An Introductory Example

On his superb website at *data.princeton.edu*(which I strongly recommend as a source for reading and examples), Germán Rodríguez presents an introductory example involving data from the World Fertility Study.

The Children Ever Born (ceb) Data

The dataset has 70 rows representing grouped individual data. Each row has entries for:

- The cell number (1 to 71, cell 68 has no observations)
- Marriage duration (1=0–4, 2=5–9, 3=10–14, 4=15–19, 5=20–24, 6=25–29)
- Residence (1=Suva, 2=Urban, 3=Rural)
- Education (1=none, 2=lower primary, 3=upper primary, 4=secondary+)
- Mean number of children ever born (e.g. 0.50)
- Variance of children ever born (e.g. 1.14)
- Number of women in the cell (e.g. 8)

Reference: Little, R. J. A. (1978). Generalized Linear Models for Cross-Classified Data from the WFS. *World Fertility Survey Technical Bulletins, Number 5*.

Introduction

The Poisson Distribution

- A tabular presentation shows data on the number of children ever born to married Indian women classified by:
  1. Duration since their first marriage (grouped in six categories)
  2. Type of place of residence (Suva, other urban and rural)
  3. Educational level (classified in four categories: none, lower primary, upper primary, and secondary or higher)
- Each cell in the table shows the mean, the variance and the number of observations.

Introduction

The Poisson Distribution

TABLE 4.1: Number of Children Ever Born to Women of Indian Race By Marital Duration, Type of Place of Residence and Educational Level (Each cell shows the mean, variance and sample size)

Marr.	Suva				Urban				Rural			
Dur.	N	LP	UP	S+	N	LP	UP	S+	N	LP	UP	S+
0–4	0.50	1.14	0.90	0.73	1.17	0.85	1.05	0.69	0.97	0.96	0.97	0.74
	1.14	0.73	0.67	0.48	1.06	1.59	0.73	0.54	0.88	0.81	0.80	0.59
	8	21	42	51	12	27	39	51	62	102	107	47
5–9	3.10	2.67	2.04	1.73	4.54	2.65	2.68	2.29	2.44	2.71	2.47	2.24
	1.66	0.99	1.87	0.68	3.44	1.51	0.97	0.81	1.93	1.36	1.30	1.19
	10	30	24	22	13	37	44	21	70	117	81	21
10–14	4.08	3.67	2.90	2.00	4.17	3.33	3.62	3.33	4.14	4.14	3.94	3.33
	1.72	2.31	1.57	1.82	2.97	2.99	1.96	1.52	3.52	3.31	3.28	2.50
	12	27	20	12	18	43	29	15	88	132	50	9
15–19	4.21	4.94	3.15	2.75	4.70	5.36	4.60	3.80	5.06	5.59	4.50	2.00
	2.03	1.46	0.81	0.92	7.40	2.97	3.83	0.70	4.91	3.23	3.29	–
	14	31	13	4	23	42	20	5	114	86	30	1
20–24	5.62	5.06	3.92	2.60	5.36	5.88	5.00	5.33	6.46	6.34	5.74	2.50
	4.15	4.64	4.08	4.30	7.19	4.44	4.33	0.33	8.20	5.72	5.20	0.50
	21	18	12	5	22	25	13	3	117	68	23	2
25–29	6.60	6.74	5.38	2.00	6.52	7.51	7.54	–	7.48	7.81	5.80	–
	12.40	11.66	4.27	–	11.45	10.53	12.60	–	11.34	7.57	7.07	–
	47	27	8	1	46	45	13	–	195	59	10	–

Introduction

The Poisson Distribution

- The unit of analysis is the individual woman.
- The response variable is the number of children given birth to, and the potential predictor variables are
  1. Duration since her first marriage
  2. Type of place where she resides
  3. Her educational level, classified in four categories.

The Poisson Regression Model

- The Poisson regression model assumes that the sample of n observations y_i are observations on independent Poisson variables Y_i with mean μ_i.
- Note that, if this model is correct, the equal variance assumption of classic linear regression is violated, since the Y_i have means equal to their variances.
- So we fit the generalized linear model,

$$\log(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta} \tag{2}$$

- We say that the Poisson regression model is a generalized linear model with Poisson error and a log link.

The Poisson Regression Model

- An alternative version of Equation 2 is

$$\mu_i = \exp(\mathbf{x}_i'\boldsymbol{\beta}) \tag{3}$$

- This implies that one unit increases in an x_j are associated with a *multiplication* of μ_j by $\exp(\beta_j)$.

The Poisson Regression Model

Grouped Data and the Offset

- Note that the model of Equation 2 refers to individual observations, but the table gives summary measures. Do we need the individual observations to proceed?
- No, because, as Germán Rodríguez explains very clearly in his lecture notes, we can apply the result of Equation 1.

The Poisson Regression Model

Grouped Data and the Offset

- Specifically, define Y_{ijkl} to be the number of children borne by the l-th woman in the (i, j, k)-th group, where i denotes marital duration, j residence and k education.
- Let $Y_{ijk\bullet} = \sum_l Y_{ijkl}$ be the group total shown in the table. Then if each of the observations in this group is a realization of an independent Poisson variate with mean μ_{ijk}, then the group total will be a realization of a Poisson variate with mean $n_{ijk}\mu_{ijk}$, where n_{ijk} is the number of observations in the (i, j, k)-th cell.

The Poisson Regression Model

Grouped Data and the Offset

- Suppose now that you postulate a log-linear model for the individual means, say

$$\log(\mu_{ijkl}) = \log E(Y_{ijkl}) = \mathbf{x}'_{ijk}\boldsymbol{\beta} \tag{4}$$

- Then the log of the expected value of the group total is

$$\begin{aligned} \log(E(Y_{ijk})) &= \log(n_{ijk}\mu_{ijk}) && (5) \\ &= \log(n_{ijk}) + \mathbf{x}'_{ijk}\boldsymbol{\beta} && (6) \end{aligned}$$

The Poisson Regression Model

Grouped Data and the Offset

- Thus, the group totals follow a log-linear model with exactly the same coefficients $\boldsymbol{\beta}$ as the individual means, except for the fact that the linear predictor includes the term $\log(n_{ijk})$.
- This term is referred to as the *offset*. Often, when the response is a count of events, the offset represents the log of some measure of exposure, in this case the number of women.

Testing Models of the Fertility Data

- Let's consider some models for predicting the fertility data from our potential predictors. Our first 4 models are:
  1. The null model, including only an intercept.
  2. A model predicting number of children from Duration (D).
  3. A model predicting number of children from Residence (R).
  4. A model predicting number of children from Education (E).
- To fit the models with Poisson regression, we use the `glm` package, specifying a `poisson` family (the log link is the default).

Testing Models of the Fertility Data

Simple One-Variable Models

- Here we fit simple models that predict number of children from duration, region of residence, and education. Let's begin by looking carefully at a model that predicts number of children solely from the duration of their childbearing years.]

```
> ceb.data <- read.table("ceb.dat",header=T)
> fit.D <- glm(y~dur,family="poisson",
+     offset=log(n),data=ceb.data)
> fit.E <- glm(y~educ,family="poisson",
+     offset=log(n),data=ceb.data)
> fit.R <- glm(y~res,family="poisson",
+     offset=log(n),data=ceb.data)
```

- Note that, in order to fit the model correctly, we had to specify `family ="poisson"` and `offset=log(n)`.

Testing Models of the Fertility Data

Predicting Children Ever Born from Duration

- The `dur` variable is categorical, so R automatically codes its 6 categories into 5 variables. Each of these variables takes on a value of 1 for its respective category.
- The first category, `00-04`, and has no variable representing it. Consequently, it is the "reference category" and has a score of zero.
- All the other categories are represented by dummy predictor variables that take on the value 1 if `dur` has that category—otherwise the dummy variable has a code of zero.

Testing Models of the Fertility Data

Predicting Children Ever Born from Duration

- Let's look at some output:

```
> summary(fit.D)

Call:
glm(formula = y ~ dur, family = "poisson", data = ceb.data, offset = log(n))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.5626  -1.4608  -0.5515   0.6060   4.0093

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.10413    0.04416  -2.358   0.0184 *
dur05-09     1.04556    0.05241  19.951   <2e-16 ***
dur10-14     1.44605    0.05025  28.779   <2e-16 ***
dur15-19     1.70719    0.04976  34.310   <2e-16 ***
dur20-24     1.87801    0.04966  37.818   <2e-16 ***
dur25-29     2.07923    0.04752  43.756   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3731.52  on 69  degrees of freedom
Residual deviance:  165.84  on 64  degrees of freedom
AIC: Inf

Number of Fisher Scoring iterations: 4
```

Testing Models of the Fertility Data

Predicting Children Ever Born from Duration

- Consider a woman whose first marriage was in the last 0–4 years. On average, such women have $\exp(-0.1) = 0.9$ children.
- Consider, on the other hand, a woman whose duration is 15–19 years. Such women have, on average $\exp(-0.1 + 1.71) = 4.97$ children.

Testing Models of the Fertility Data

Predicting Children Ever Born from Duration

- Next, let's look at education alone as a predictor.

```
> summary(fit.E)

Call:
glm(formula = y ~ educ, family = "poisson", data = ceb.data,
    offset = log(n))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-19.2952   -3.0804    0.7426    3.8574   13.1418

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.43567    0.01594  90.090   <2e-16 ***
educnone     0.21154    0.02168   9.759   <2e-16 ***
educsec+    -1.01234    0.05176 -19.557   <2e-16 ***
educupper   -0.40473    0.02951 -13.714   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3731.5  on 69  degrees of freedom
Residual deviance: 2661.0  on 66  degrees of freedom
AIC: Inf

Number of Fisher Scoring iterations: 5
```

Testing Models of the Fertility Data

Predicting Children Ever Born from Duration

- With 4 education categories, we need 3 dummy variables. Which category is the "reference" category in this case?
- Consider a woman whose education was "lower primary." Such women have, on average, $\exp(1.44) = 4.2$ children.
- Consider, on the other hand, a woman whose educational level is secondary+. Such women have, on average, $\exp(1.44 + -1.01) = 1.53$ children.

Testing Models of the Fertility Data

Re-ordering the Levels of a Factor

- In the preceding analysis, we discovered that R ordered the levels of the `educ` factor in a way that was somewhat suboptimal for presentation purposes.
- Since the levels of education form a natural order, we would like `educnone` to be the reference category, since it is a "natural" category for the intercept.
- There are a number of ways our intention can be communicated to R. For example, R has a `relevel` function you can use.

Testing Models of the Fertility Data

Re-ordering the Levels of a Factor

- However, perhaps the most natural approach, which also scores high on "code readability," is to recode the factor and recompute the fit object as shown below.

```
> ceb.data$educ <- factor(ceb.data$educ,levels = c("none","lower","upper","sec+"))
> fit.E <- glm(y~ educ,family="poisson", offset=log(n),data=ceb.data)
> summary(fit.E)
Call:
glm(formula = y ~ educ, family = "poisson", data = ceb.data,
    offset = log(n))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-19.2952   -3.0804    0.7426    3.8574   13.1418

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.64721    0.01469 112.104   <2e-16 ***
educlower   -0.21154    0.02168  -9.759   <2e-16 ***
educupper   -0.61627    0.02886 -21.353   <2e-16 ***
educsec+    -1.22388    0.05139 -23.814   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3731.5  on 69  degrees of freedom
Residual deviance: 2661.0  on 66  degrees of freedom
AIC: Inf

Number of Fisher Scoring iterations: 5
```

Testing Models of the Fertility Data

You Try It!

- Examine the model predicting number of children solely from place of residence. What is the reference category?
- What is the average number of children ever born for women in the reference category?

Testing Models of the Fertility Data

Two-Factor Additive Models

- Next we add education as a predictor to duration. The anova function helps us to see that there is a significant improvement.

```
> fit.NULL <-glm(y~1,family="poisson",
+   offset=log(n),data=ceb.data)
> fit.D.E <- glm(y~dur+educ,family="poisson",
+   offset=log(n),data=ceb.data)
> anova(fit.NULL,fit.D,fit.D.E)
Analysis of Deviance Table

Model 1: y ~ 1
Model 2: y ~ dur
Model 3: y ~ dur + educ
  Resid. Df Resid. Dev Df Deviance
1        69     3731.5
2        64      165.8  5   3565.7
3        61      100.0  3     65.8
```

Testing Models of the Fertility Data

Three-Factor Additive Model

- Next we add residence to duration and education.

```
> fit.D.E.R <- glm(y~dur+educ+res,
+  family="poisson",offset=log(n),data=ceb.data)
> anova(fit.NULL,fit.D,fit.D.E,fit.D.E.R)
Analysis of Deviance Table

Model 1: y ~ 1
Model 2: y ~ dur
Model 3: y ~ dur + educ
Model 4: y ~ dur + educ + res
  Resid. Df Resid. Dev Df Deviance
1        69     3731.5
2        64      165.8  5   3565.7
3        61      100.0  3     65.8
4        59       70.7  2     29.4
```

Testing Models of the Fertility Data

Three-Factor Additive Model

```
> summary(fit.D.E.R)

Call:
glm(formula = y ~ dur + educ + res, family = "poisson", data = ceb.data,
    offset = log(n))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2912  -0.6649   0.0759   0.6606   3.6790

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.03387    0.04870   0.696  0.48674
dur05-09     0.99765    0.05275  18.912  < 2e-16 ***
dur10-14     1.37053    0.05108  26.833  < 2e-16 ***
dur15-19     1.61423    0.05121  31.524  < 2e-16 ***
dur20-24     1.78549    0.05122  34.856  < 2e-16 ***
dur25-29     1.97679    0.05005  39.500  < 2e-16 ***
educlower    0.02308    0.02266   1.019  0.30832
educupper   -0.10167    0.03099  -3.281  0.00104 **
educsec+    -0.30958    0.05519  -5.609 2.03e-08 ***
resSuva     -0.15122    0.02833  -5.338 9.37e-08 ***
resurban    -0.03896    0.02462  -1.582  0.11363
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3731.525  on 69  degrees of freedom
Residual deviance:   70.653  on 59  degrees of freedom
AIC: Inf

Number of Fisher Scoring iterations: 4
```

Testing Models of the Fertility Data

Three-Factor Additive Model

- What is the predicted average number of children for women married 5–9 years, living in Suva, with post-secondary education?

Modeling Overdispersion

- As we mentioned at the outset, Poisson distributed variables have variances equal to their means.
- Consequently, if we observe a set of observations x_i that truly are realizations of a Poisson random variable X, these observations should show a sample variance that is reasonably close to their sample mean.
- In a similar vein, if we observe a set of sample proportions $\hat{p}_i$, each based on N_i independent observations, and our model is that they all represent samples in a situation where p remains stable, then the variation of the $\hat{p}_i$ should be consistent with the formula $p(1-p)/N_i$.

Modeling Overdispersion

Observing Overdispersion in Practice

- There are numerous reasons why overdispersion can occur in practice. Let's consider sample proportions based on the binomial.
- Suppose we hypothesize that the support enjoyed by President Obama is constant across 5 midwestern states. That is, the proportion of people in the populations of those states who would answer "Yes" to a particular question is constant.
- We perform opinion polls by randomly sampling 200 people in each of the 5 states.

Modeling Overdispersion

Observing Overdispersion in Practice

- We observe the following results: Wisconsin 0.285, Michigan 0.565, Illinois 0.280, Iowa 0.605, Minnesota .765.
- An unbiased estimate of the average proportion in these states can be obtained by simply averaging the 5 proportions, since each was based on a sample of size $N = 200$.
- Using R, we obtain:

```
> data <- c(0.285,0.565,0.280,0.605,.765)
> mean(data)
[1] 0.5
```

Modeling Overdispersion

Observing Overdispersion in Practice

- These proportions have a mean of 0.50. They also show considerable variability.
- Is the variability of these proportions consistent with our binomial model, which states that they are all representative of a constant proportion p?
- There are several ways we might approach this question, some involving brute force statistical simulation, others involving the use of statistical theory. Recall that sample proportions based on $N = 200$ *independent* observations should show a variance of $p(1 - p)/N$.
- We can estimate this quantity in this case as

```
> 0.50*(1-0.50)/200
[1] 0.00125
```

Modeling Overdispersion

Observing Overdispersion in Practice

- On the other hand, these 5 sample proportions show a variance of

```
> var(data)
[1] 0.045025
```

- The variance ratio is

```
> variance.ratio = var(data) / (0.50*(1-0.50)/200)
> variance.ratio
[1] 36.02
```

- The variance of the proportions is 36.02 times as large as it should be. There are several statistical tests we could perform to assess whether this variance ratio is statistically significant, and they all reject the null hypothesis that the actual variance ratio is 1.

Modeling Overdispersion

Observing Overdispersion in Practice

- As an example, we could look at the residuals of the 5 sample proportions from their fitted value of .50. The residuals are:

```
> residuals <- data - mean(data)
> residuals
[1] -0.215  0.065 -0.220  0.105  0.265
```

- Each residual can be converted to a standardized residual z-score by dividing by its estimated standard deviation.

```
> standardized.residuals <- residuals / sqrt(0.50*(1-0.50)/200)
```

- We can then generate a χ^2 statistic by taking the sum of squared residuals. The statistic has the value

```
> chi.square <- sum(standardized.residuals^2)
> chi.square
[1] 144.08
```

Modeling Overdispersion

Observing Overdispersion in Practice

- We have to subtract one degree of freedom because we estimated p from the mean of the proportions. Our χ^2 statistic can be compared to the χ^2 distribution with 4 degrees of freedom. The 2-sided $p - value$ is

```
> 2*(1-pchisq(chi.square,4))
[1] 0
```

Modeling Overdispersion

Observing Overdispersion in Practice

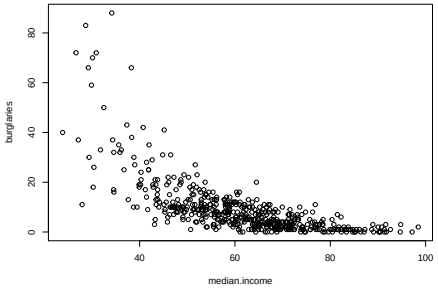
- Our sample proportions show overdispersion. Why?
- The simplest explanation in this case is that they are not samples from a population with a constant proportion p. That is, there is heterogeneity of support for Obama across these 5 states.
- Can you think of another reason why a set of proportions might show overdispersion? (C.P.)
- How about underdispersion? (C.P.)

Modeling Overdispersion

Observing Overdispersion in Practice

- Since counts are free to vary over the integers, they obviously can show a variance that is either substantially greater or less than their mean, and thereby show overdispersion or underdispersion relative to what is specified by the Poisson model.
- As an example, suppose we examine the impact of the median income (in thousands) of families in a neighborhood on the number of burglaries per month. Load the *burglary.txt* data file, then plot `burglaries` as a function of `median.income`. These data represent burglary counts for 500 metropolitan and suburban neighborhoods.

```
> plot(median.income,burglaries)
```



Modeling Overdispersion

Observing Overdispersion in Practice

- Let's examine some data for evidence of overdispersion. First, we'll grab scores corresponding to a `median.income` between 59 and 61.

```
> test.data <- burglaries[median.income > 59 & median.income < 61]
> var(test.data)
[1] 22.53846
> mean(test.data)
[1] 7.333333
> var(test.data) / mean(test.data)
[1] 3.073427
```

- The variance for these data is more than 3 times as large as the mean.

Modeling Overdispersion

Observing Overdispersion in Practice

- Let's try another region of the plot.

```
> test.data <- burglaries[median.income > 39 & median.income < 41]
> var(test.data)
[1] 97.14286
> mean(test.data)
[1] 21.85714
> var(test.data) / mean(test.data)
[1] 4.444444
```

Modeling Overdispersion

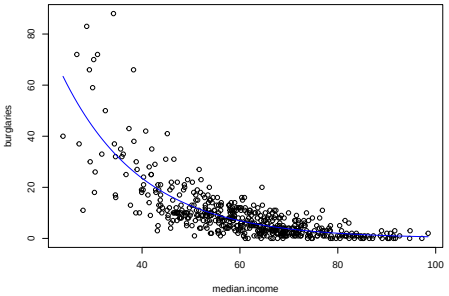
Observing Overdispersion in Practice

- The data show clear evidence of overdispersion. Let's fit a standard Poisson model to the data.

```
> standard.fit <- glm(burglaries ~ median.income, family = "poisson")
> summary(standard.fit)

Call:
glm(formula = burglaries ~ median.income, family = "poisson")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.6106  -1.2794  -0.2884   0.9102   7.7649

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    5.612422   0.055996  100.23   <2e-16 ***
median.income -0.061316   0.001091  -56.19   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 4721.4  on 499  degrees of freedom
Residual deviance: 1452.6  on 498  degrees of freedom
AIC: 3196.4

Number of Fisher Scoring iterations: 5
```

Fitting the Overdispersed Poisson Model

```
> plot(median.income,burglaries)
> curve(exp(coef(standard.fit)[1] + coef(standard.fit)[2]*x),add=TRUE,col="blue")
```



- The expected mean line, plotted with the coefficients from the model, looks like a nice fit to the data.
- However, the variance is several times the mean in this model, and since the standard errors are based on the assumption that the variance is equal to the mean, this creates a problem. The actual variance is several times what it should be, and so the standard errors printed by the program are underestimates.

Fitting the Overdispersed Poisson Model

- There are two fairly standard ways of handling this in R.
- One way assumes simply that the conditional distribution is like the Poisson, but with the variance a constant multiple of the mean rather than being equal to the mean.
- This approach is used in `glm` by selecting `family="quasipoisson"`. Notice how the dispersion parameter is estimated, and the estimated standard errors from the Poisson fit are divided by the square root of this parameter to obtain the revised standard errors shown on the next slide.

Fitting the Overdispersed Poisson Model

```
> overdispersed.fit <- glm(burglaries ~ median.income,family="quasipoisson")
> summary(overdispersed.fit)

Call:
glm(formula = burglaries ~ median.income, family = "quasipoisson")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.6106  -1.2794  -0.2884   0.9102   7.7649

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.612422   0.096108   58.40   <2e-16 ***
median.income -0.061316   0.001873  -32.74   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 2.945783)

    Null deviance: 4721.4  on 499  degrees of freedom
Residual deviance: 1452.6  on 498  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 5
```

Fitting the Overdispersed Poisson Model

- Another more sophisticated approach uses quasi-likelihood estimation to fit the negative binomial model, which assumes that the log means predicted from `median.income` are perturbed by random variation (having a gamma distribution).
- This random variation means that individual observations, for a given value of the predictors, can have different means, centered around $\mathbf{x}'\boldsymbol{\beta}$.
- This leaves the conditional mean line the same, but inflates the variance relative to that predicted by the Poisson.
- The variance inflation is not constant, however. In the negative binomial, there is an overdispersion parameter θ, but the variance and mean are related as follows:

$$\sigma^2 = \mu(1 + \mu/\theta) \tag{7}$$

Fitting the Overdispersed Poisson Model

- We can fit the negative binomial model, using the MASS library function `glm.nb`. (Make sure the MASS library is loaded.)

```
> negative.binomial.fit <- glm.nb(burglaries ~ median.income)
> summary(negative.binomial.fit)
Call:
glm.nb(formula = burglaries ~ median.income, init.theta = 4.956789611,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.8813  -0.8490  -0.1922   0.6297   2.9637

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    5.57414    0.12042   46.29   <2e-16 ***
median.income -0.06060    0.00207  -29.27   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(4.9568) family taken to be 1)

    Null deviance: 1606.97  on 499  degrees of freedom
Residual deviance:  545.33  on 498  degrees of freedom
AIC: 2730.7

Number of Fisher Scoring iterations: 1


              Theta:  4.957
          Std. Err.:  0.550

 2 x log-likelihood:  -2724.713
```

Fitting the Overdispersed Poisson Model

- In this case, the data were artificial. I created them according to the negative binomial model $\mu = -.06x + 5.5$, with overdispersion parameter $\theta = 5$.
- As you can see, in this case `glm.nb` estimates were very close to the true values, and the χ^2 fit statistic of 545.33 fails to reach significance at the .05 level, meaning that the hypothesis of perfect fit cannot be rejected.
- On the other hand, the `quasipoisson` family model fit, which assumes that the variance is a constant multiple of the mean, could not fit these data nearly as well. The deviance statistic of 1452.6 is much higher.

Fitting the Overdispersed Poisson Model

- Consider an instructive case, when median.income is 30. In this case, the mean and variance are actually

```
> m <- exp(-.06 * 30 + 5.5)
> v <- m * (1+m/5)
> m
[1] 40.4473
> v
[1] 367.6442
```

- The quasipoisson fit estimates them as

```
> m <- exp(coef(overdispersed.fit)[1] + coef(overdispersed.fit)[2] * 30)
> v <- m * 2.945783
> m
(Intercept)
   43.50732
> v
(Intercept)
   128.1631
```