36-617: Applied Linear Regression

Generalized Linear Models

Brian Junker
132E Baker Hall
brian@stat.cmu.edu

Announcements

- Midsemester grades - UPDATE
  - Mostly A’s (90+), some B’s (80+), lots of room for changes before the end of the semester.
  - Numerical score & letter grade in canvas gradebook under “Total”. If you are not getting info you want from canvas, please bug me.
- Midsemester evaluation – see “0 – midsemeter course evaluation” folder in files area on Canvas
  - Happy that you like the teaching and how accessible Lorenzo and I are for questions etc!
  - You all seem very overworked!
- Let’s talk about getting you through the next two weeks
  - I will also adjust my hw’s for the rest of the semester too

Outline

- Linear Regression, Logistic Regression
- Generalized Linear Models (GLM)
- Quick Taste of Ordered Logistic Regression
- Poisson Regression
  - Exposure and Offsets
  - Overdispersion
  - Zero-inflation
- Quick Taste of Ordered Logistic Regression
  - Many ways to model multinomial data
  - Example

Linear Regression, Logistic Regression

- The ***linear regression*** model is:

$$
\begin{aligned}
y_i &\overset{indep}{\sim} N(\theta_i, \sigma^2),\ i = 1, \ldots, n \\
\theta_i &= X_i\beta = \beta_0 X_{i0} + \cdots \beta_p X_{ip}
\end{aligned}
$$

  - ❑ Each $y_i \in (-\infty, \infty)$ has some mean θ_i = E[y_i]
  - ❑ Each θ_i has some linear structure
  - ❑ There is a statistical distribution N(*, σ^2) that describes unmodeled variation around θ_i = E[y_i]

- The ***logistic regression*** model is:

$$
\begin{aligned}
y_i &\overset{indep}{\sim} Binomial(n_i, p_i),\ i = 1, \ldots, n \\
\theta_i &= \log \frac{p_i}{1 - p_i} = X_i\beta = \beta_0 X_{i0} + \cdots \beta_p X_{ip}
\end{aligned}
$$

  - ❑ Each y $\in$ {0, 1} has some mean p_i = E[y_i]
  - ❑ Each θ_i = g(p_i) has some linear structure [g(p) = log p/(1-p) !]
  - ❑ There is a statistical distribution Binomial(n_i, *) that describes unmodeled variation around p_i = E[y_i]
  - ❑ (if $n_i = 1$ for all i, this is the *Bernoulli*(p_i) logistic regression model)

Generalized Linear Models

- The ***generalized linear model (glm)*** is:

$$\begin{aligned} y_i &\overset{indep}{\sim} f(y_i|\mu_i,\ldots),\ i=1,\ldots,n \\ \theta_i &= g(\mu_i) = X_i\beta = \beta_1 X_{i1} + \cdots \beta_k X_{ik} \end{aligned}$$

  - Each y_i has some mean $\mu_i = E[y_i]$
  - Each $\theta_i = g(\mu_i)$ has some linear structure [$g(\mu)$ is the "link function"]
  - There is a statistical distribution $f(y_i|\mu_i, \ldots)$ that describes unmodeled variation around $\mu_i = E[y_i]$
  - *There may be other parameters "…" in* $f(y_i|\mu_i, \ldots)$ *but the "main" parameter is* $\mu_i = g^{-1}(\theta_i) = g^{-1}(X_i\beta)$
- For *ordinary linear regression*
  - $f(y_i|\mu_i, \ldots) = N(\mu_i,\sigma^2)$ [$\mu_i = E[y_i]$]
  - $g(\mu) = \mu$ [the "identity link function"]
- For *logistic regression*
  - $f(y_i|p_i)$ = Binomial(n_i, p_i) or Bernoulli(p_i) [$p_i = E[y_i]$]
  - $g(p) = \log p/(1-p)$ [the "logit link function"]

Two Other Common GLM's

- Poisson Regression Model
  - $y_i \epsilon \{0, 1, 2, 3, \ldots\}$
  - $f(y_i | \lambda_i) = \text{Poiss}(\lambda_i)$ $[\lambda_i = E[y_i]]$
  - $\theta_i = \log(\lambda_i) = X_i \beta$
- Multinomial Logit Model
  - $y_i \epsilon \{1, 2, \ldots, K\}$
  - $f(y_i \mid \pi_{i1}, \ldots, \pi_{iK})$: $y_i \sim Multinom(\pi_{i1}, \ldots, \pi_{iK})$
  - We need to invent some logits for these π_{ik}'s…
  - This leads to many different "***<u>multinomial regression</u>***" models…

Poisson Regression

- Poisson Regression Model
  - ❑ $y_i \epsilon \{0, 1, 2, 3, \ldots\}$
  - ❑ $f(y_i | \lambda_i) = \text{Poiss}(\lambda_i)\ [\lambda_i = E[y_i]]$
  - ❑ $\theta_i = \log(\lambda_i) = X_i \beta$
- We will fit this model to data, and then look at some modifications of the model involving
  - ❑ offsets
  - ❑ overdispersion
  - ❑ zero-inflation

  (the same kinds of modifications can be helpful with logistic regression and other GLM's…)

Poisson Regression – The Data

- Data from an experiment on the effectiveness of an "integrated pest management system" in apartment buildings in a particular city

```
roachdata <- read.csv ("roachdata.csv")

str(roachdata)

 'data.frame':   262 obs. of  6 variables:
  $ X        : int  1 2 3 4 5 6 7 8      [observation number]
  $ y        : int  153 127 7 7 0 0      [# of roaches trapped after expmt]
  $ roach1   : num  308 331.25 1.67      [# of roaches before experiment]
  $ treatment: int  1 1 1 1 1 1 1 1      [pest mgmt tx in this apt bldg?]
  $ senior   : int  0 0 0 0 0 0 0 0      [apts restricted to sr citzns?]
  $ exposure2: num  0.8 0.6 1 1 1.14     [avg # of trap-days per apt for y]
```

Poisson Regression – Fitting the Model

```
> glm.0 <- glm (y ~ roach1 + treatment + senior,
  family=poisson)
> summary(glm.0)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.136e+00  2.124e-02  147.64   <2e-16 ***
roach1       6.444e-03  8.832e-05   72.97   <2e-16 ***
treatment   -5.124e-01  2.465e-02  -20.79   <2e-16 ***
senior      -3.760e-01  3.355e-02  -11.21   <2e-16 ***
```

$$
\begin{aligned}
\lambda_i &= E[Y_i] \\
\log \lambda_i &= 3.14 + 0.0064(roach1) - 0.5(treatment) - 0.38(senior) \\
\lambda_i &= \exp(3.14 + 0.0064(roach1) - 0.5(treatment) - 0.38(senior)) \\
&= \exp(3.14)\exp(0.0064(roach1))\exp(-0.5(treatment))\exp(-0.38(senior))
\end{aligned}
$$

Poisson Regression – Interpreting the Coefficients

- ***Intercept = 3.14***: exp(3.14) = 23.10 is the average # of roaches trapped *after* the experiment, in an apt bldg with no roaches *before* the experiment (roach1=0), no treatment (treatment=0) and not a seniors' building (senior=0).
  - In this case there are about 60 buildings with no roaches at the start of the experiment, so this is probably a meaningful description
- ***roach1 = 0.00644***: exp(0.00644) = 1.006 is the factor increase in average roaches caught *after* the experiment, per roach caught *before* the experiment (does this make sense?).
- ***treatment = -0.512***: exp(-0.512) = 0.60 is the factor reduction in average roaches caught *after* the experiment, due to treatment
- ***senior = -0.38***: exp(-0.38) = 0.68 is the factor reduction in the average roaches caught *after* the experiment, due to being a senior bldg

Poisson Regression - Exposure

- We have not made use of exposure2 = average number of trap-days
  - If twice as many traps, expect to catch 2x roaches
  - If 3 times as many days, expect to catch 3x roaches
- To accommodate this multiplicative effect, we can try

$$\lambda_i = u_i e^{X_i \beta}$$

where u_i = exposure2.

Poisson Regression – Exposure

- Taking logs, the “linear regression” form is

$$\log(\lambda_i) = \log(u_i) + X_i\beta$$

  This is like including $\log(u_i)$ in the model, and basically forcing its coefficient to be exactly 1.

- In R we accomplish this with the “offset” argument
- This makes interpretation of the coefficients easier
  - coefficients measure deviations from expected counts under the various numbers of trap-days
  - This “unconfounds” exposure from treatment, bldg type, etc.

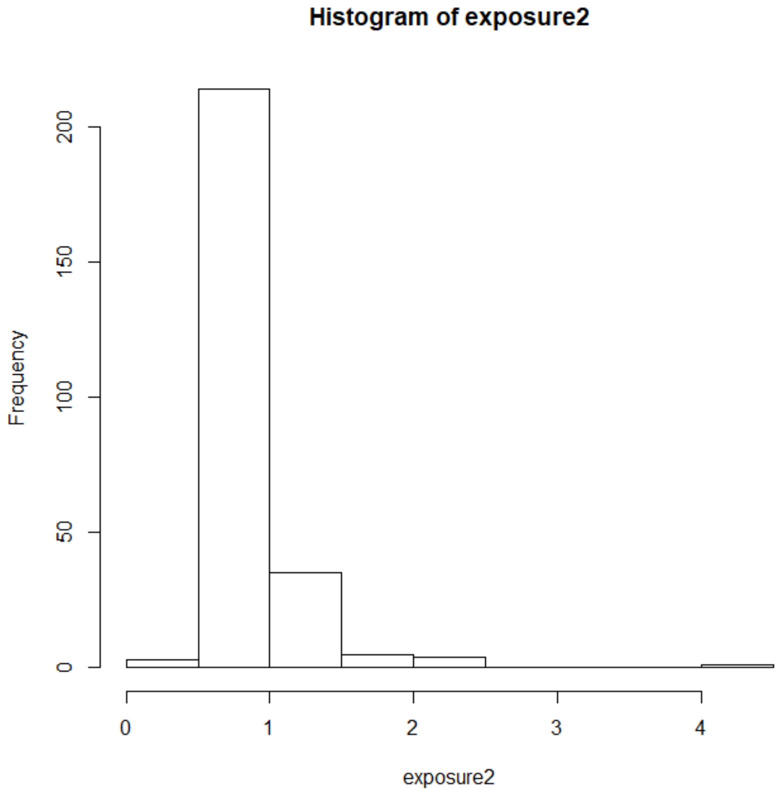
Poisson Regression – Exposure and Offsets

```
> glm.1 <- glm (y ~ roach1 + treatment +
   senior, family=poisson,
   offset=log(exposure2))
> round(cbind(glm.0=coef(glm.0),
   glm.1=coef(glm.1)),4)
```

```
                 glm0       glm1
(Intercept)    3.1360     3.0892
roach1         0.0064     0.0070
treatment     -0.5124    -0.5167
senior        -0.3760    -0.3799
```

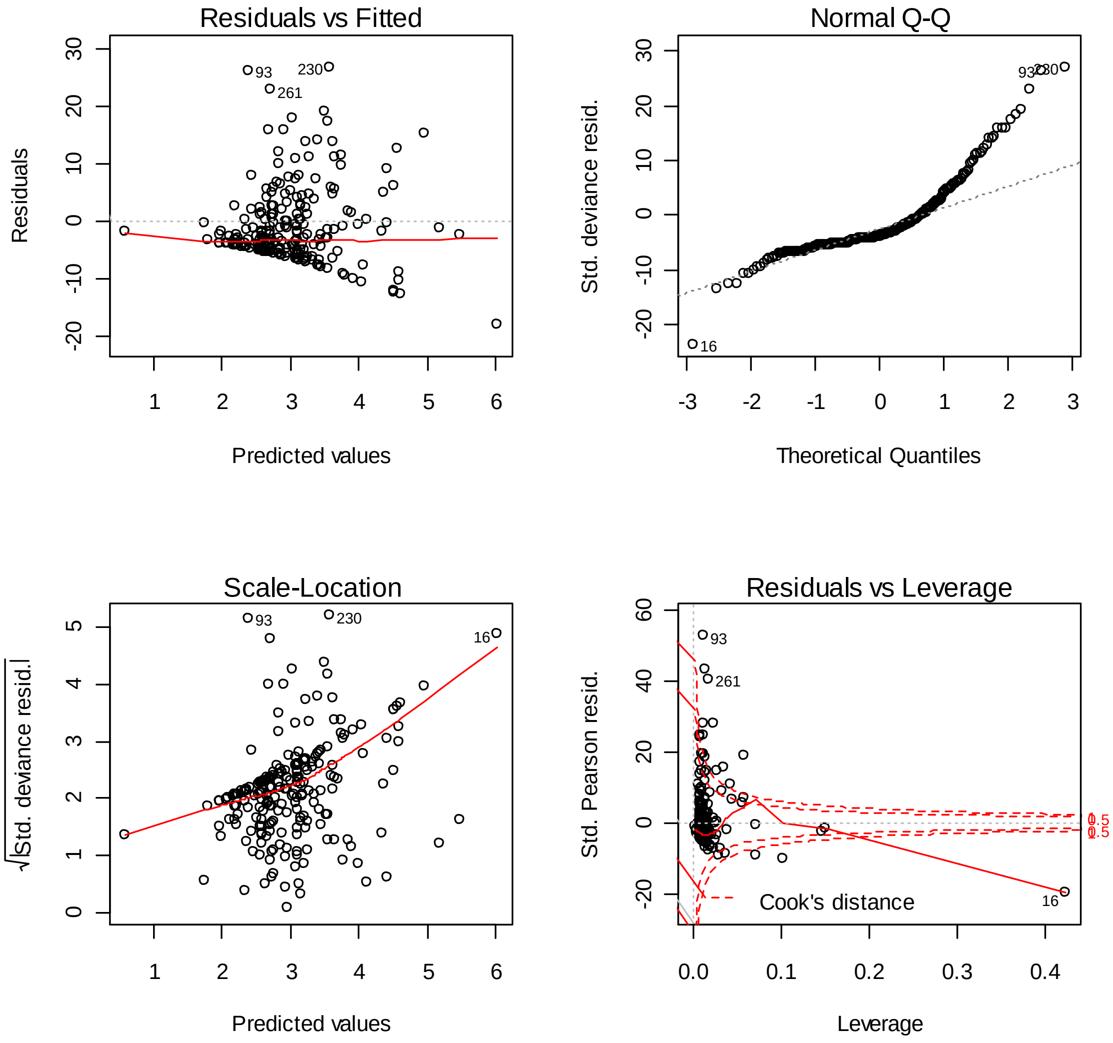
Why didn’t log(exposure2) matter much?

```
> hist(exposure2)
> table(round(exposure2,2))

 0.2  0.4 0.46 0.57  0.6 0.69
   1    1    1    1    5    2
0.77  0.8 0.86 0.91    1 1.03
   3   37    8    2  156    3
1.14 1.29 1.43 1.49 1.57 1.71
  19    4    7    2    2    2
1.86 2.29 2.43 4.29
   1    3    1    1
```



Poisson Regression – Looking at Residuals

```
par(mfrow=c(2,2))
plot(glm.1)
```

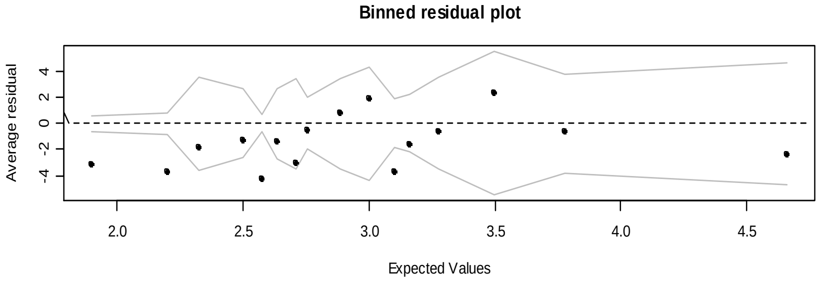


R’s residual plots for glm()’s

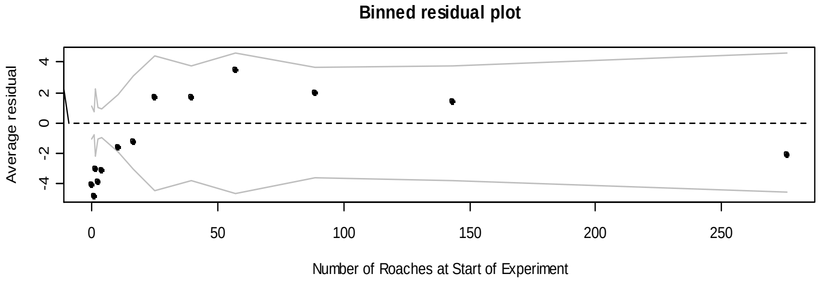
- The std residual plots tend to be useful as long as the individual y_i’s are (theoretically) approximately normal.
  - Binary logistic regression – never really useful
  - Binomial logistic regression: As long as $E[y_i] = n_i p_i \geq 5$ and $E[n_i - y_i] = n_i(1 - p_i) \geq 5$ (CLT!)
  - Poisson regression: As long as $E[y_i] = \lambda_i \geq 10$ (CLT!)
- In the “residuals vs fitted” plot the “fitted values” (x-axis) are $X_i\,\hat{\beta}$’s, not $\widehat{y_i}$’s
  - For the plot on the prev slide, the lowest x-value is $X_i\,\hat{\beta} = 1$; $\mathrm{E[y]} = \hat{\lambda} = e^1 = 2.718 < 10$, so we should proceed with caution
  - Most of the data has $X_i\,\hat{\beta} \geq 2$ or 3, so $\widehat{\lambda_i} = e^{X_i\,\widehat{\beta}}$ is 7 or 20 or more, so most of the data in the plots can be used to assess model fit
  - This doesn’t get rid of issues like over-predicting observed values of $y_i = 0$ in the data (the curved lower boundary in the residual plot)…

Poisson Regression – Looking at Residuals

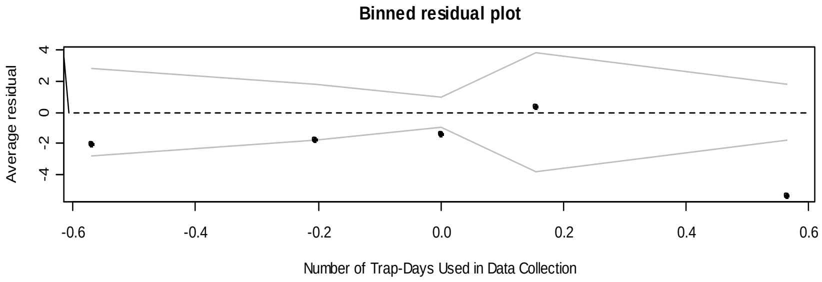
```
par(mfrow=c(3,1))
xvar <- predict(glm.1)
yvar <- resid(glm.1)
binnedplot(xvar,yvar)
```



```
xvar <- roach1
binnedplot(xvar,yvar,xlab
   ="Number of Roaches at
   Start of Experiment")
```



```
xvar <- log(exposure2)
binnedplot(xvar,yvar,xlab
   ="Number of Trap-Days
   Used in Data
   Collection")
```



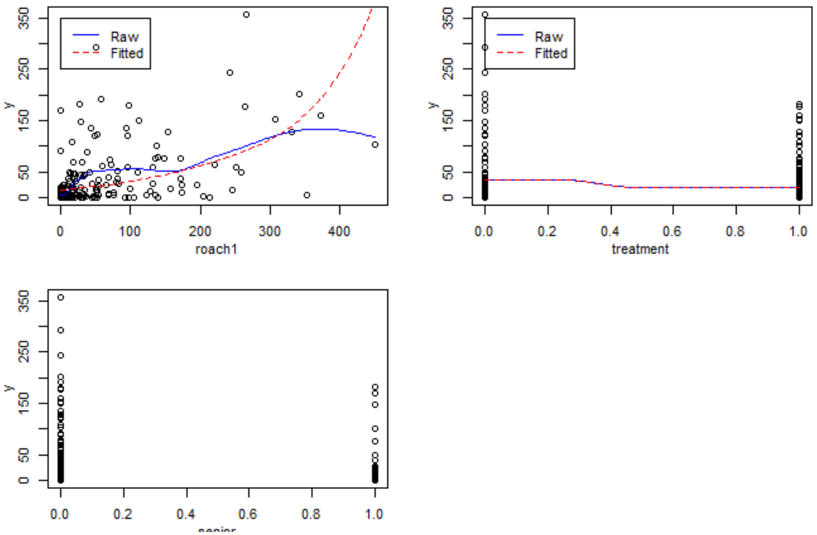
What can we learn from the binned residual plots?

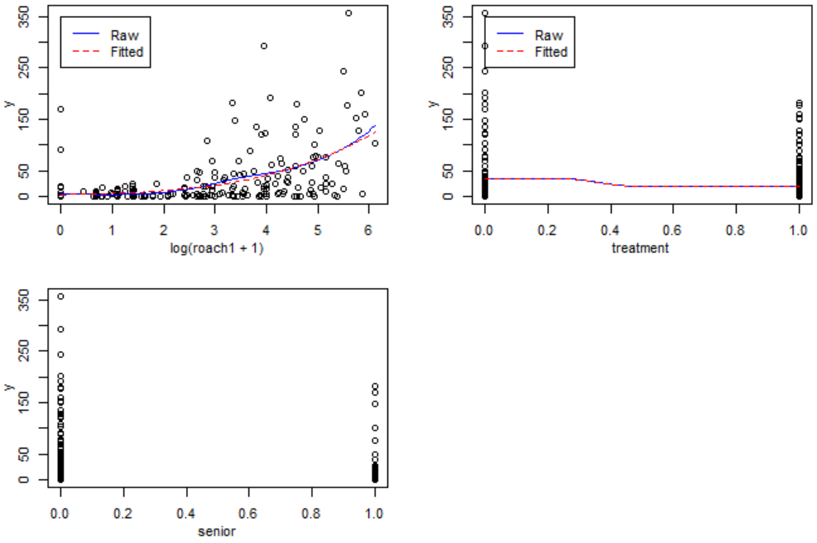
- $y_i - \hat{y}_i < 0$ when $\hat{y}$ is low – we are likely “over-predicting” low values of *y*.
- The variables roach1, exposure2 both seem to have long right tails
  - boxCox() from library(car) suggests logarithms for both variables
  - We are already using log(exposure2) as an offset
  - roach1 has some zeros, so perhaps experiment with log(roach1+1)
- More to do here, but we turn to fit of Poisson Dist.

Let’s see what mmplot tells us…

```
> library(marginalmodelplots)
> glm.2 <- glm (y ~ log(roach1) +
+ treatment + senior, family=poisson,
+ offset=log(exposure2))
> mmplot(glm.1)
> mmplot(glm.2)
```

- Clearly log(roach1+1) cleaned up the mmplot.
- We would see similar improvement in the binned residual plots
  - especially plotting against fitted values or against log(roach1+1)

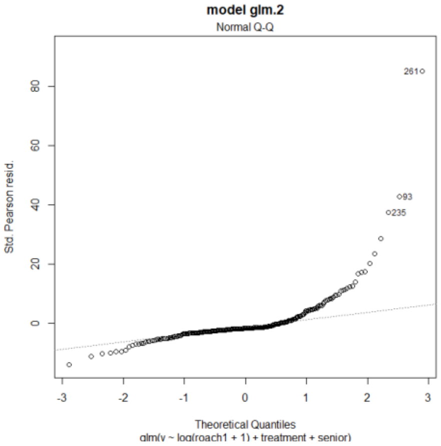


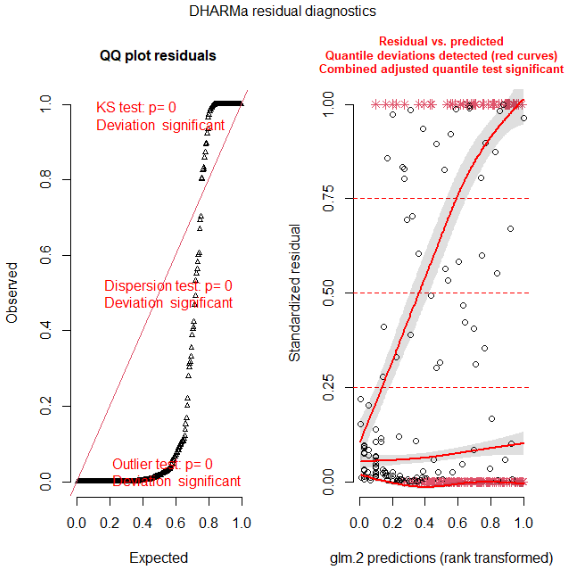


Check distribution of the residuals…

```
> ## Raw residuals
> plot(glm.2,which=2,main="model glm.2")

> ## DHARMa residuals
> library(DHARMa)
> d.resids.glm.2 <-
+ simulateResiduals(glm.2,plot=F)
> plot(d.resids.glm.2, xlab =
+ "glm.2 predictions (rank transformed)")
```

- The S-shape in qq plots suggests overdispersion
  - Easier to see in DHARMa plot
- The resid vs fitted plot suggests variance assumption violated
  - Solid quantile regressions do not line up with dashed guidelines
- Many high and low outliers consistent with overdispersion





Poisson Regression: Testing Overdispersion

- If $y_i \sim$ Poisson(λ_i) then the *Pearson residual*

$$z_i = \frac{y_i - \lambda_i}{\sqrt{\lambda_i}}$$

is approximately normal, so that

$$\sum_{i=1}^{n} z_i^2$$

should follow a χ^2 distribution on n – p – 1 df

Poisson Regression – Testing Lack of Fit

```
> E.y. <-
    predict(glm.2,type="response")
> z <- (y - E.y.)/sqrt(E.y.)

> test.statistic <- sum(z^2)

> n <- length(y)
> pp1 <- length(coef(glm.2))

> pchisq(test.statistic,n-
    pp1,lower.tail=F)

 [1] 0
```

```
> test.statistic

 [1] 17635.29       # this is *huge*!

> n-pp1

 [1] 258

>  test.statistic/(n-pp1)

 [1] 68.35382
```

We found that the residuals are extremely *overdispersed*: the variability of the z’s is about 68 times what it should be!

Poisson Regression - Overdispersion

- We can adjust our inferences for overdispersion by adjusting the standard errors of the coefficients:

```
round(coef(summary(glm.2))[,1:2],2)
##                 Estimate Std. Error
## (Intercept)         1.67       0.04
## log(roach1 + 1)     0.60       0.01
## treatment          -0.60       0.03
## senior             -0.32       0.03
round(coef(summary(glm.2))[,1:2] %*%
  diag(c(1,sqrt(test.statistic/(n-p-1)))),2)
##                     [,1]       [,2]
## (Intercept)         1.67       0.32
## log(roach1 + 1)     0.60       0.07
## treatment          -0.60       0.21
## senior             -0.32       0.28
```

After adjusting, “everything except “senior” housing status retain signif. coefficients…

Poisson Regression - Overdispersion

- We can also get R to estimate the overdispersed poisson regression model directly.

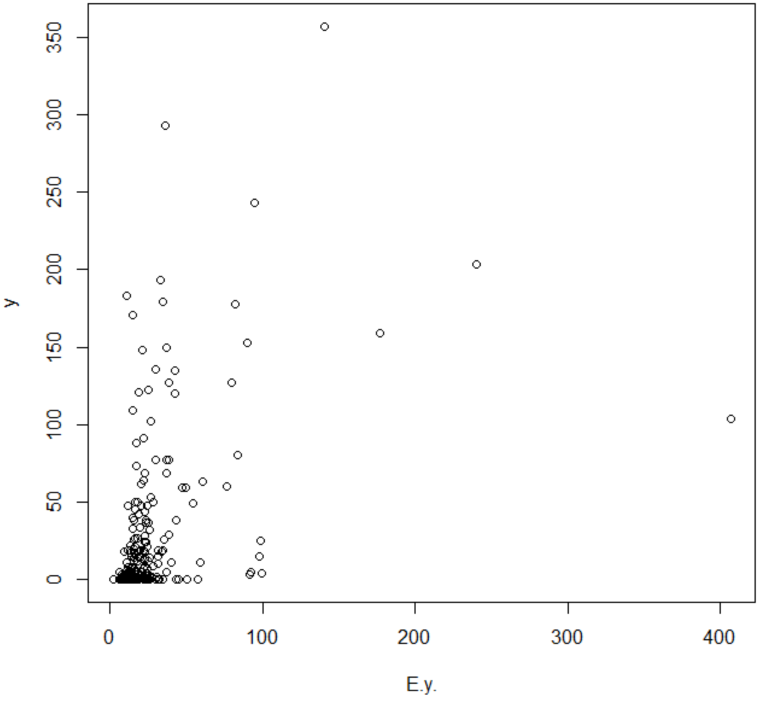
```
> glm.2 <- glm (y ~ log(roach1+1) + treatment + senior,
+ family=quasipoisson, offset=log(exposure2))
> summary(glm.2)

## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.67145    0.31765   5.262 3.00e-07 ***
## log(roach1 + 1)  0.59875    0.06873   8.711 3.68e-16 ***
## treatment       -0.60347    0.20849  -2.894  0.00412 **
## senior          -0.31629    0.27715  -1.141  0.25484
## ---
## Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
##
## (Dispersion parameter for quasipoisson family taken to be 68.35973)
```

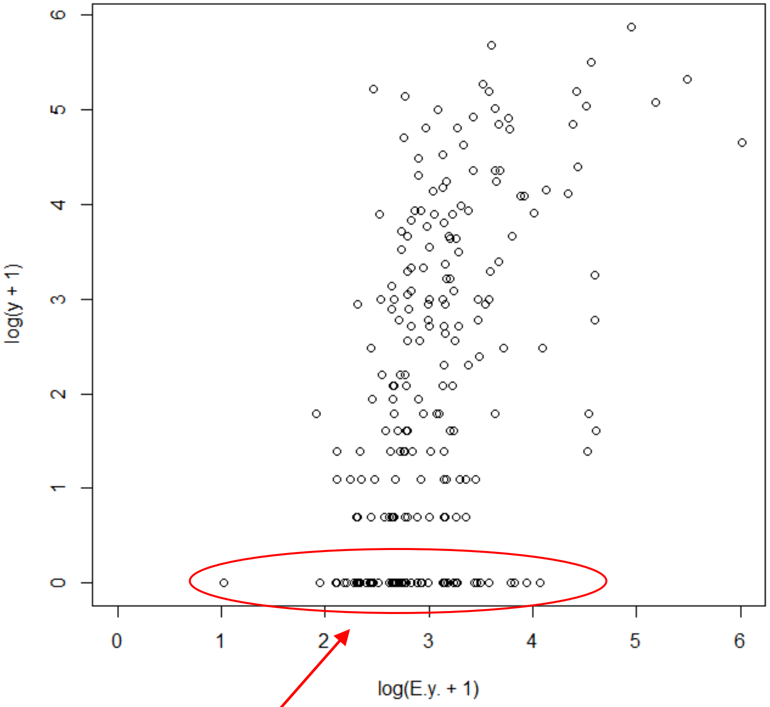
Overdispersion is often an indicator that one or more important predictors is missing from the model!

Poisson Regression – Zero Inflation

- If we explore the data a little more we find that there may be too many zeros for the Poisson model to fit well:

```
> plot(E.y., y)
> plot(log(E.y.+1), log(y+1), xlim=c(0,6))
```

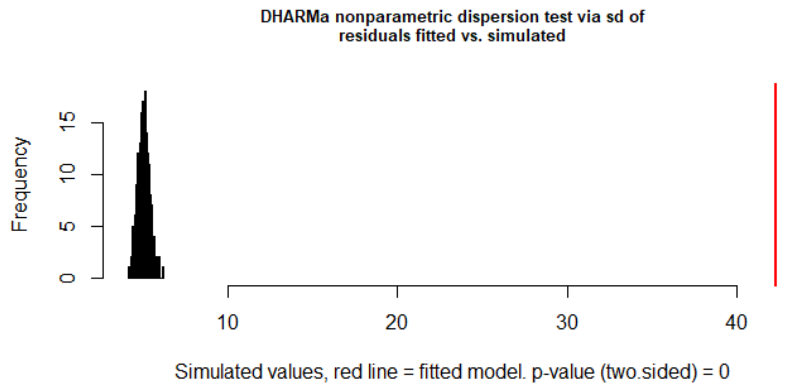


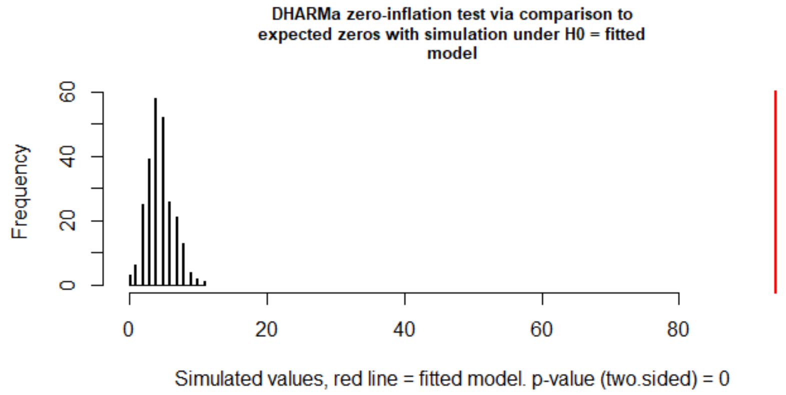


There are too many y=0 values for nonzero E[y]'s!
Note: y=0 iff log(y+1)=0

Aside: DHARMa also has tests for overdispersion and zero-inflation

```
## from library(DHARMa):
> par(mfrow=c(2,1))
> testDispersion(glm.2)
##
## ratioObsSim = 8.3961, p-value < 2.2e-16
## alternative hypothesis: two.sided
> testZeroInflation(glm.2)
## ratioObsSim = 20.632, p-value < 2.2e-16
## alternative hypothesis: two.sided
```


DHARMa nonparametric dispersion test via sd of residuals fitted vs. simulated


DHARMa zero-inflation test via comparison to expected zeros with simulation under H0 = fitted model

- The histogram is distribution under H0: glm.2 correct
- The red line indicates value from the data
- P-value = area to right of red line
- Clearly our data is overdispersed and zero-inflated

Poisson Regression – Zero Inflation

- In cases like this it can also be useful to separately model
  - What distinguishes zero-cockroach buildings from others; and
  - what drives cockroach counts in the buildings that have them
- We will try to combine a logistic regression analysis and a Poisson regression analysis to try to answer these questions
  - This is a crude approximation to the “right” model, which distinguishes “real” zeros from “sampling zeros”

Crude "Model" for Zero Inflation

```
> some.cockroaches <-
   ifelse(y>0, 1, 0)
> zero.fit <-
   glm(some.cockroaches ~
   log(roach1+1) + treatment +
   senior + exposure2,
   family=binomial)
> display(zero.fit)

##                   est      se
## (Intercept)     -0.02    0.61
## log(roach1+1)    0.75    0.11
## treatment       -0.85    0.33
## senior          -0.69    0.33
## exposure2       -0.06    0.50
```

Everything is a significant predictor, except for # of trap-days

```
> glm.3 <- glm (y ~ log(roach1+1) +
    treatment + senior,
    family=poisson,
    offset=log(exposure2),
    subset = (y>0))

> display(glm.3)

##                      est        se
## (Intercept)         2.33      0.04
## log(roach1+1)       0.47      0.01
## treatment          -0.60      0.03
## senior             -0.24      0.03
```

It is somewhat weird to eliminate all the y=0 cases from data for a Poisson model – this is a real weakness of this approach. Better would be to include some y's that are "zero by chance" according to the zero.fit model.

Poisson Regression – Zero Inflation

- *A building with no roaches at the start of the experiment (roach1=0) in the treatment group (treatment=1) that is a seniors' building (senior=1) with 1.5 trap-days (exposure2=1.5) has probability*

  invlogit(-0.02 + (0)*(0.75) + (-0.85)*(1) +

  (-0.69)*(1) + (1.5)*(-0.06)) = 0.16

  of having roaches at the end of the experiment

- *Given that the building does have roaches at the end, the expected number of roaches is*

  exp(log(1.5) + 2.33 + (0)*(0.47) +

  (1)*(-0.60) + (1)*(-0.24)) = 6.65

Modeling multinomial data

- We say $y_i \sim Multinom(\pi_{i1}, \dots, \pi_{iK})$ if
  - $y_i \in \{1, 2, \dots, K\}$
  - $P[y_i = k] = \pi_{ik}$, and $\pi_{i1} + \cdots + \pi_{iK} = 1$
- Many different logits possible
  - Cumulative logits: $\theta_{ik} = \log \frac{P[y_i \leq k]}{1 - P[y_i \leq k]} = \log \frac{\pi_{i1} + \cdots + \pi_{ik}}{\pi_{i(k+1)} + \cdots + \pi_{iK}}$
  - Baseline category logits: $\theta_{ik} = \log \frac{P[y_i = k]}{P[y_i = 1]} = \log \frac{\pi_{ik}}{\pi_{i1}}$
  - Adjacent category logits: $\theta_{ik} = \log \frac{P[y_i = k]}{P[y_i = k-1]} = \log \frac{\pi_{ik}}{\pi_{i(k-1)}}$

A Cumulative Logit Model

- polr from library(MASS):

$$\theta_{ik} = \log \frac{P[y_i \leq k]}{1 - P[y_i \leq k]} = c_k - X_i\beta,$$

$$k = 1, \dots, K - 1$$

β does not depend on k

- Not the only possible cumulative logit model
  - Has a "proportional odds" property that we will see in a few slides
  - Heightens interpretability, costs some flexibility
  - vglm() from library(VGAM) has more flexibility

A Baseline Category Logit Model

- multinom from library(nnet):

$$\theta_{ik} = \log \frac{P[y_i = k]}{P[y_i = 1]} = c_k + X_i\beta_k$$

β can depend on k

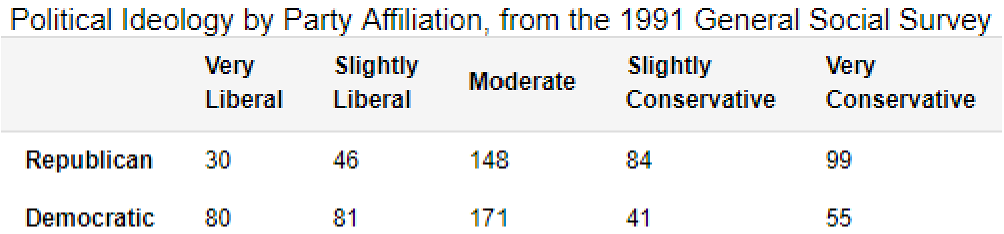
$$k = 1, \dots, K - 1$$

- Very general baseline category logit model
  - Can even use to build “saturated model”
- *Cumulative logit model* often used with *ordered categories*
- *Baseline logit model* often used with *unordered categories*

vglm from library(VGAM) offers both of these models and more… see “multinomial regression R_examples.pdf” in this week’s folder...

Let’s try on a small data set with ordered categories…

- Data from 1991 General Social Survey[1]



Political Ideology by Party Affiliation, from the 1991 General Social Survey

	Very Liberal	Slightly Liberal	Moderate	Slightly Conservative	Very Conservative
Republican	30	46	148	84	99
Democratic	80	81	171	41	55

- Constructing the data set:

```
> party <- factor(rep(c("Rep","Dem"), c(407, 428)),
+                 levels=c("Rep","Dem"))
> rpi <- c(30, 46, 148, 84, 99) # cell counts; sum(rpi)=407
> dpi <- c(80, 81, 171, 41, 55) # cell counts; sum(dpi)=428
> ideology <- c("Very Liberal","Slightly Liberal","Moderate",
+ "Slightly Conservative","Very Conservative")
> pol.ideology <- factor(c(rep(ideology, rpi),
+                          rep(ideology, dpi)), levels = ideology)
> dat <- data.frame(party, pol.ideology)
> table(dat) # check that “dat” matches the table above…
```

This forces R to put the levels in the order we want, instead of alphabetical order

[1] Source: Agresti, A. (1996). *An Introduction to Categorical Data Analysis.* NY: Wiley.

Fitting the model...

```
> library(MASS)
> summary(pom <- polr(pol.ideology ~ party, data=dat))
Call:
polr(formula = pol.ideology ~ party, data = dat)

Coefficients:
           Value Std. Error t value
partyDem -0.9745     0.1292  -7.545

Intercepts:
                                          Value    Std. Error t value
Very Liberal|Slightly Liberal              -2.4690   0.1318   -18.7363
Slightly Liberal|Moderate                  -1.4745   0.1090   -13.5314
Moderate|Slightly Conservative              0.2371   0.0942     2.5165
Slightly Conservative|Very Conservative     1.0695   0.1039    10.2923

Residual Deviance: 2474.985
AIC: 2484.985
```

df = 5 (four c_k's and a β)

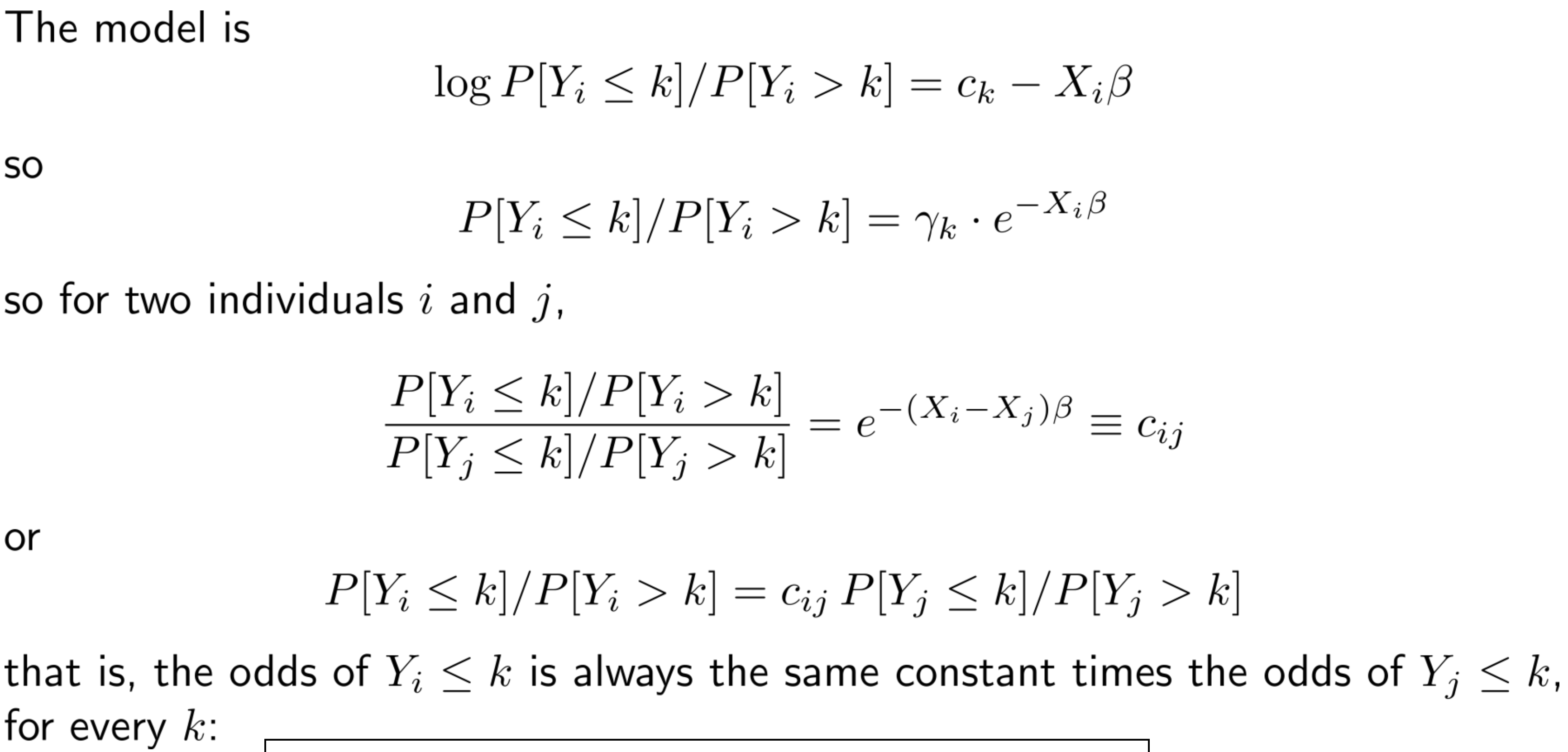
$$\begin{aligned}\log P[Y_i \le k]/P[Y_i > k] &= c_k - (-0.9745)\cdot 1_{\{i \text{ is a Dem}\}}\\ &= c_k + (0.9745)\cdot 1_{\{i \text{ is a Dem}\}}\end{aligned}$$

5 categories, 4 boundaries

Predictions…

```
> predict(pom,newdata = data.frame(party="Dem"),type="probs")
         Very Liberal     Slightly Liberal             Moderate 
            0.1832505            0.1942837            0.3930552 
Slightly Conservative    Very Conservative 
            0.1147559            0.1146547 

> predict(pom,newdata = data.frame(party="Rep"),type="probs")
         Very Liberal     Slightly Liberal             Moderate 
           0.07806044           0.10819225           0.37275214 
Slightly Conservative    Very Conservative 
           0.18550357           0.25549160 

> round(rbind(Dem=predict(pom,newdata = data.frame(party="Dem"),type="probs"),
+ Rep=predict(pom,newdata = data.frame(party="Rep"),type="probs")),2)

    Very Liberal Slightly Liberal Moderate Slightly Conservative Very Conservative
Dem         0.18             0.19     0.39                  0.11              0.11
Rep         0.08             0.11     0.37                  0.19              0.26

> # compare with raw estimates...
> names(dpi) <- ideology; round(rbind(Dem=dpi/sum(dpi),Rep=rpi/sum(rpi)),2)
    Very Liberal Slightly Liberal Moderate Slightly Conservative Very Conservative
Dem         0.19             0.19     0.40                  0.10              0.13
Rep         0.07             0.11     0.36                  0.21              0.24
```

Aside… What does "polr" mean?



The model is

$$\log P[Y_i \le k]/P[Y_i > k] = c_k - X_i\beta$$

so

$$P[Y_i \le k]/P[Y_i > k] = \gamma_k \cdot e^{-X_i\beta}$$

so for two individuals i and j,

$$\frac{P[Y_i \le k]/P[Y_i > k]}{P[Y_j \le k]/P[Y_j > k]} = e^{-(X_i - X_j)\beta} \equiv c_{ij}$$

or

$$P[Y_i \le k]/P[Y_i > k] = c_{ij}\, P[Y_j \le k]/P[Y_j > k]$$

that is, the odds of $Y_i \le k$ is always the same constant times the odds of $Y_j \le k$, for every k:

The cumulative odds for i are proportional to the cumulative odds for j, uniformly in the cutoff k

This works because β doesn't depend on k

- polr = "proportional odds logistic regression"

Is the polr model valid for this data?

- Compare with the “baseline category logit” model that has a full 8 parameters, vs. 5 from polr().

df = 8 (four c_k’s, & four β’s)

```
> library(nnet)
> summary(mlm <- multinom(pol.ideology ~ party, data=dat))
Call:
multinom(formula = pol.ideology ~ party, data = dat)
```

$$\begin{aligned} \log P[Y_i = k|X_i]/P[Y_i = 1|X_i] &= c_k + X_i\beta_k \\ & c_k + \beta_k 1_{\{i \text{ is a Dem}\}} \end{aligned}$$

```
Coefficients:
                      (Intercept)   partyDem
Slightly Liberal        0.4274512 -0.4150328
Moderate                1.5960039 -0.8363535
Slightly Conservative   1.0296159 -1.6980658
Very Conservative       1.1939108 -1.5685961

Std. Errors:
                      (Intercept)  partyDem
Slightly Liberal        0.2346741 0.2826974
Moderate                0.2002246 0.2417393
Slightly Conservative   0.2126919 0.2865802
Very Conservative       0.2084087 0.2722426

Residual Deviance: 2471.297
AIC: 2487.297
```

```
> M0 <- logLik(pom)
> M1 <- logLik(mlm)
> (G <- -2*(M0[1] - M1[1]))
[1] 3.687678
> pchisq(G,8-5,lower.tail = FALSE)
[1] 0.2972241
```

⟹ Polr model (M0) ok…

Summary

- Linear Regression, Logistic Regression
- Generalized Linear Models (GLM)
- Quick Taste of Ordered Logistic Regression
- Poisson Regression
  - Exposure and Offsets
  - Overdispersion
  - Zero-inflation
- Quick Taste of Ordered Logistic Regression
  - Many ways to model multinomial data
  - Example