

Poisson Regression

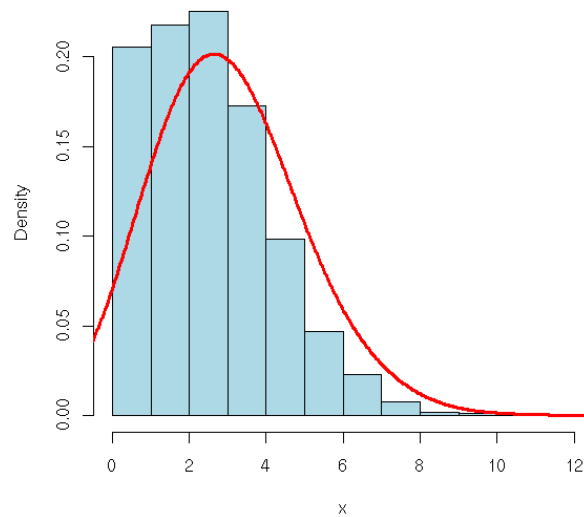
Using count data

What if we have count data?

- ▶ Suppose we have data that are *aggregate counts* of some event over a given area
 - ▶ Area-level vs. individual-level data
- ▶ Nature of count data
 - ▶ Discrete, skewed distribution
 - ▶ High proportion of zero outcomes
 - ▶ Always > 0
- ▶ Why OLS won't work
 - ▶ The relationship between X and Y is nonlinear
 - ▶ Counts are heteroskedastic
 - ▶ Can't predict non-negative values

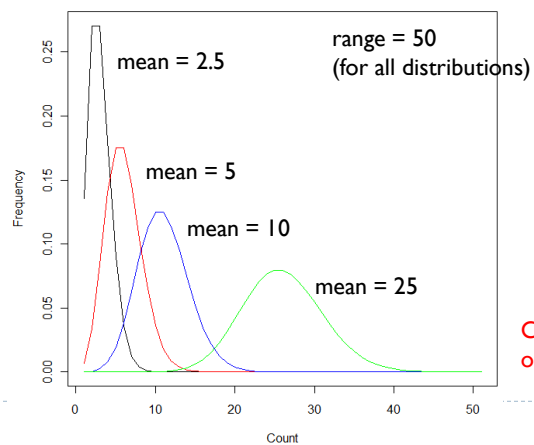


What is a Poisson distribution?



“Large” vs. “small” Poissons

- ▶ “Large Poissons are like Gaussians, but small Poissons are quite different”
- ▶ As mean increases, Poisson approximates Normal distribution



Can we use OLS regression on “large” Poissons?



Link functions

- ▶ Remember, for Generalized Linear Models, we use the link function to transform y :
 - ▶ Normal: $G(y) = y$ (identity link)
 - ▶ Binomial/logistic: $G(y) = \log\left(\frac{P}{1-P}\right)$
 - ▶ Poisson: $G(y) = \log(y)$



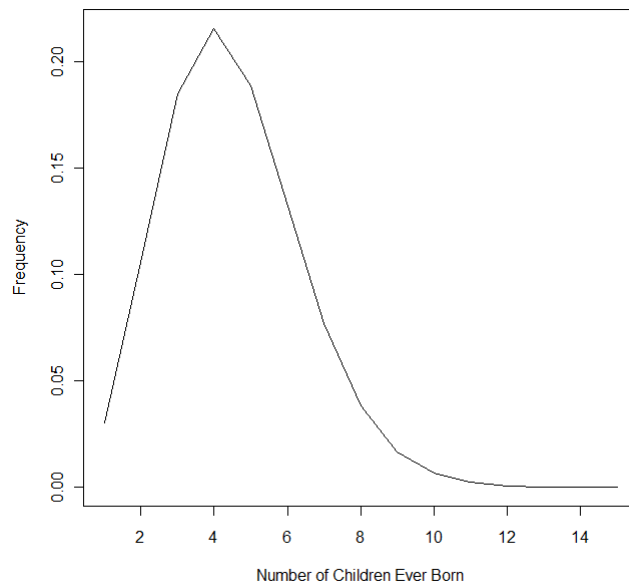
Interpretation

- ▶ If we take the transformed y and add a regression equation, we get a Poisson regression:
$$\log(y) = \alpha + \beta x$$
- ▶ As in least-squares regression, the relationship between the $\log(y)$ and x is assumed to be linear
 - ▶ $\log(y)$ changes linearly as a function of explanatory variables
 - ▶ Or one unit change in $y = \exp(x)$ change in x



Example: children ever born

- ▶ The dataset has 70 rows representing group-level data on the number of children ever born to women in Fiji:
 - ▶ Number of children ever born
 - ▶ Number of women in the group
 - ▶ Duration of marriage
 - ▶ 1=0-4, 2=5-9, 3=10-14, 4=15-19, 5=20-24, 6=25-29
 - ▶ Residence
 - ▶ 1=Suva (capital city), 2=Urban, 3=Rural
 - ▶ Education
 - ▶ 1=none, 2=lower primary, 3=upper primary, 4=secondary+



Poisson regression in R

```
> ceb1<-glm(y ~ educ + res, offset=log(n), family = "poisson",
  data = ceb)
```

Need to account for different population sizes in each area/group unless data are from same-size populations

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.43029	0.01795	79.691	<2e-16 ***
educnone	0.21462	0.02183	9.831	<2e-16 ***
educsec+	-1.00900	0.05217	-19.342	<2e-16 ***
educupper	-0.40485	0.02956	-13.696	<2e-16 ***
resSuva	-0.05997	0.02819	-2.127	0.0334 *
resurban	0.06204	0.02442	2.540	0.0111 *

Null deviance: 3731.5 on 69 degrees of freedom
 Residual deviance: 2646.5 on 64 degrees of freedom
 AIC: Inf



Assessing model fit

1. Examine AIC score – smaller is better
2. Examine the deviance as an approximate goodness of fit test

► Expect the residual deviance/degrees of freedom to be approximately 1

```
> ceb2$deviance/ceb2$df.residual
[1] 41.35172
```

3. Compare residual deviance to a χ^2 distribution

```
> pchisq(2646.5, 64, lower=F)
[1] 0
```



Interpretation

$$\log(y) = 1.43 + .21x_{edunone} - 1.0x_{eduse+} - 0.41x_{eduup} - 0.06x_{resSuva} + 0.06x_{resurb}$$

- ▶ The predicted number of children for a women with no education living in Suva is given by

$$\begin{aligned}\log(y) &= 1.43 + .21(1) - 1.0(0) - 0.41(0) - 0.06(1) + 0.06(0) \\ &= 1.58 \\ \exp(1.58) &= 4.85\end{aligned}$$

- ▶ The predicated number of children for a woman with a secondary education, living in a rural area is:

$$\begin{aligned}\log(y) &= 1.43 + .21(0) - 1.0(1) - 0.41(0) - 0.06(0) + 0.06(0) \\ &= .43 \\ \exp(.43) &= 1.53\end{aligned}$$



Model fitting: analysis of deviance

- ▶ Similar to logistic regression, we want to compare the differences in the size of residuals between models

```
> ceb1<-glm(y~educ, family="poisson", offset=log(n), data=
ceb)
> ceb2<-glm(y~educ+res, family="poisson", offset=log(n),
data= ceb)

> 1-pchisq(deviance(ceb1)-deviance(ceb2),
df.residual(ceb1)-df.residual(ceb2))
[1] 0.0007124383
```

- ▶ Since the p-value is small, there is evidence that the addition of **res** explains a significant amount (more) of the deviance



Overdispersion in Poisson models

- ▶ A characteristic of the Poisson distribution is that its mean is equal to its variance
- ▶ Sometimes the observed variance is greater than the mean
 - ▶ Known as overdispersion
 - ▶ Poisson model may not be appropriate
 - ▶ Common reason is omission of relevant lvs
- ▶ Another common problem with Poisson regression is **excess zeros**
 - ▶ Are more zeros than a Poisson regression would predict



Overdispersion

- ▶ Use `family="quasipoisson"` instead of `"poisson"` to estimate the dispersion parameter
- ▶ Doesn't change the estimates for the coefficients, but may change their standard errors
 - ▶ Test statistics and their p-values
 - ▶ Adjusting the interpretation of coefficients to take account of the over-dispersion

```
> ceb2<-glm(y~educ+res, family="quasipoisson",
  offset=log(n), data=ceb)
```



Poisson vs. quasipoisson

Family = “poisson”

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.43029    0.01795  79.691 <2e-16 ***
educnone     0.21462    0.02183   9.831 <2e-16 ***
educsec+    -1.00900    0.05217 -19.342 <2e-16 ***
educupper   -0.40485    0.02956 -13.696 <2e-16 ***
resSuva     -0.05997    0.02819  -2.127  0.0334 *
resurban     0.06204    0.02442   2.540  0.0111 *
---
(Dispersion parameter for poisson family taken to be
1)
```

```
Null deviance: 3731.5 on 69 degrees of freedom
Residual deviance: 2646.5 on 64 degrees of freedom
```

Family = “quasipoisson”

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.43029    0.10999  13.004 < 2e-16 ***
educnone     0.21462    0.13378   1.604  0.11358
educsec+    -1.00900    0.31968  -3.156  0.00244 **
educupper   -0.40485    0.18115  -2.235  0.02892 *
resSuva     -0.05997    0.17277  -0.347  0.72965
resurban     0.06204    0.14966   0.415  0.67988
---
(Dispersion parameter for quasipoisson taken to be
37.55359)
```

```
Null deviance: 3731.5 on 69 degrees of freedom
Residual deviance: 2646.5 on 64 degrees of freedom
```



Models for overdispersion

- ▶ When overdispersion is a problem, use a negative binomial model
 - ▶ Will adjust β estimates and standard errors

```
> library(MASS)
> library(lmtest)
> ceb.nb <- glm.nb(y~educ+res+offset(log(n)), data= ceb)
OR
> ceb.nb<-glm.nb(ceb2)
> summary(ceb.nb)
```



NB model in R

```
glm.nb(formula = ceb2, x = T, init.theta = 3.38722121141125, link = log)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	1.490043	0.160589	9.279	< 2e-16 ***
educnone	0.002317	0.183754	0.013	0.98994
educsec+	-0.630343	0.200220	-3.148	0.00164 **
educupper	-0.173138	0.184210	-0.940	0.34727
resSuva	-0.149784	0.165622	-0.904	0.36580
resurban	0.055610	0.165391	0.336	0.73670

(Dispersion parameter for Negative Binomial(3.3872) family taken to be 1)

Null deviance: 85.001 on 69 degrees of freedom

Residual deviance: 71.955 on 64 degrees of freedom

AIC: 740.55

Theta: 3.387

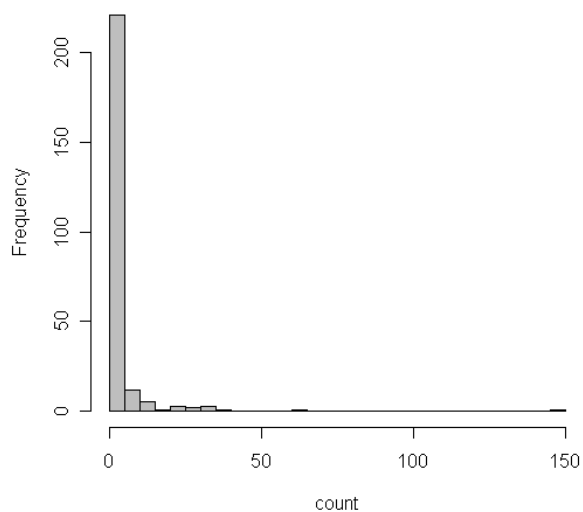
Std. Err.: 0.583

2 x log-likelihood: -726.555

```
> ceb.nb$deviance/ceb.nb$df.residual
[1] 1.124297
```



What if your data looked like...



Zero-inflated Poisson model (ZIP)

- ▶ If you have a large number of 0 counts...

```
> install.packages("pscl")  
> library(pscl)  
  
> ceb.zip <- zeroinfl(y~educ+res, offset=log(n),  
  data= ceb)
```

