Count Regression

Paul E. Johnson[1]

[1] Center for Research Methods and Data Analysis

2018

CENTER FOR RESEARCH METHODS & DATA ANALYSIS

College of Liberal Arts & Sciences

Outline

Poisson entry point to GLM

- Logistic and Poisson regression are the two most common entry point to generalized linear models
- The Poisson model is a discrete probability distribution.
  - build a model that predicts the expected value of a Poisson distribution
  - odd implication: expected value same as variance
- Rapidly advancing software and theory: this was not feasible when I was in graduate school, very significant advances 1988-2000.

The Standard Playbook

- Try the basic Poisson GLM (estimated by maximum likelihood)
- Check diagnostics
- If diagnostics indicate trouble, explore follow-up models that might be more likely to have generated the data

What comes after Poisson?

- Negative Binomial model: same expected value, higher variance
- Quasi-likelihood: same expected value, higher variance
- "Rate" models (response variable becomes $y_i/exposure_i$)
- Zero-Inflated mixture models
  - "Hurdle model": No count is observed for cases below a threshold
  - Can be separate formula predicting "0 or not", and then a Poisson or NB model for the observed counts (jargon "two-part model")

log and exp

- My logistic regression notes (LogitProbit-1) have graphs of logarithmic (log) and exponential (exp) functions
- Important to remember natural log and exponential are inverses of each other, so if

$$y = e^x$$

apply natural log on both sides

$$\begin{aligned} ln(y) &= ln(e^x) \\ &= x \end{aligned}$$

That's by definition of natural log:
$log_e(y) = x$, means x is the power to which we must raise e to equal y , so

$$y = e^x$$

log and exp ...

- The general idea of an inverse function: one that "undoes" the other. If a function is

$$g(x)$$

  - and if g^{-1} is an inverse, then

$$g^{-1}(g(x)) = x$$

- Notation: the superscript is inspired by idea that

$$\frac{1}{x} \times x = 1$$

$$x^{-1}x = 1$$

$$x = x$$

exponential detail

- Remember this fact?

$$2^{3+5} = 2^3 \times 2^5 \quad (1)$$

- That's true for all exponential formulas, so

$$e^{\beta_0+\beta_1 X1_i+\beta_2 X2_i} = e^{\beta_0} \times e^{\beta_1 X1_i} \times e^{\beta_2 X2_i} \quad (2)$$

- And if we are saving vertical space, write that as

$$exp(\beta_0+\beta_1 X1_i+\beta_2 X2_i) = exp(\beta_0) \times exp(\beta_1 X1_i) \times exp(\beta_2 X2_i) \quad (3)$$

- What's the big point. By using the transformation exp, we create interactions "automatically".

exponential detail 2

- Later on, I'm going to need to claim

$$e^{X_i\beta} \times \delta_i$$

can be re-written as:

$$e^{X_i\beta + log(\delta_i)}$$

- Why? Put the previous exponential facts together

$$\begin{aligned} & e^{X_i\beta} \times \delta_i \\ = \; & e^{X_i\beta} \times exp(log(\delta_i)) \\ = \; & e^{X_i\beta + log(\delta_i)} \end{aligned}$$

- So, multiplying $e^{X_i\beta}$ by a number δ_i is identical to adding $log(\delta_i)$ in the exponent.

Counts

Definition: Observed outcomes are integers 0, 1, 2, ...

Expected Value: a positive floating point number, the "mean" of the "data generating process" (AKA population).

- Think of the data generating process (DGP) like this
  1. $\eta_i = X_i\beta$ (your predictors and coefficients)
  2. $transform(\eta_i)$ to get an expected value of the outcome
  3. draw a random value from the probability model using input from step 2.

Generalized Linear Model Approach

(McCullagh & Nelder, 1989)

Linear Predictor $\eta_i = X_i\beta = \beta_0 + \beta_1 X1_i + \beta_2 X2_i \ldots$.

- LP is a "bottleneck", or "funnel": Effects of all predictors and coefficients "go through" the LP.

Expected value Our regression models ("predicts"?) the expected value of y_i, commonly referred to as μ_i
I think of this as a transformation of η_i into the expected value

$$E[y|X_i] = \mu_i = transform(\eta_i) \qquad (4)$$

We require the $transform$ to be an "increasing" function: as η_i goes up, then μ_i must go up.

Examples of transformations that work

Keep in mind $\eta_i = X_i\beta$, the linear predictor

Examples of transformations

- $\mu_i = X_i\beta = \eta_i$ (linear!)
- $\mu_i = exp(X_i\beta)$ (exponential)
- $\mu_i = \frac{1}{1+e^{-X_i\beta}} = \frac{exp(X_i\beta)}{1+exp(X_i\beta)}$ (logistic)
- $\mu_i = \Phi(X_i\beta)$ (probit, Φ is CDF of Normal distribution)

IMHO, we are asked to think backwards

When McCullagh & Nelder invented this, they thought of the transformation on the other side of the equation

$$g(\mu_i) = \eta_i \tag{5}$$

They called $g(\mu_i)$ the *Link Function* converts (expected value of y_i back onto η_i)

Example: In my way of thinking, the substantively meaningful transformation is exponential

$$\mu_i = exp(\eta_i)$$

But in their way of thinking, it is a “log link”:

$$log(\mu_i) = \eta_i$$

Mathematically, they are same.

But the “link” function always seemed backwards to my brain!

IMHO, we are asked to think backwards ...

Inverse Link Function

- We can "swap" between my way and their way because of assumption: $g(u_i)$ is increasing (monotonic)
- These are equivalent

$$g(\mu_i) = \eta_i \, , \text{and}$$

$$\mu_i = g^{-1}(\eta_i)$$

- "inverse link" function g^{-1}
  - However, carrying around notation g^{-1} is tedious, so we give that transform some other letter, say h

$h()$: The thing I called $transform$ above

$$\mu_i = h(\eta_i) = h(X_i\beta)$$

IMHO, we are asked to think backwards ...

- McCullagh & Nelder provided an estimation algorithm that works for a wide variety of models (probability distributions from the exponential family, which includes the Normal, Binomial, Poisson, Gamma, and others)

Normal

- OLS from a GLM point-of-view.
- The outcome is Normal, but the μ_i parameter is different for every case:

$$y_i \sim N(\eta_i, \sigma_\varepsilon^2)$$

- $\eta_i = \mathbf{X}_i\beta$ dials the Normal distribution's expected value up and down.
- Example of the "Identity" link
- σ_ε^2: A variance parameter same for all i.
- Flaws: The Normal distribution will not serve well with count data
  - Count data: we require discrete outcomes 0, 1, 2, ...
  - η_i may be negative

More on Poisson and Neg Binomial

My “distribution writeups” folder has more detailed descriptions and illustrations

- Poisson http://pj.freefaculty.org/guides/stat/Distributions/DistributionWriteups/Poisson/Poisson_v1-2.pdf
- Negative Binomial http://pj.freefaculty.org/guides/stat/Distributions/DistributionWriteups/NegativeBinomial/NegativeBinomial.pdf

Building a set for instructional purposes

- Will have data ready for use in R and Stata, with some example code.
- gavote: the "undercount" (lost ballot) problem in Georgia (from R package faraway)
- workdays: days missed at work due to mental health concern (from R package smdata)
- goals: soccer goals scored by players in the European league
- COUNT: extracted from the R package COUNT

Ex: Georgia vote data

The outcome that I model is "wasted ballots" in Georgia counties as a function of the urban/rural classification of the county, the percent African American, and the type of voting equipment that is used.

```
library(rockchalk)
library(faraway)
gavote$undercount <- gavote$ballots - gavote$votes
options.orig <- options()
options(width=65)
summarize(gavote)
```

Ex: Georgia vote data ...

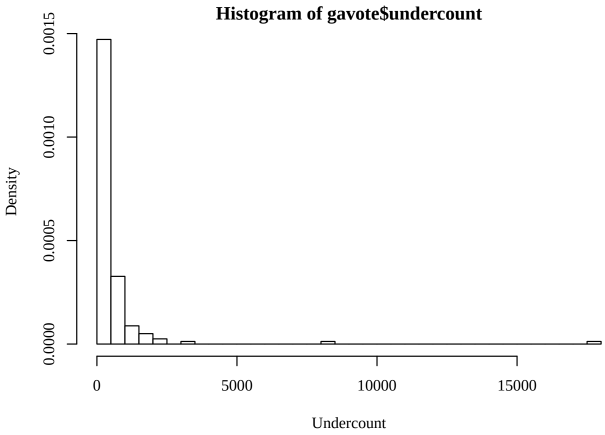
```
Numeric variables
             perAA         gore        bush        other
min          0             249         271         5
med          0.233         2326        3597        86
max          0.765         154509      140494      7920
mean         0.243         7020.314    8929.057    381.654
sd           0.163         19317.780   18029.960   1150.975
skewness     0.477         6.111       5.141       5.399
kurtosis     -0.325        40.686      30.035      29.803
nobs         159           159         159         159
nmissing     0             0           0           0
             votes         ballots     undercount
min          832           881         0
med          6299          6712        296
max          263211        280975      17764
mean         16331.025     16926.503   595.478
sd           36623.274     37865.152   1584.281
skewness     5.081         5.129       8.798
kurtosis     27.360        28.094      87.478
nobs         159           159         159
nmissing     0             0           0

Nonnumeric variables
                equip                 econ
 LEVER: 74               middle: 69
```

Ex: Georgia vote data ...

```
OS-CC: 44                 poor   : 72
OS-PC: 22                 rich   : 18
PAPER: 2
PUNCH: 17
nobs          : 159.000 nobs          : 159.000
nmiss         :   0.000 nmiss         :   0.000
entropy       :   1.846 entropy       :   1.396
normedEntropy:   0.795 normedEntropy:   0.881
                rural                   atlanta
rural: 117                Atlanta    : 15
urban: 42                 notAtlanta: 144



nobs          : 159.000 nobs          : 159.000
nmiss         :   0.000 nmiss         :   0.000
entropy       :   0.833 entropy       :   0.451
normedEntropy:   0.833 normedEntropy:   0.451
```

```
options(options.orig)
```

Ex: Georgia vote data



Outline

Poisson Distribution

- Poisson has one parameter, commonly called λ_i.
- The probability mass function (PMF: probability of value y given parameter λ) is

$$Pr(y|\lambda) = \frac{exp(-\lambda)\lambda^y}{y!}, \text{or}$$

$$\frac{e^{-\lambda}\lambda^y}{y!}$$

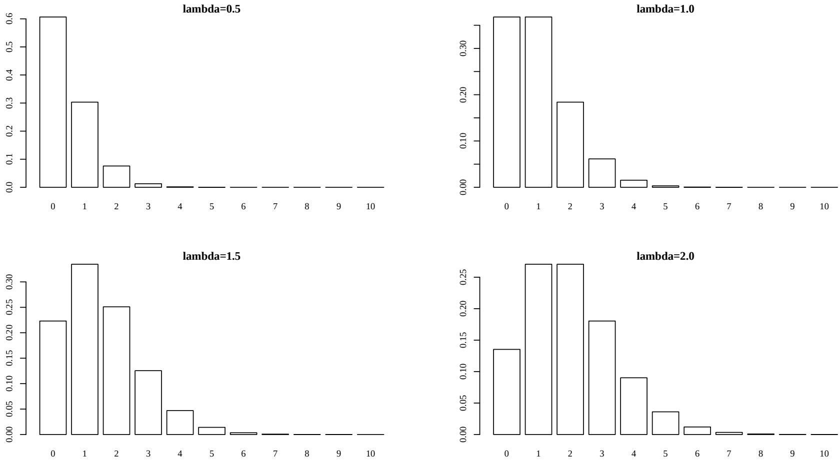
- Does not make sense if $\lambda < 0$. I mean, we require $\lambda \geq 0$.
- We'll need subscript i
    - i to remember this value may differ among observations, $i \in \{1, ..., N\}$
- In regression, we are going to say that each row of y is drawn from a different PMF. λ_i is different for each:

$$Pr(y|\lambda_i) = \frac{exp(-\lambda_i)\lambda_i^{y_i}}{y!}$$

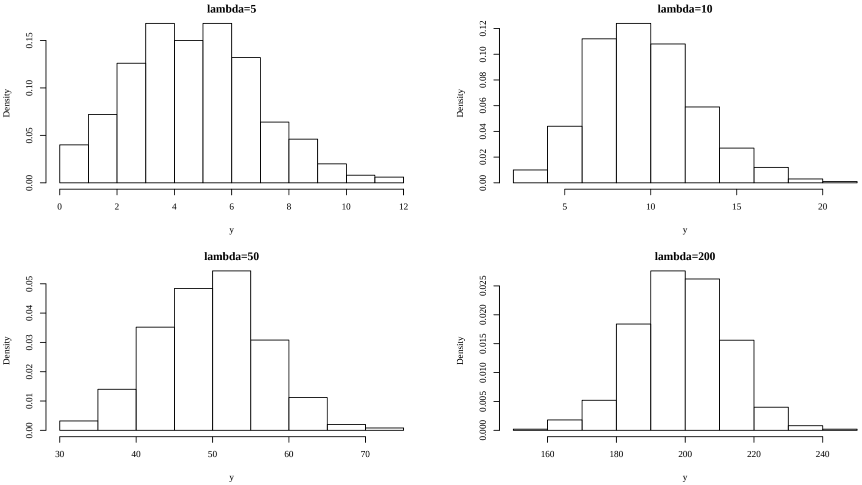
Poisson Distribution ...

- Using this, we can say how likely any value of y might be.
- In particular, for likelihood analysis, we can say how likely a particular observed value y_i was for a given λ_i.

Visualize the Poisson Probability Mass Function



Poisson Sample, large mu



Special Poisson Properties

- The Expected Value of the Poisson PMF is equal to its Variance
- And both are equal to the value of the parameter λ_i
  - $E[y_i] = \lambda_{ii}$
  - $Var[y_i] = \lambda_i$
- Beware of my errors: Sometimes, my notes have μ_i in place of λ_i because the expected value is commonly referred to as μ_i in GLM literature

Need to Be Careful Modeling λ_i

- The wrong link function might lead to disaster:

$$\lambda_i = \mathbf{X}_i\boldsymbol{\beta} \tag{6}$$

  - If $\mathbf{X}_i\boldsymbol{\beta} < 0$, the PMF is undefined!

- A transformation h is needed!

$$h(\mathbf{X}_i\boldsymbol{\beta}) \tag{7}$$

- The most common transformation is the exponential function.

$$\lambda_i = exp(\eta_i) = exp(\mathbf{X}_i\boldsymbol{\beta})$$

- Implies the log link function, $log(\mu_i) = \eta_i$

Noteworthy

1. Theoretically: EV $=$ Var $= \lambda_i$.
   - That's simple, but is does observed data have this property?
2. The shape changes.
   - When λ_i is small, it has lots of 0's, is very asymmetric
   - As λ_i grows, approaches a unimodal shape.
3. Because large $\lambda_i \rightarrow$ a more "Normal" appearance, some argue in favor of using a GLM with a Normally distributed outcome in that case.

Simulated X and Y

$$X1_i \sim N(20,5)$$
$$X2_i \sim N(10,3),$$

$$\eta_i = 0.1 + 0.75X1_i - 0.15X2_i \tag{8}$$

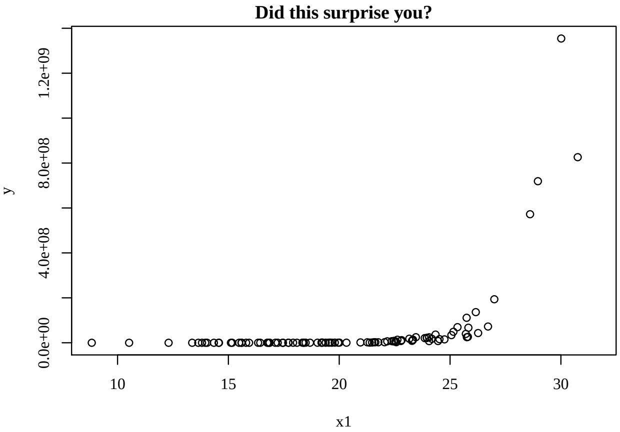
Draw observed scores

$$y|\eta \sim Pois(exp(\eta_i)) \tag{9}$$

You'll see the fit is very accurate for slope coefficients

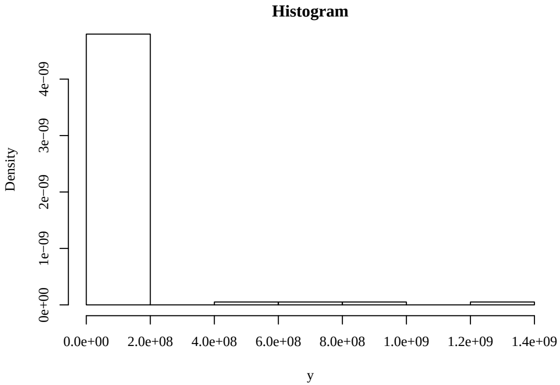
Residual deviance is VERY "touchy" with small changes in range of predictors or coefficients

Does This Surprise You?



Thinking something went wrong? I was.

The Histogram of y



- Discouraged? Don't worry.
- Estimates next page are almost ridiculously accurate!

Poisson GLM Fit

```
glm1 <- glm(y ~ x1 + x2, data=dat,
    family=poisson(link=log))
summary(glm1)
```

```
Call:
glm(formula = y ~ x1 + x2, family = poisson(link = log), data = dat)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1831  -0.6880   0.1172   0.7351   2.5707

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) 9.978e-02  2.032e-04     491   <2e-16 ***
x1          7.500e-01  7.970e-06   94100   <2e-16 ***
x2         -1.500e-01  5.053e-06  -29686   <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: 2.0717e+10  on 98  degrees of freedom
Residual deviance: 1.0285e+02  on 96  degrees of freedom
```

Poisson GLM Fit ...

```
   (1 observation deleted due to missingness)
AIC: 1664.3

Number of Fisher Scoring iterations: 3
```

“Residual deviance” is in ballpark of df (discuss overdispersion below)

Just for Curiosity, fit OLS

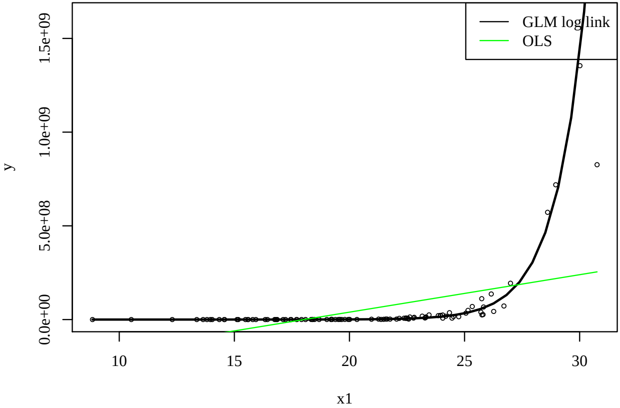
```
lm1 <- lm(y ~ x1 + x2, data=dat)
summary(lm1)
```

```
Call:
lm(formula = y ~ x1 + x2, data = dat)

Residuals:
      Min          1Q     Median         3Q        Max
-133325385  -81548444  -29144773   26306378 1114657874

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -366588914   85770267  -4.274 4.53e-05 ***
x1            19986500    3681228   5.429 4.25e-07 ***
x2              621443    4685999   0.133    0.895
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 160200000 on 96 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.2384,  Adjusted R-squared:  0.2225
F-statistic: 15.03 on 2 and 96 DF,  p-value: 2.102e-06
```

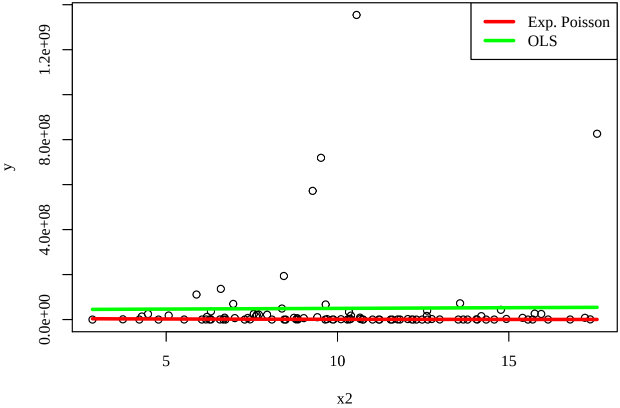
Manipulate plotCurves to get overlay

```
library(rockchalk)
## harvest lm information, ignore drawing
lm1ps <- plotSlopes(lm1, plotx = "x1")
plotCurves(glm1, plotx = "x1", plotLegend = FALSE)
lines(fit ~ x1, data = lm1ps$newdata, col =
    "green")
legend("topright", legend = c("GLM log link",
    "OLS"), col = c("black", "green"), lty =
    c(1,1))
```

plotCurves output



From x2's point of view



Outline

Maximum Likelihood Analysis

- Replace λ_i in PMF with $exp(\mathbf{X}_i\boldsymbol{\beta})$

$$Pr(y_i|\mathbf{X}_i,\boldsymbol{\beta}) = \frac{exp(-exp(\mathbf{X}_i\boldsymbol{\beta}))(exp(\mathbf{X}_i\boldsymbol{\beta}))^{y_i}}{y_i!}$$

or it looks slightly less ugly if we write:

$$Pr(y_i|\mathbf{X}_i\boldsymbol{\beta}) = \frac{exp(-e^{\mathbf{X}_i\boldsymbol{\beta}})(e^{\mathbf{X}_i\boldsymbol{\beta}})^{y_i}}{y_i!}$$

Estimation: ML

- Choose β to maximize the Likelihood function:

$$L(\beta; y, X) = Pr(y_1|X_i\beta) * Pr(y_2|X_i\beta) * ... * Pr(y_N|X_i\beta)$$

- The log likelihood

$$lnL = \sum_{i=1}^{N} ln(Pr(y_i|\mathbf{X}_i\boldsymbol{\beta})) \quad (10)$$

- McCullagh & Nelder (1989) demonstrated a method of iterative calculation (IRLS) that will find the maximum for all members of the exponential family of distributions.

Outline

GLM model diagnostics

- In 2006, I wrote up the most gigantic GLM notes ever.... http://pj.freefaculty.org/guides/stat/Regression-GLM/GLM1-Overview/GLM1-guide.pdf http://pj.freefaculty.org/guides/stat/Regression-GLM/GLM2-SigTests/GLM-2-guide.pdf
- A primary worry about Poisson regression is "overdispersion".

  Overdispersion: observed variance of y is greater than anticipated by the Poisson theory

GLM model diagnostics

- Poisson regression output includes "residual deviance". A "rule of thumb" is that if the residual deviance is much greater than the degrees of freedom, then the mismatch between the model and the data is unacceptably large
- Definition: "deviance": the difference in the quality of fit between
  1. a fitted model (your model) and
  2. the **saturated model**
     - the best fitting model that could be obtained by making a prediction μ_i for each unique group of predictor values.
     - The saturated model is a very very good model, fits the data as well as possible.

  - The research question: is "your model" unacceptably far from the saturated model.

GLM model diagnostics ...

- The comparison of the likelihoods is written like this:

$$scaled\,model\,deviance = -2ln\left[\frac{L(fitted)}{L(saturated)}\right]$$

$$scaled\,model\,deviance = 2lnL(saturated\,model) - 2lnL(fitted\,model)$$

- similar to a likelihood ratio test (makes you expect a χ^2 test)
- If y is Normal, then as the sample size grows, this test statistic is increasingly similar to a Chi-squared random variable.
- For other distributions of y, or for small samples, the comparison of the scaled model deviance against a $Chi - squared$ distribution is approximate
- If the fitted model residual deviance is grossly different from the degrees of freedom, something is wrong.
  - Often our first suspect is "**overdispersion**".
  - Or something else wrong in the variance function

More rigorous overdispersion tests

A common way to 'discover' over- or underdispersion is to notice that the residual deviance is appreciably different from the residual degrees of freedom... This can be seriously misleading. The theory is asymptotic. (Venables & Ripley, MASS 4ed, p. 208)
Other suggestions in Smithson & Merkle

1. *sum of squared Pearson residuals as a* χ^2 *distributed variable*
2. *AER package dispersiontest.*

- However, I wouldn't fuss to much over a formal test. If residual deviance is noticeably more than degrees of freedom, then generally people expect you to move on to a more sophisticated model.

Another diagnostic suggestion

In the COUNT package, it is suggested to make a table like so

observed value	Proportion Observed	Proportion Predicted	Difference
0	.4	.2	.2
1	.15	.18	-0.3
2	.20	.15	.05
⋮			

Their function `poi.obs.pred` can produce that table for a poisson model.

Stop run examples here

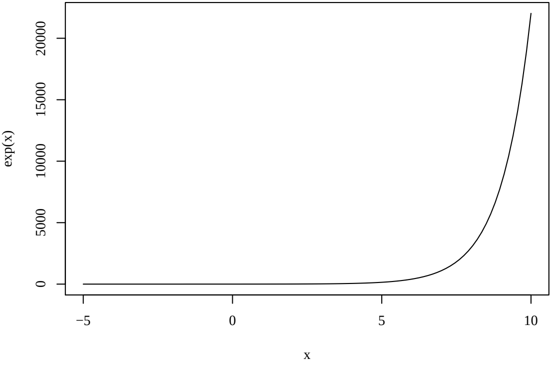
STOP. Run examples here, probably

1. Soccer Scores
2. Sick days

Outline

Nonlinearity $\mu_i = exp(X_i\beta)$

- Small $X\beta$: no noticeable effect
- Larger $X_i\beta$: more noticeable effect



The effect of anything depends on everything else.

Interpretation: Think of Predicted Values

- The expected value of y_i depends on $\mu_i = exp(X_i\beta)$.

$$E(y_i|X_i) = exp(X_i\hat{\beta})$$

- If the k'th variable changes, the impact is

$$\frac{\partial E(y_i|X_i)}{\partial x_k} = \hat{\beta}_k * E(y_i|X_i)$$

$$= \hat{\beta}_k * exp(X_i\hat{\beta})$$

- Scott Long's excellent book *Regression Models for Categorical and Limited Dependent Variables* has many helpful interpretations

percentage change interpretation

- I usually get confused on this, lets try to stay close to the book:

Smithson & Merkle (p. 117)

$$\lambda_i = exp(\beta_0)exp(\beta_1 X1_i)exp(\beta_2 X2_i)\dots$$

"Focusing on interpretation of, say, β_1, we want to compare the predicted value of $\lambda_i$at $X1_i$ and at $(X1_i + 1)$(with all other Xk held constant). The equation can be used to show that the prediction at $(X1_i + 1)$ equals the prediction at $X1_i$ times $exp(\beta_1)$. Thus, $exp(\beta_1)$ is associated with the percentage change in the predicted value of λ associated with every point increase in $X1_i$."

percentage change interpretation ...

- Let me see if I can work that out with a 2 predictor model

$$\lambda 1_i = exp(\beta_0)exp(\beta_1 X1_i)exp(\beta_2 X2_i)\ldots$$
$$\lambda 2_i = exp(\beta_0)exp(\beta_1(X1_i+1))exp(\beta_2 X2_i)$$

$$\lambda 2_i/\lambda 1_i = \frac{\cancel{exp(\beta_0)}exp(\beta_1(X1_i+1))\cancel{exp(\beta_2 X2_i)}}{\cancel{exp(\beta_0)}exp(\beta_1 X1_i)\cancel{exp(\beta_2 X2_i)}}$$
$$= e^{\beta_1 X1_i+\beta_1}e^{-\beta_1 X1_i}$$
$$= e^{\beta_1}$$

So, the "proportional increase" due to a 1 unit change of the predictor is $exp(\beta_1)$.

- However, put $\lambda 1$ on RHS and we see

$$\lambda 2_i = e^{\beta_1}\lambda 1_i$$

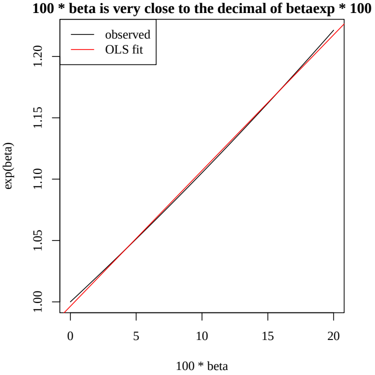
The predicted $\lambda 2_i$ is proportionally related to $e^{\beta_1}\lambda 1_i$.

percentage change interpretation ...

- if $\beta_1 = 1$, then $\lambda 2_i = \lambda 1_i$
- if $\beta_1 = 0.1$, then $\lambda 2_i = exp(0.1)\lambda 1_i = 1.1051\lambda 1_i$

- A proportional change of 1.1051 is the same as a 10.51 percent change.
  - If we want the percent increase, multiply by 100.
- Smithson & Merkle mention that if the β coefficient is small ($|\beta| < 2$), then a rule of thumb applies: 100β is very close to the decimal part of $exp(\beta_1) \times 100$.
- I've just verified that in R by running this:

percentage change interpretation ...



The slope of the red line is about 0.011.

Summary Poisson Shortcomings

1. **Overdispersion**: Suppose for a given $\mathbf{X}_i$ the predicted value is 7 but there are scores ranging from 0 to 60 and the variance of observations for that $\mathbf{X_i}$is 20. That's the problem that most of the researchers discuss
2. **Zero-Inflation**: Observe more 0's than expected, given λ_i.
3. Underdispersion: Less widely discussed

Example 1: Overdispersion in action

Julian Faraway's R package "faraway" includes a data set "gavote".

A Poisson Regression

```
myPois1<- glm(undercount ~ rural + perAA + equip,
    family = poisson, data = gavote)
summary(myPois1)
```

```
Call:
glm(formula = undercount ~ rural + perAA + equip, family = poisson,
    data = gavote)

Deviance Residuals:
    Min        1Q    Median        3Q       Max
-77.623   -11.794   -2.844    8.385   165.005

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.536222   0.010373  437.32   <2e-16 ***
ruralurban   1.216001   0.007586  160.30   <2e-16 ***
perAA        2.213451   0.020352  108.76   <2e-16 ***
equipOS-CC   0.754020   0.010677   70.62   <2e-16 ***
equipOS-PC   0.937110   0.011196   83.70   <2e-16 ***
equipPAPER  -1.504971   0.094463  -15.93   <2e-16 ***
equipPUNCH   1.556716   0.010217  152.37   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A Poisson Regression ...

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 184237  on 158  degrees of freedom
Residual deviance:  77702  on 152  degrees of freedom
AIC: 78893

Number of Fisher Scoring iterations: 6
```

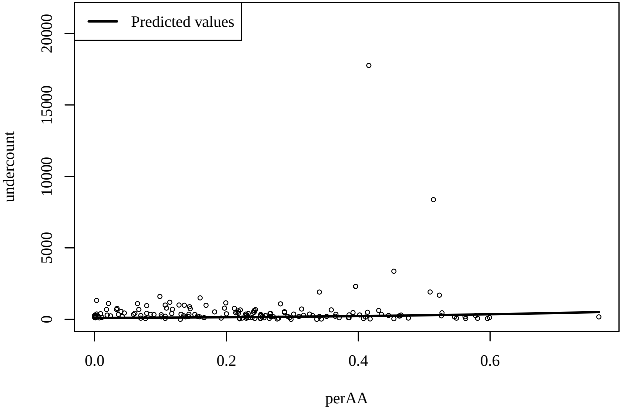
```
predictOMatic(myPois1)
```

```
$rural
  rural     perAA equip      fit
1 rural 0.2429811 LEVER 159.8201
2 urban 0.2429811 LEVER 539.1800

$perAA
  rural  perAA equip      fit
1 rural 0.0000 LEVER  93.3375
2 rural 0.1115 LEVER 119.4648
3 rural 0.2330 LEVER 156.3279
4 rural 0.3480 LEVER 201.6438
5 rural 0.7650 LEVER 507.5077
```

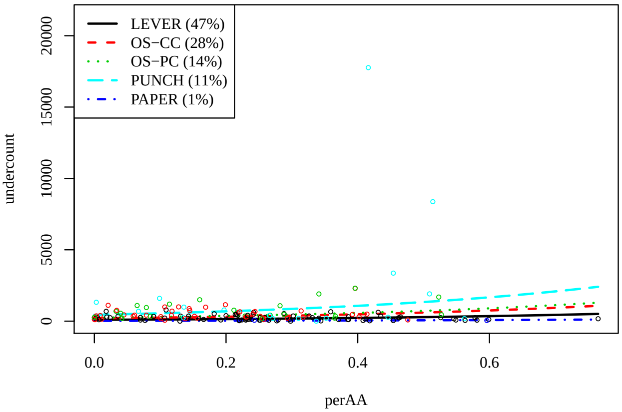
A Poisson Regression ...

```
$equip
  rural     perAA equip       fit
1 rural 0.2429811 LEVER 159.82010
2 rural 0.2429811 OS-CC 339.70184
3 rural 0.2429811 OS-PC 407.95591
4 rural 0.2429811 PUNCH 758.06202
5 rural 0.2429811 PAPER  35.48384
```

A Couple of Plots, Because I Can



A Couple of Plots, Because I Can



Outline

Insert a Random Effect

- Supplement the linear predictor $\mathbf{X}_i \boldsymbol{\beta}$ with an added noise v_i.
- We insert $\mathbf{X}_i \boldsymbol{\beta} + v_i$ into the Poisson data generating process where we previously inserted $\mathbf{X}_i \boldsymbol{\beta}$.
- This is additional individual-level randomness:
  - Two observations with same $\mathbf{X}_i$ can have different Poisson probabilities because v_i differs among individuals.
- The terms "frailty" or "heterogeneity" are used for this individual-level random noise.

Log Link

- We will keep the assumption of the log link, so $\mu_i = exp(\mathbf{X}_i\boldsymbol{\beta} + v_i)$
- We re-phrase the data generating process from this:

$$Pr(y_i|\mathbf{X}_i\boldsymbol{\beta}) = \frac{exp(-e^{\mathbf{X}_i\boldsymbol{\beta}})(e^{\mathbf{X}_i\boldsymbol{\beta}})^{y_i}}{y_i!}$$

to this:

$$Pr(y_i|\mathbf{X}_i\boldsymbol{\beta}) = \frac{exp(-e^{\mathbf{X}_i\boldsymbol{\beta}+v_i})(e^{\mathbf{X}_i\boldsymbol{\beta}+v_i})^{y_i}}{y_i!} \tag{11}$$

Math Reminders: Is your error additive or multiplicative?

Remember this is a Mathematical law

$$e^{\mathbf{X}_i\boldsymbol{\beta}+v_i} = e^{\mathbf{X}_i\boldsymbol{\beta}} \times e^{v_i} \tag{12}$$

Let $ln(\delta_i) = v_i$which implies $\delta_i = e^{v_i}$.

- (12) is the same as

$$e^{\mathbf{X}_i\boldsymbol{\beta}} \times \delta_i \tag{13}$$

- or

$$e^{\mathbf{X}_i\boldsymbol{\beta}+ln(\delta_i)} \tag{14}$$

- One can think of this new randomness as either
  1. Adding the log of δ into $X_i\beta$
  2. Multiplying $exp(X_i\beta)$

Error term should be neutral!

- In the multiplicative story, $e^{\mathbf{X}_i\boldsymbol{\beta}} \times e^{v_i} = e^{\mathbf{X}_i\boldsymbol{\beta}} \times \delta_i$, we want

$$E[\delta_i] = 1$$

- In the additive story $e^{\mathbf{X}_i\boldsymbol{\beta}+ln(\delta_i)} = e^{\mathbf{X}_i\boldsymbol{\beta}+v_i}$

$$E[ln(\delta_i)] = E[v_i] = 0$$

- Remember that $ln(1) = 0$? These things are consistent!
- "On average" the extra error term has "no effect".

Gamma error is a natural candidate

- We require a random variable δ_i with
  - expected value of 1
  - $\delta_i > 0$

$$f_y(y|shape, exp(X\beta)) = \frac{\Gamma(shape+y)}{\Gamma(shape)y!} \bullet \frac{(exp(X\beta)^y shape^{shape}}{(exp(X\beta)+shape)^{shape+y}}$$

(Venables and Ripley, 4th ed, p. 206)

- We'll use the Gamma distribution.
- Technical convenience: it is known (in a math stats book) that $Poisson(\mathbf{X}_i\boldsymbol{\beta} + log(\delta_i))$ is distributed as a Negative Binomial random variable.

NB: A Mixture Distribution

- If the outcome is conditional Poisson (depending on δ_i),

$$y_i \,|\, \delta_i \sim Poisson(exp(X_i\beta) * \delta_i)$$

- Then y_i follows a Negative Binomial Distribution.

Negative Binomial: a discrete PMF that has separate expected value and scale parameters

http://pj.freefaculty.org/guides/stat/Distributions/DistributionWriteups/NegativeBinomial/NegativeBinomial.pdf

- Story: it is the number of "tails" that occur before a "heads" in a sequence of coin flips, with a possibly biased coin. (Think "tails" = nothing observed yet or no failure yet and "heads" = something happened, failure observed, etc.)

Simpler Version of Gamma

- A Gamma distribution has two parameters, shape and scale, $Gamma(shape, scale)$.

http://pj.freefaculty.org/guides/stat/Distributions/DistributionWriteups/Gamma/Gamma-02.pdf

- Here we simplify by assuming $scale = 1/shape$, so a two parameter random distribution

$$\delta_i \sim Gamma(shape, scale)$$

becomes a one parameter distribution

$$\delta_i \sim Gamma(shape, 1/shape)$$

- Because for Gamma $E[\delta_i] = shape \cdot scale$, then this special one-parameter model has $E[\delta] = 1.0$.

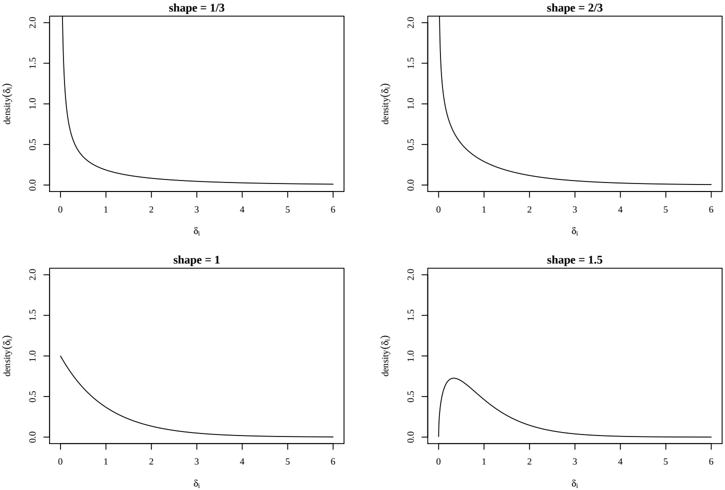
Simpler Version of Gamma ...

- Under the restriction between $shape$ and $scale$,

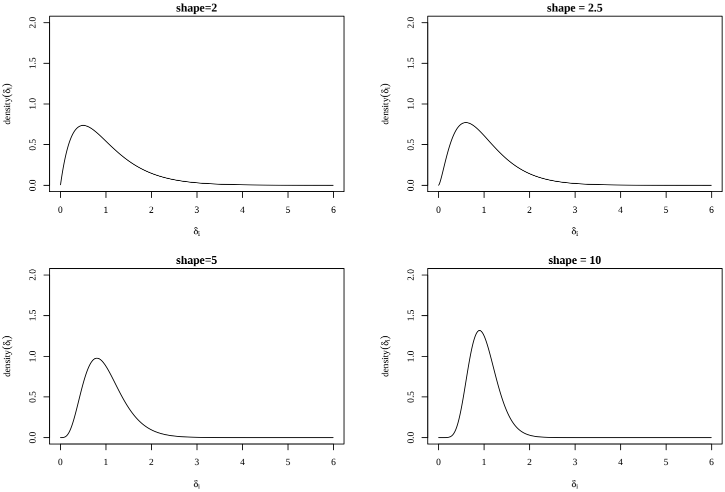
$$Var[\delta_i] = 1/shape$$

- So, as the shape parameter gets bigger, the variance of δ_i gets smaller, and eventually vanishes

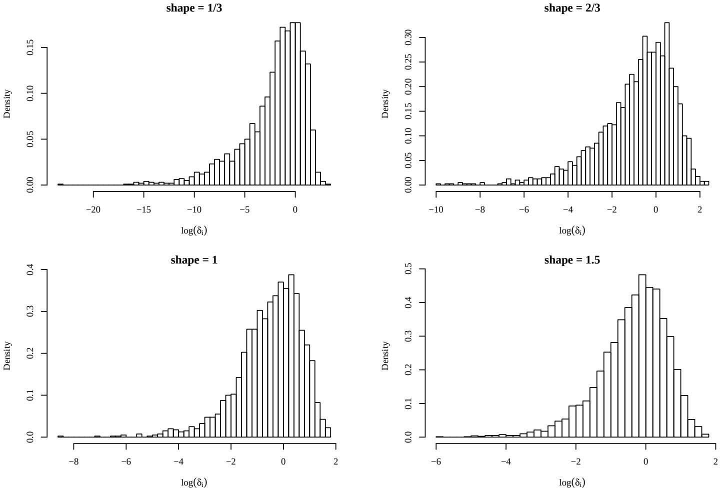
Gamma PDF assuming shape*scale=1



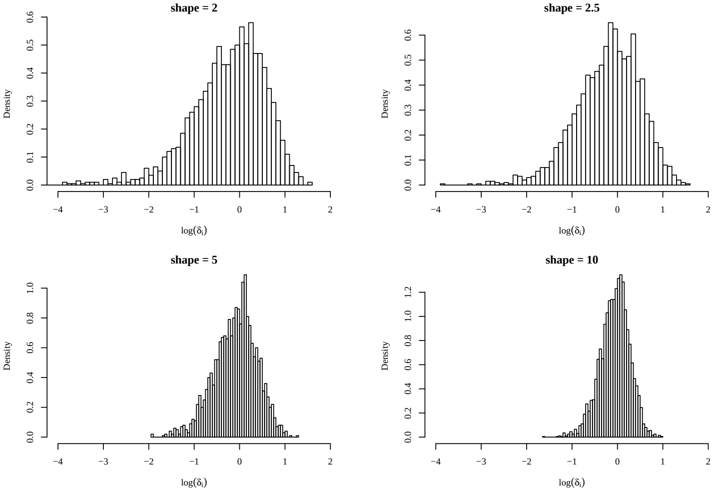
Gamma PDF assuming shape*scale=1



The density of the additive errors, $ln(\delta)$



The density of the additive errors, $ln(\delta)$



Estimating

- Fitting is an iterative, two-stage process.
  - The shape estimate is chosen
  - Then the slope parameters are estimated.
- Repeat until estimates converge to stable values.
- The MASS package for R provides a procedure "glm.nb" which will do maximum likelihood to estimate the b's and the shape parameter. (In Venables & Ripley, p. 207, the "shape" parameter is called θ).

Neg. Binomial $\Longleftrightarrow$ Poisson

- Neg Binomial has the same expected value as the Poisson:

$$E(y_i) = exp(X_i\beta)$$

- But more variance

$$Var(y_i) = exp(X\beta) + exp(X\beta)^2/shape$$

- If $shape \rightarrow \infty$, then the variance of y_i tends to $exp(X_i\beta)$, same as Poisson.

Outline

Do we really need to bother with this?

- If the data is produced by a "mixed model" of the NB type, a Poisson regression model's estimates
  - are inefficient and
  - have bad standard errors (hypo tests are wrong).
- Some authors will keep the parameter estimates, but use "robust" standard errors (to avoid introducing another parameter for the mixture variance).
- The counter-argument is that we should generally begin with the NB model.

An informative LR test

- Fit both NB and Poisson, then try to decide if the NB is a significant improvement
- The Poisson model is "nested" inside the NB model, but $shape \to \infty$ to lead to equivalence.
- This is a "nonstandard" hypothesis test because null hypo is on boundary of parameter space.
- Achim Zeileis suggests a likelihood ratio test with the lrtest in the lmtest package http://stats.stackexchange.com/questions/127505/compare-poisson-and-negative-binomial-regression-with-lr-test
- Long p. 237 discusses other tests.

Example 1: Continued

```
library(faraway)
library(MASS)
gavote$undercount <- gavote$ballots - gavote$votes
myNB1<- glm.nb(undercount ~ rural + perAA +
    equip, data = gavote)
summary(myNB1)
```

```
Call:
glm.nb(formula = undercount ~ rural + perAA + equip, data = gavote,
    init.theta = 1.392872977, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.0391  -0.9568  -0.2554   0.3143   3.5058

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   5.2137     0.1619  32.202  < 2e-16 ***
ruralurban    0.9094     0.1660   5.478 4.30e-08 ***
perAA         0.4746     0.4444   1.068  0.28556
equipOS-CC    0.5481     0.1696   3.231  0.00124 **
equipOS-PC    0.9117     0.2102   4.338 1.44e-05 ***
```

Example 1: Continued ...

```
equipPAPER   -1.3644     0.6177  -2.209  0.02720 *
equipPUNCH    1.4606     0.2442   5.982 2.20e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.3929) family taken to be
     1)

    Null deviance: 358.48  on 158  degrees of freedom
Residual deviance: 180.33  on 152  degrees of freedom
AIC: 2228.8

Number of Fisher Scoring iterations: 1


              Theta:  1.393
          Std. Err.:  0.146

 2 x log-likelihood:  -2212.833
```

```
predictOMatic(myNB1)
```

Example 1: Continued ...

```
$rural
  rural      perAA equip       fit
1 rural 0.2429811 LEVER 206.2314
2 urban 0.2429811 LEVER 512.0371

$perAA
  rural  perAA equip      fit
1 rural 0.0000 LEVER 183.7701
2 rural 0.1115 LEVER 193.7562
3 rural 0.2330 LEVER 205.2568
4 rural 0.3480 LEVER 216.7704
5 rural 0.7650 LEVER 264.2083

$equip
  rural      perAA equip       fit
1 rural 0.2429811 LEVER 206.23142
2 rural 0.2429811 OS-CC 356.76262
3 rural 0.2429811 OS-PC 513.21477
4 rural 0.2429811 PUNCH 888.53690
5 rural 0.2429811 PAPER  52.69805
```

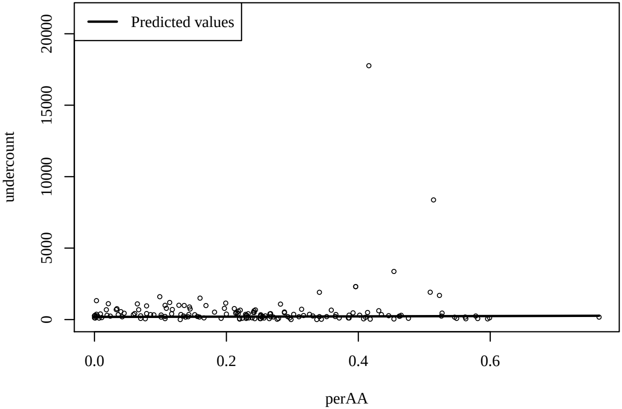
Compare output side-by-side

```
outreg(list("Poisson" = myPois1, "Neg. Binom" =
    myNB1), tight=FALSE)
```

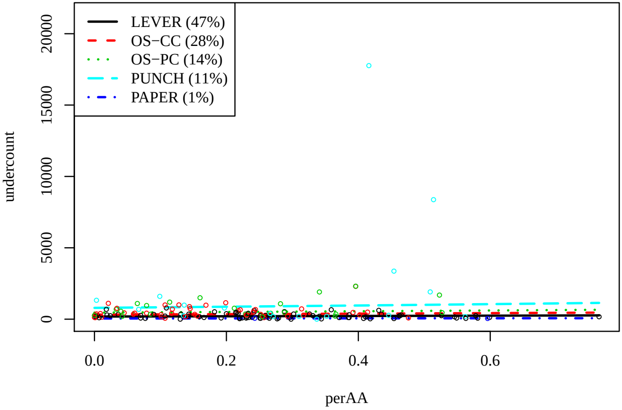
	Poisson		Neg. Binom	
	Estimate	(S.E.)	Estimate	(S.E.)
(Intercept)	4.536***	(0.010)	5.214***	(0.162)
ruralurban	1.216***	(0.008)	0.909***	(0.166)
perAA	2.213***	(0.020)	0.475	(0.444)
equipOS-CC	0.754***	(0.011)	0.548**	(0.170)
equipOS-PC	0.937***	(0.011)	0.912***	(0.210)
equipPAPER	-1.505***	(0.094)	-1.364*	(0.618)
equipPUNCH	1.557***	(0.010)	1.461***	(0.244)
N	159		159	
Deviance	77702.066		180.331	
$-2LLR(Model \chi^2)$	106535.170***		178.152***	

$*p \le 0.05{**}\ p \le 0.01{***}p \le 0.001$

Plots not too much different



Plots not too much different



Compare output side-by-side

```
outreg(list("Poisson" = myPois1, "Neg. Binom" =
    myNB1), tight=FALSE)
```

	Poisson		Neg. Binom	
	Estimate	(S.E.)	Estimate	(S.E.)
(Intercept)	4.536***	(0.010)	5.214***	(0.162)
ruralurban	1.216***	(0.008)	0.909***	(0.166)
perAA	2.213***	(0.020)	0.475	(0.444)
equipOS-CC	0.754***	(0.011)	0.548**	(0.170)
equipOS-PC	0.937***	(0.011)	0.912***	(0.210)
equipPAPER	-1.505***	(0.094)	-1.364*	(0.618)
equipPUNCH	1.557***	(0.010)	1.461***	(0.244)
N	159		159	
Deviance	77702.066		180.331	
$-2LLR(Model \chi^2)$	106535.170***		178.152***	

$*p \le 0.05 ** \, p \le 0.01 *** p \le 0.001$

Hypo test

```
anova(myPois1, myNB1, test = "LR")
```

```
Analysis of Deviance Table

Model 1: undercount ~ rural + perAA + equip
Model 2: undercount ~ rural + perAA + equip
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       152      77702
2       152        180  0    77522
```

```
library(lmtest)
lrtest(myPois1, myNB1)
```

```
Likelihood ratio test

Model 1: undercount ~ rural + perAA + equip
Model 2: undercount ~ rural + perAA + equip
  #Df LogLik Df Chisq Pr(>Chisq)
1   7 -39439
2   8  -1106  1 76666  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Start at the Beginning, Again

- Poisson is "underdispersed" compared to the data

$$Pr(y_i|\mathbf{X}_i\boldsymbol{\beta}) = \frac{exp(-e^{\mathbf{X}_i\boldsymbol{\beta}})(e^{\mathbf{X}_i\boldsymbol{\beta}})^{y_i}}{y_i!}$$

- Insert an additive error v_i:

$$Pr(y_i|\mathbf{X}_i\boldsymbol{\beta}) = \frac{exp(-e^{\mathbf{X}_i\boldsymbol{\beta}+v_i})(e^{\mathbf{X}_i\boldsymbol{\beta}+v_i})^{y_i}}{y_i!} \tag{15}$$

- Now we'll say that's Normally distributed, $v_i \sim N(0, \sigma_v^2)$.

The Too-Many-Zeroes Problem

- Poisson Count models allow for “a lot of 0’s”
- NB model allows even more 0’s.
- Suppose the data has still more 0’s.

Process Explanations

- How many packs of cigarettes did you buy this month?
    - many respondents say 0, they don't want cancer (thanks for asking)
    - Some will say 1, 2, ..., 10, or 40

Two-Part Model

- The "zero-inflated" Poisson or Negative Binomial models are tailored to that kind of story.
- The count that will be eventually observed is related to $exp(X_i\beta)$.
- However, the observed count will be 0 unless $X_i\beta$ is above a threshold.
  - Call that "stage 1"
  - Modeled by any dichotomous regression model, such as logit or probit.
- Then a Poisson or Negative Binomial model is used to predict the count that is observed.

Two Part Models

- This is a “mixture model”, supposing there are two types of cases.
  - Probability ψ_i: this is a case that will have count $= 0$
  - Probability $(1 - \psi_i)$: this is a case that will have count determined by a draw from a Poisson or Negative Binomial process.
- Details.
  - Is the linear predictor in your logit model the same formula that is used in your count model?
    - Should the predictors be the same ones that are used in the Poisson or NB regression?
    - Should the coefficients be the same?
- From a data-generating point of view, some mystery remains
  - If a non-zero count is observed, then we know for sure the case exceeded the hurdle
  - But if a 0 is observed, then we are unsure which category that row belongs with.

Two Part Models ...

    - The stage 1 random draw may pass cases that will still have counted values of 0.
  - Hence, there are 2 processes through which we might observe $y_i = 0$.
    1. Stage 1 gives 0
    2. Stage 1 Poisson draw gives 0.
- Jargon alert: a "hurdle model" (as in the R package pscl) is one where stage 1 determines whether count is greater than 0, which is a different logic.

ZIP model: Preparing for likelihood analysis

1. the probability of observing a 0 is

$$P(y_i = 0|X_i) = \psi_i + (1 - \psi_i) * exp(-exp(X_i\beta))$$

(Write out the poisson for y=0 to understand the last term).

2. The probability of a non-zero outcome is same as the Poisson or Neg Binomial model, multiplied by $(1 - \psi_i)$:

$$P(y_i|X_i) = (1 - \psi_i) * Poisson(X_i\beta)$$

Make up some test cases, explore

- In R, the best package for exploring this is "pscl" (Simon Jackman et al)
    - zeroinfl is estimator
    - Compare to pscl::hurdle
    - vuong test is a non-nested test for the statistical significance of the difference between a zero-inflated model and its counterpart.

I'll have a rate handout

I made a report to a client on Poisson models for rates, I'll copy that.

The bible of count data is Cameron (1998); Cameron & Trivedi (2013)

I first appreciated the details by reading Long (1997) and King (1988).

Cameron, A. C. (1998). *Regression Analysis of Count Data*. Number no. 30 in Econometric Society monographs. Cambridge ; New York: Cambridge University Press.

Cameron, A. C. & Trivedi, P. K. (2013). *Regression Analysis of Count Data*. Cambridge ; New York, NY: Cambridge University Press, 2 edition edition.

King, G. (1988). Statistical Models for Political Science Event Counts: Bias in Conventional Procedures and Evidence for the Exponential Poisson Regression Model. *American Journal of Political Science*, 32(3), 838–863.

Long, J. S. (1997). *Regression Models for Categorical and Limited Dependent Variables*. Number 7 in Advanced quantitative techniques in the social sciences. Thousand Oaks: Sage Publications.

McCullagh, P. & Nelder, J. A. (1989). *Generalized Linear Models, Second Edition*. Boca Raton: Chapman and Hall/CRC, 2 edition edition.

Session

```r
sessionInfo()
```

```
R version 3.5.1 (2018-07-02)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.10

Matrix products: default
BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.8.0
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.8.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] stationery_0.98.5.5
```

Session ...

```
loaded via a namespace (and not attached):
 [1] Rcpp_0.12.17     rprojroot_1.3-2 digest_0.6.15    plyr_1.8.4
 [5] backports_1.1.2 xtable_1.8-2     magrittr_1.5     stats4_3.5.1
 [9] evaluate_0.10.1 zip_1.0.0        stringi_1.2.3    pbivnorm_0.6.0
[13] openxlsx_4.1.0  rmarkdown_1.10   tools_3.5.1      stringr_1.3.1
[17] foreign_0.8-71  kutils_1.51      compiler_3.5.1   mnormt_1.5-5
[21] htmltools_0.3.6 knitr_1.20       lavaan_0.6-1
```