POLSCI 9592

Lecture 7: Count Data Models

Dave Armstrong

Goals for This Session

1. Develop Count models - poisson, negative binomial, binomial
2. Effects and Effect Displays
3. Model Fit and Evaluation
4. Discuss Overdispersion
5. Zero-inflation and hurdle models

Count outcomes

- Non-negative, integer values (i.e., the number of times the outcome happened).
- May also have a variable that measures potential count (i.e., the number of times the outcome could have happened.)

Under the right circumstances, the linear model could be applied to these data.

- under the wrong circumstances, the linear model (i.e., the wrong distributional assumptions) can induce inefficiency, inconsistency and bias.

Modeling counts

- We want to know, what is the probability that y takes on the count we observe given some variables $\mathbf{X}$.
- To know anything about the probability of something happening, we need to know its probability distribution.

The simplest model for count outcomes is the Poisson model. The PMF (discrete analog to PDF) of the poisson distribution is:

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

where the only parameter in the model is μ, the mean (sometimes called the rate).

Properties of Poisson Distribution

1. As μ increases, the bulk of the distribution moves to the right, with less probability given to zero.
2. $var(y) = \mu$, the variance and the mean are the same (called equidispersion). We will talk about models for overdispersed count data later.
3. As μ increases $Pr(y = 0)$ decreases, so often times, more zeros are observed than predicted
4. As μ increases, the poisson distribution becomes approximately normal.

Poisson Regression Model

The Poisson Regression model parameterizes μ_i from the poisson PDF in the following way:

$$\mu_i = \exp(\mathbf{x}_i \beta)$$

So, then:

$$\begin{aligned} Pr(y_i|\mathbf{x}_i) &= \frac{\exp(-\mu_i)\mu_i^{y_i}}{y_i!} \\ &= \frac{\exp(-\exp(\mathbf{x}_i\beta))\exp(\mathbf{x}_i\beta)^{y_i}}{y_i!} \end{aligned}$$

Likelihood Function

$$L(\mu) = \prod_{i=1}^{N} \frac{\exp(-\exp(\mathbf{x}_i\beta))\exp(\mathbf{x}_i\beta)^{y_i}}{y_i!}$$

and the log-likelihood function is:

$$lnL(\mu) = -n\exp(\mathbf{x}_i\beta) + \left(\sum_{i=1}^{n} y_i\right) ln(\exp(\mathbf{x}_i\beta)) - \sum_{i=1}^{n} ln(y_i!)$$

or ...

$$LL = \sum_{i=1}^{n} log\left(f\left(y_i, e^{\mathbf{x}_i\beta}\right)\right)$$

where $f(y, \mu)$ is the poisson PDF of y evaluated at μ.

Example

```r
library(rio)
dat <- import("data/count.dta")
dat <- dat %>%
  mutate(across(c("unemployed", "religimp"), factorize))
mod <- glm(volorgs ~ age + educ + unemployed + hhincome_num +
             leftright + numkids + religimp,
           data=dat,
           family=poisson)
```

```r
summary(mod)
```

```
##
## Call:
## glm(formula = volorgs ~ age + educ + unemployed + hhincome_num +
##     leftright + numkids + religimp, family = poisson, data = dat)
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.052300   0.308804  -9.884  < 2e-16 ***
## age                    0.010045   0.002528   3.973 7.08e-05 ***
## educ                   0.170059   0.018214   9.337  < 2e-16 ***
## unemployedNot Employed -1.230194   0.292337  -4.208 2.57e-05 ***
## hhincome_num           0.021967   0.007534   2.916  0.00355 **
## leftright             -0.044094   0.014646  -3.011  0.00261 **
## numkids                0.045372   0.031955   1.420  0.15566
## religimpimportant      0.074319   0.082740   0.898  0.36906
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1613.1  on 885  degrees of freedom
## Residual deviance: 1356.1  on 878  degrees of freedom
##   (707 observations deleted due to missingness)
## AIC: 2438.9
##
## Number of Fisher Scoring iterations: 6
```

Interpretation

Let's imagine changing x_k from some specified value to that value plus δ, holding all of the other x variables constant at some value. Then,

$$\frac{E(y|\mathbf{x}, x_k + \delta)}{E(y|\mathbf{x}, x_k)} = \exp(\beta_k \delta)$$

where $E(y|\mathbf{x})$ is simply $\mu = \exp(\mathbf{x}_i \beta)$. So, the count will increase by a factor of $\exp(\beta_k \delta)$ for a δ unit increase in variable x_k.

- We expect the number of voluntary organizations to increase by a factor of 1.185 for every additional year of formal education.
- If we predicted 3 voluntary organizations for someone with 12 years of education, we would expect someone with 13 years of education to join $3 \times 1.185 = 3.56$ voluntary organizations.

Illustration

```
tmpdf <- data.frame(
    age = 45,
    educ = c(11,12,16,17),
    unemployed = factor(0, levels=c(0,1), labels=levels(dat$unemployed)),
    hhincome_num = 15,
    leftright = 5,
    numkids = 0,
    religimp = factor(1, levels=0:1, labels=levels(dat$religimp)))
preds <- predict(mod, newdata=tmpdf, type="response")
preds
```

```
##         1         2         3         4
## 0.5791113 0.6864639 1.3553155 1.6065568
```

```
preds[2]/preds[1]
```

```
##        2
## 1.185375
```

```
preds[4]/preds[3]
```

```
##        4
## 1.185375
```

Interpretation II

We can also figure out by how many percent your count will increase for a δ unit change in x_k:

$$100 \times \frac{E(y|\mathbf{x}, x_k + \delta) - E(y|\mathbf{x}, x_k)}{E(y|\mathbf{x}, x_k)} = 100 \times \{\exp(\beta_k \delta) - 1\}$$

- We expect the number of voluntary organizations to increase by 18.54% for every additional year of formal education.

```
100*(preds[2] - preds[1])/preds[1]
```

```
##        2
## 18.53748
```

```
100*(preds[4] - preds[3])/preds[3]
```

```
##        4
## 18.53748
```

Discrete Changes (MERs)

```r
library(marginaleffects)
comparisons(mod,
            newdata = "median",
            variables=list(
              age = "2sd",
              educ = c(12,16),
              unemployed= "minmax",
              hhincome_num = "2sd",
              leftright = c(2,8),
              numkids =c(0,2),
              religimp = "minmax")) %>%
  select(1:7)
```

```
##
##         Term                   Contrast Estimate Std. Error      z Pr(>|z|)
##   age          (x + sd) - (x - sd)        0.3025     0.0726  4.164  < 0.001
##   educ         16 - 12                    0.6755     0.0769  8.786  < 0.001
##   hhincome_num (x + sd) - (x - sd)        0.2320     0.0795  2.919  0.00351
##   leftright    8 - 2                     -0.2701     0.0954 -2.831  0.00464
##   numkids      2 - 0                      0.0925     0.0654  1.415  0.15702
##   religimp     important - not important  0.0698     0.0764  0.914  0.36097
##   unemployed   Not Employed - Employed   -0.6895     0.0942 -7.321  < 0.001
##
## Columns: rowid, term, contrast, estimate, std.error, statistic, p.value
```

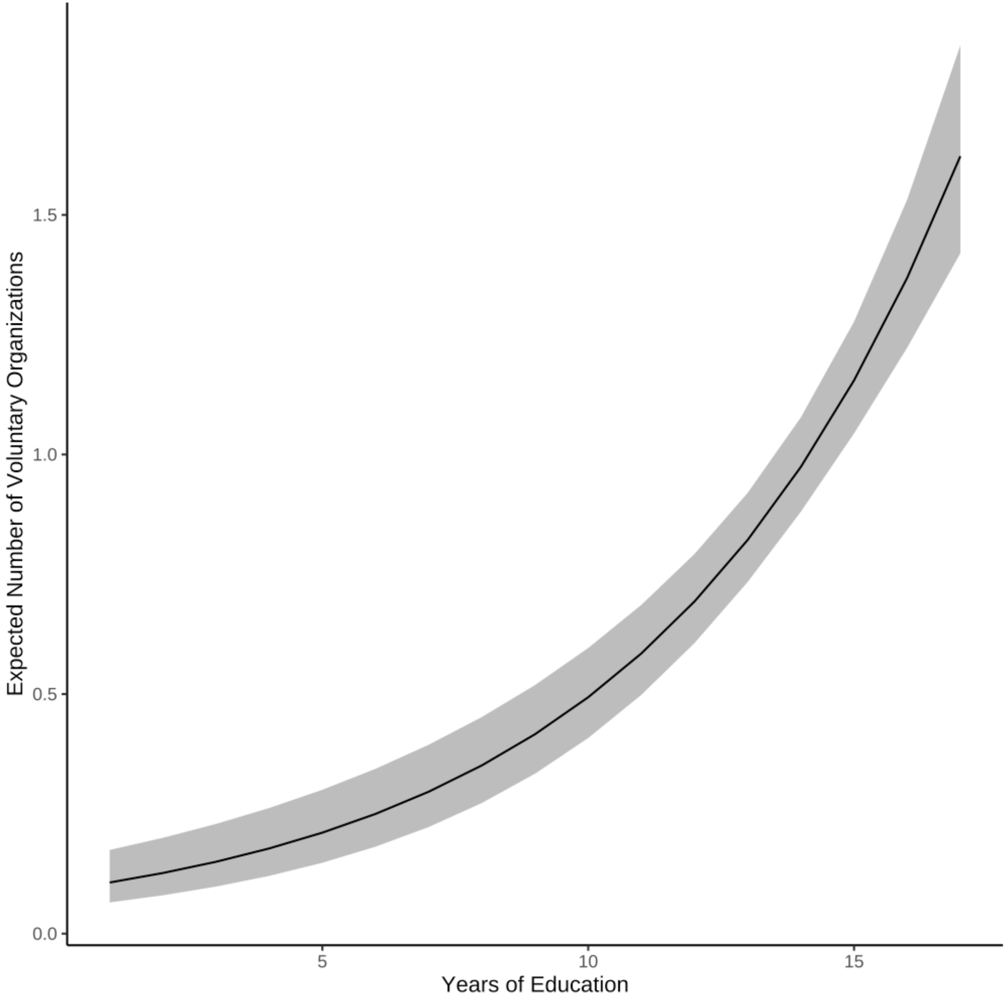
Discrete Changes (AMEs)

```
avg_comparisons(mod,
           variables=list(
             age = "2sd",
             educ = c(12,16),
             unemployed= "minmax",
             hhincome_num = "2sd",
             leftright = c(2,8),
             numkids =c(0,2),
             religimp = "minmax")) %>%
  select(1:6)
```

```
##
##          Term                              Contrast Estimate Std. Error      z
##   age          mean(x + sd) - mean(x - sd)            0.3065     0.0775  3.958
##   educ         mean(16) - mean(12)                    0.6650     0.0720  9.235
##   hhincome_num mean(x + sd) - mean(x - sd)            0.2354     0.0788  2.988
##   leftright    mean(8) - mean(2)                     -0.2760     0.0958 -2.882
##   numkids      mean(2) - mean(0)                      0.0917     0.0652  1.406
##   religimp     mean(important) - mean(not important)  0.0730     0.0798  0.915
##   unemployed   mean(Not Employed) - mean(Employed)   -0.7318     0.0946 -7.739
##   Pr(>|z|)
##    < 0.001
##    < 0.001
##    0.00281
##    0.00395
##    0.15964
##    0.36017
##    < 0.001
##
## Columns: term, contrast, estimate, std.error, statistic, p.value
```

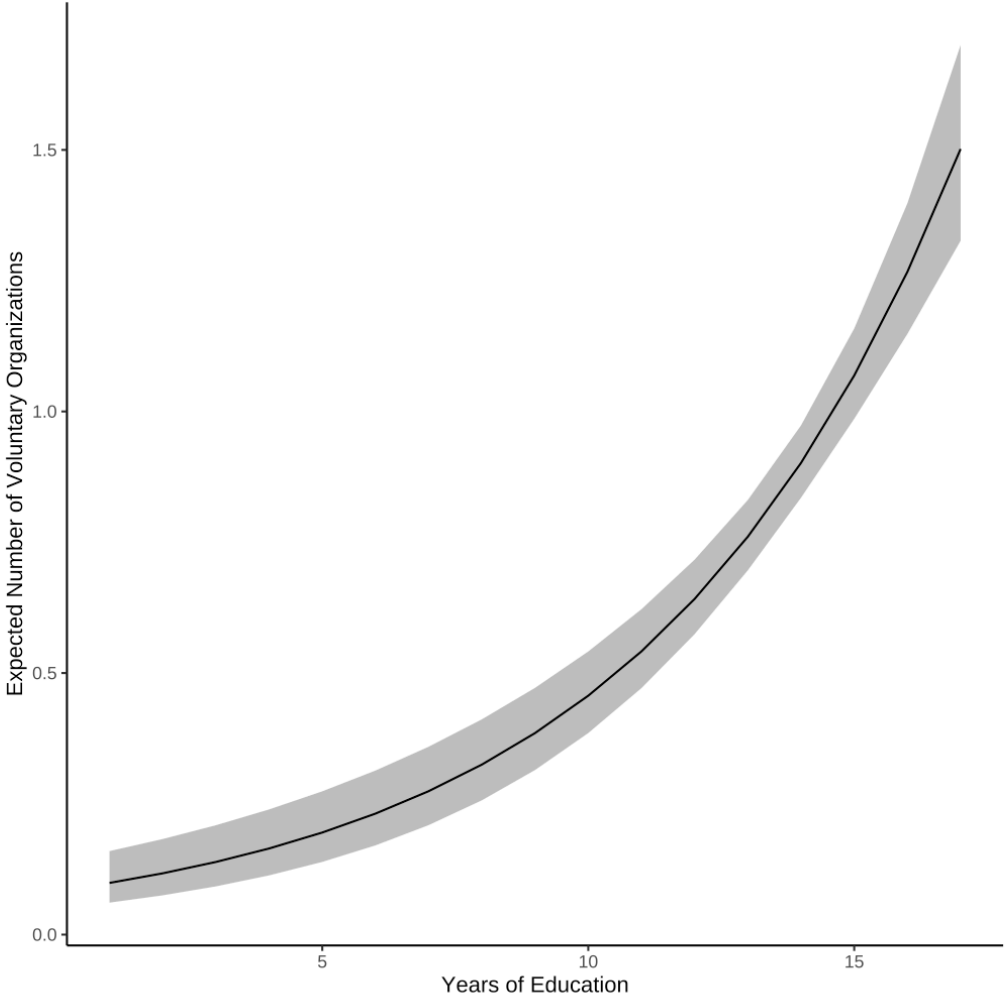
Effects Plot

```
preds <- predictions(mod,
            newdata="median",
            variables = list(educ = 1:17))
ggplot(preds,
       aes(x=educ, y=estimate,
           ymin=conf.low,
           ymax=conf.high)) +
  geom_ribbon(fill="gray75") +
  geom_line() +
  theme_classic() +
  labs(x="Years of Education", y="Expected Number of Voluntary Or
```



Effects Plot (AME)

```
apreds <- avg_predictions(mod,
            variables = list(educ = 1:17))
ggplot(apreds,
       aes(x=educ, y=estimate,
           ymin=conf.low,
           ymax=conf.high)) +
  geom_ribbon(fill="gray75") +
  geom_line() +
  theme_classic() +
  labs(x="Years of Education", y="Expected Number of Voluntary Or
```



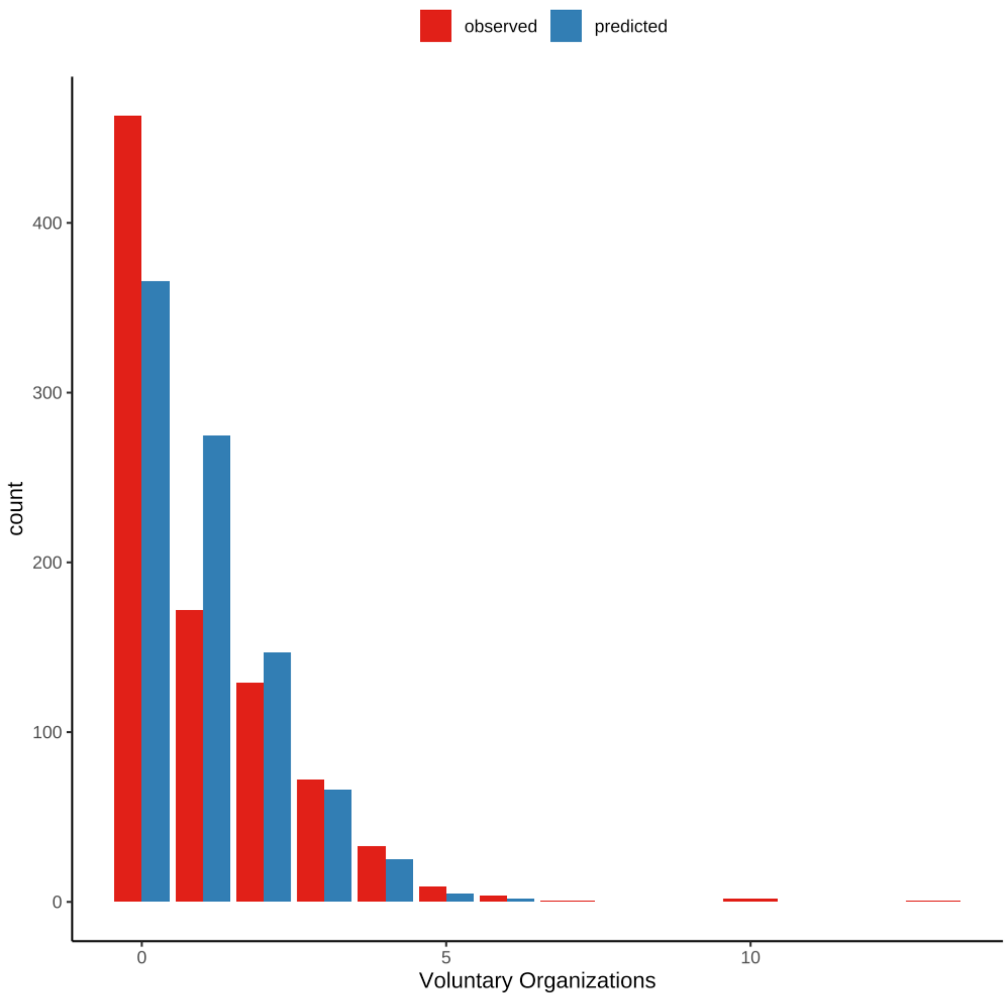
Model Fit

```
poisfit(mod)
```

```
##                              Estimate p-value
## GOF (Pearson)                1552.688 0.000
## GOF (Deviance)               1356.087 0.000
## ML R2                        0.252    NA
## McFadden R2                  0.096    NA
## McFadden R2 (Adj)            0.090    NA
## Cragg-Uhler(Nagelkerke) R2 0.265    NA
```

Predicted vs. Actual

```
yhat <- predict(mod, type="response")
draw <- rpois(length(yhat), yhat)
fitdat <- tibble::tibble(
  val = c(model.response(model.frame(mod)), draw),
  type = factor(rep(c("observed", "predicted"), each=length(draw)
)
ggplot(fitdat,
       aes(x=val, fill=type)) +
  geom_bar(position="dodge") +
  theme_classic() +
  scale_fill_brewer(palette="Set1") +
  labs(x="Voluntary Organizations", fill="") +
  theme(legend.position = "top")
```



Negative Binomial model

The negative binomial model is used when we have an overdispersed variable.

- Overdispersion is when variance is *greater than* the mean.
- Overdispersion is an attribute of the outcome variable *and* a model. Data are not themselves overdispersed, independent of a particular model.
- It is possible to model away some of the overdispersion, but usually only if the variance is 2 or 3 times the mean.

The NBRM adds an error term to the linear predictor that has expectation 0 and is assumed uncorrelated with the remainder of the X variables.

$$\begin{aligned}\mu &= \exp(\mathbf{X}\beta + \varepsilon) \\ &= \exp(\mathbf{X}\beta)\exp(\varepsilon) \\ &= \exp(\mathbf{X}\beta)\delta\end{aligned}$$

NBRM Example

```
mod2 <- MASS::glm.nb(volorgs ~ age + educ + unemployed +
                 hhincome_num + leftright + numkids +
                 religimp,
               data=dat)
```

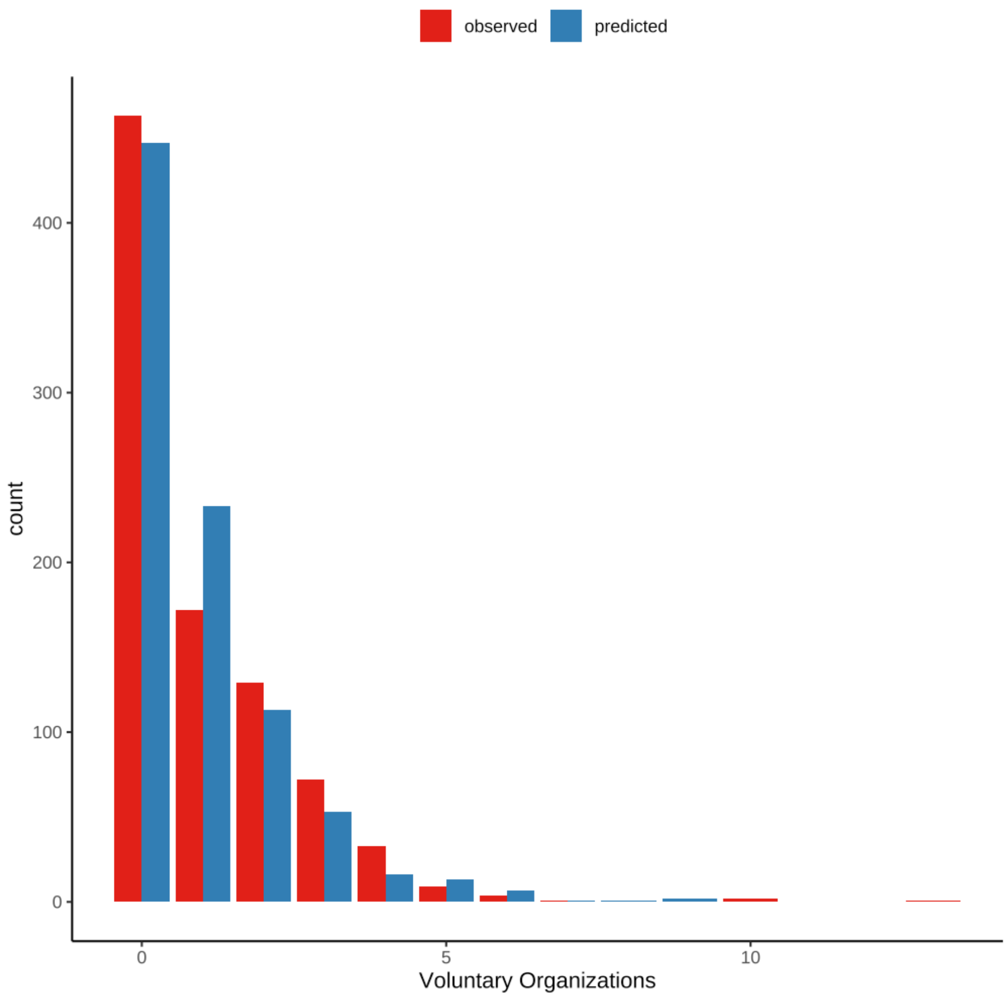
- The `Theta` term here is the overdispersion parameter.

```
summary(mod2)
```

```
##
## Call:
## MASS::glm.nb(formula = volorgs ~ age + educ + unemployed + hhincome_nu
##     leftright + numkids + religimp, data = dat, init.theta = 1.5599164
##     link = log)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -2.972767   0.396719  -7.493 6.71e-14 ***
## age                     0.009092   0.003299   2.756  0.00585 **
## educ                    0.168377   0.023401   7.195 6.23e-13 ***
## unemployedNot Employed -1.251872   0.319325  -3.920 8.84e-05 ***
## hhincome_num            0.022505   0.009686   2.323  0.02016 *
## leftright              -0.047494   0.019456  -2.441  0.01465 *
## numkids                 0.042735   0.041668   1.026  0.30508
## religimpimportant       0.078662   0.110603   0.711  0.47695
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.5599) family taken to be
##
##     Null deviance: 1046.47  on 885  degrees of freedom
## Residual deviance:  886.68  on 878  degrees of freedom
##   (707 observations deleted due to missingness)
## AIC: 2333
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.560
##           Std. Err.:  0.236
##
##  2 x log-likelihood:  -2314.991
```

Predicted vs. Actual

```
yhat <- predict(mod2, type="response")
draw <- MASS::rnegbin(length(yhat), yhat, theta=mod2$theta)
fitdat <- tibble::tibble(
  val = c(model.response(model.frame(mod)), draw),
  type = factor(rep(c("observed", "predicted"), each=length(draw)
)
ggplot(fitdat,
       aes(x=val, fill=type)) +
  geom_bar(position="dodge") +
  theme_classic() +
  scale_fill_brewer(palette="Set1") +
  labs(x="Voluntary Organizations", fill="") +
  theme(legend.position = "top")
```



Hurdles and Zero-inflation

Sometimes, there are even more zeros than we would expect taking account of overdispersion. We can deal with this in two ways:

- Hurdle - assumes that one process governs zero vs non-zero and then a separate count process (poisson or NB) governs the positive counts.
- Zero-inflation - assumes that the zeros can come from one of two processes - a hurdle-like process that separates zeros from non-zeros and zero from the count part of the model.

Hurdle Model

The hurdle model estimates the probability of zero as a separate process from the non-zero counts.

$$
\begin{aligned}
LL_i = {} & I(y_i = 0)\log(1 - F_1(\mathbf{z}_i\gamma)) + I(y_i = 1) log(F_1(\mathbf{z}_i\gamma)) \\
& + [log(f_2(y_i, \mathbf{x}_i\beta)) - log(F_2(\mathbf{x}_i\beta))]
\end{aligned}
$$

Let's break it down:

- $I(y_i = 0)\log(1 - F_1(\mathbf{z}_i\gamma))$ is the log of the probability that $y_i = 0$ given a logistic regression of y on $\mathbf{Z}$ for the zeros.
- $I(y_i = 1) log(F_1(\mathbf{z}_i\gamma))$ is the log of the probability that $y_i \neq 0$ given a logistic regression of y on $\mathbf{Z}$ for the non-zeros.
- $[log(f_2(y_i, \mathbf{x}_i\beta)) - log(F_2(\mathbf{x}_i\beta))]$ is the log of the probability that y takes on its observed value in the *truncated* poisson (or NB) pdf for the non-zeros.

Estimation in R

By assumption, the hurdle and count models are specified the same way.

```
library(pscl)
mod <- hurdle(volorgs ~ age + educ + unemployed + hhincome_num +
                leftright + numkids + religimp,
              data=dat,
              dist="negbin",
              zero.dist = "binomial")
```

Result

```
Count model coefficients (truncated negbin with log link):
                        Estimate Std. Error z value Pr(>|z|)    
(Intercept)            -1.664714   0.465743  -3.574 0.000351 ***
age                     0.002547   0.003665   0.695 0.487053    
educ                    0.121929   0.027608   4.416 1.00e-05 ***
unemployedNot Employed  0.356697   0.406697   0.877 0.380455    
hhincome_num            0.024343   0.011375   2.140 0.032356 *  
leftright              -0.030925   0.020674  -1.496 0.134687    
numkids                -0.031557   0.047376  -0.666 0.505344    
religimpimportant      -0.017834   0.117368  -0.152 0.879229    
Log(theta)              1.970148   0.494974   3.980 6.88e-05 ***
```

```
Zero hurdle model coefficients (binomial with logit link):
                        Estimate Std. Error z value Pr(>|z|)    
(Intercept)            -3.900897   0.630429  -6.188 6.11e-10 ***
age                     0.017152   0.005247   3.269  0.00108 ** 
educ                    0.213338   0.037019   5.763 8.26e-09 ***
unemployedNot Employed -2.034148   0.483148  -4.210 2.55e-05 ***
hhincome_num            0.020330   0.014975   1.358  0.17457    
leftright              -0.065082   0.031646  -2.057  0.03973 *  
numkids                 0.120404   0.066174   1.820  0.06883 .  
religimpimportant       0.198351   0.181361   1.094  0.27410    
```

Effects

With the `marginaleffects` package, you can specify that you want effects on the scale of:

1. $Pr(y_i \neq 0)$ with `"zero"`
2. $\hat{y}_i^{(c)}$, the predicted count from the truncated count part of the equation with `"count"`
3. $Pr(y_i = j)$ for $j = \{0, \ldots, J\}$, the predicted probability of being in each count with `"prob"`
4. $\hat{y}_i$, the predicted count multiplied by the probability of getting non-zero counts with `"response"`

Effect of Unemployment

```r
avg_predictions(mod, variables="unemployed", type="zero") %>%
  as_tibble() %>%
  select(unemployed, estimate, conf.low, conf.high)
```

```
## # A tibble: 2 × 4
##   unemployed   estimate conf.low conf.high
##   <fct>           <dbl>    <dbl>     <dbl>
## 1 Employed        0.702   0.633      0.772
## 2 Not Employed    0.155   0.0272     0.283
```

```r
avg_predictions(mod, variables="unemployed", type="prob") %>% hea
  as_tibble() %>%
  select(unemployed, estimate, conf.low, conf.high)
```

```
## # A tibble: 2 × 4
##   unemployed   estimate conf.low conf.high
##   <fct>           <dbl>    <dbl>     <dbl>
## 1 Employed        0.503    0.470     0.535
## 2 Not Employed    0.870    0.769     0.972
```

```r
avg_predictions(mod, variables="unemployed", type="count") %>%
  as_tibble() %>%
  select(unemployed, estimate, conf.low, conf.high)
```

```
## # A tibble: 2 × 4
##   unemployed   estimate conf.low conf.high
##   <fct>           <dbl>    <dbl>     <dbl>
## 1 Employed         1.42    1.25       1.59
## 2 Not Employed     2.03    0.426      3.63
```

```r
avg_predictions(mod, variables="unemployed", type="response")  %>
  as_tibble() %>%
  select(unemployed, estimate, conf.low, conf.high)
```

```
## # A tibble: 2 × 4
##   unemployed   estimate conf.low conf.high
##   <fct>           <dbl>    <dbl>     <dbl>
## 1 Employed        1.04    0.939      1.13
## 2 Not Employed    0.359   0.0103     0.707
```

Zero-Inflated Models

Zero-inflated models are the same as hurdles, but the zeros can come from both the binomial and the count processes.

```
mod <- zeroinfl(volorgs ~ age + educ + unemployed + hhincome_num +
             leftright + numkids + religimp,
           data=dat,
           dist="negbin",
           zero.dist = "binomial")
```

Result

```
Count model coefficients (negbin with log link):
                        Estimate Std. Error z value Pr(>|z|)    
(Intercept)            -1.688558   0.451092  -3.743 0.000182 ***
age                     0.002845   0.003523   0.808 0.419317    
educ                    0.123809   0.026611   4.653 3.28e-06 ***
unemployedNot Employed  0.312325   0.405264   0.771 0.440902    
hhincome_num            0.024697   0.011396   2.167 0.030220 *  
leftright              -0.037220   0.020046  -1.857 0.063355 .  
numkids                -0.030886   0.045406  -0.680 0.496364    
religimpimportant       0.002097   0.117832   0.018 0.985803    
Log(theta)              2.017537   0.500049   4.035 5.47e-05 ***
```

```
Zero-inflation model coefficients (binomial with logit link):
                        Estimate Std. Error z value Pr(>|z|)    
(Intercept)             2.425815   1.142650   2.123   0.0338 *  
age                    -0.022918   0.009753  -2.350   0.0188 *  
educ                   -0.157369   0.068972  -2.282   0.0225 *  
unemployedNot Employed  2.705893   0.571393   4.736 2.18e-06 ***
hhincome_num            0.004193   0.030855   0.136   0.8919    
leftright               0.037771   0.057988   0.651   0.5148    
numkids                -0.255842   0.149640  -1.710   0.0873 .  
religimpimportant      -0.269086   0.325316  -0.827   0.4082    
```

The zero equation is interpreted differently here.

- Hurdle - DV in hurdle equation is 1 if you are in the count part and 0 if you're in the always zero part.
- Zero-inflated - DV in zero inflation is 1 for observations that are 0 and 0 for observations in the count part of the model.

Effect of Unemployment

```r
avg_predictions(mod, variables="unemployed", type="zero") %>%
  as_tibble() %>%
  select(unemployed, estimate, conf.low, conf.high)
```

```
## # A tibble: 2 × 4
##   unemployed   estimate conf.low conf.high
##   <fct>           <dbl>    <dbl>     <dbl>
## 1 Employed        0.297    0.215     0.380
## 2 Not Employed    0.846    0.713     0.979
```

```r
avg_predictions(mod, variables="unemployed", type="count") %>%
  as_tibble() %>%
  select(unemployed, estimate, conf.low, conf.high)
```

```
## # A tibble: 2 × 4
##   unemployed   estimate conf.low conf.high
##   <fct>           <dbl>    <dbl>     <dbl>
## 1 Employed         1.42    1.25       1.59
## 2 Not Employed     1.94    0.410      3.48
```

```r
avg_predictions(mod, variables="unemployed", type="prob") %>% hea
  as_tibble() %>%
  select(unemployed, group, estimate, conf.low, conf.high)
```

```
## # A tibble: 2 × 5
##   unemployed   group estimate conf.low conf.high
##   <fct>        <chr>    <dbl>    <dbl>     <dbl>
## 1 Employed     0        0.503    0.466     0.540
## 2 Not Employed 0        0.874    0.773     0.976
```

```r
avg_predictions(mod, variables="unemployed", type="response")  %>
  as_tibble() %>%
  select(unemployed, estimate, conf.low, conf.high)
```

```
## # A tibble: 2 × 4
##   unemployed   estimate conf.low conf.high
##   <fct>           <dbl>    <dbl>     <dbl>
## 1 Employed        1.04   0.948        1.12
## 2 Not Employed    0.330  0.0181       0.642
```

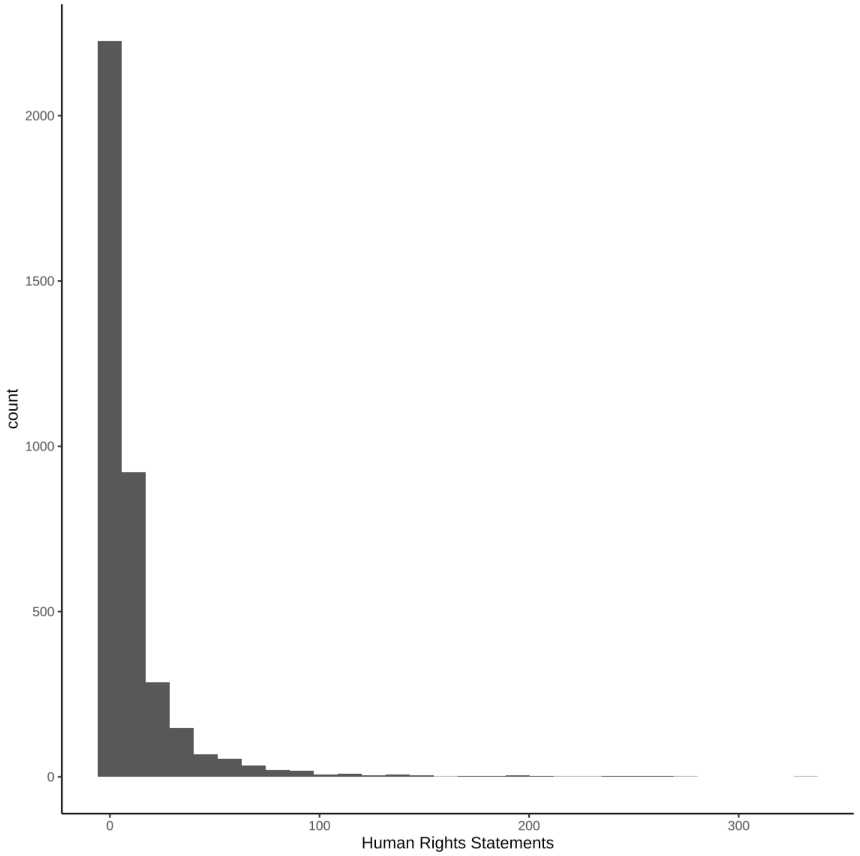
Example: Modeling Manifestos

The Comparative Manifestos Project has data on the number of statements in a party's manifesto devoted to various different topics. We want to model the number of "freedom and human rights" statements.

```
man <- import("data/man2014.dta")
man$num201 <- floor((man$per201/100)*man$total)
```

Histogram of Statements

```
ggplot(man, aes(x=num201)) + geom_histogram() + theme_classic() + labs(x="Human Rights Statements")
```



Offsets (exposure)

An offset (or exposure) term is a way of building into the model that observations have differential abilities to generate positive counts.

- Usually, the offset is the log of the maximum possible count (or exposure time).

In the Poisson model:

$$\log(E(Y|X)) = Xb$$

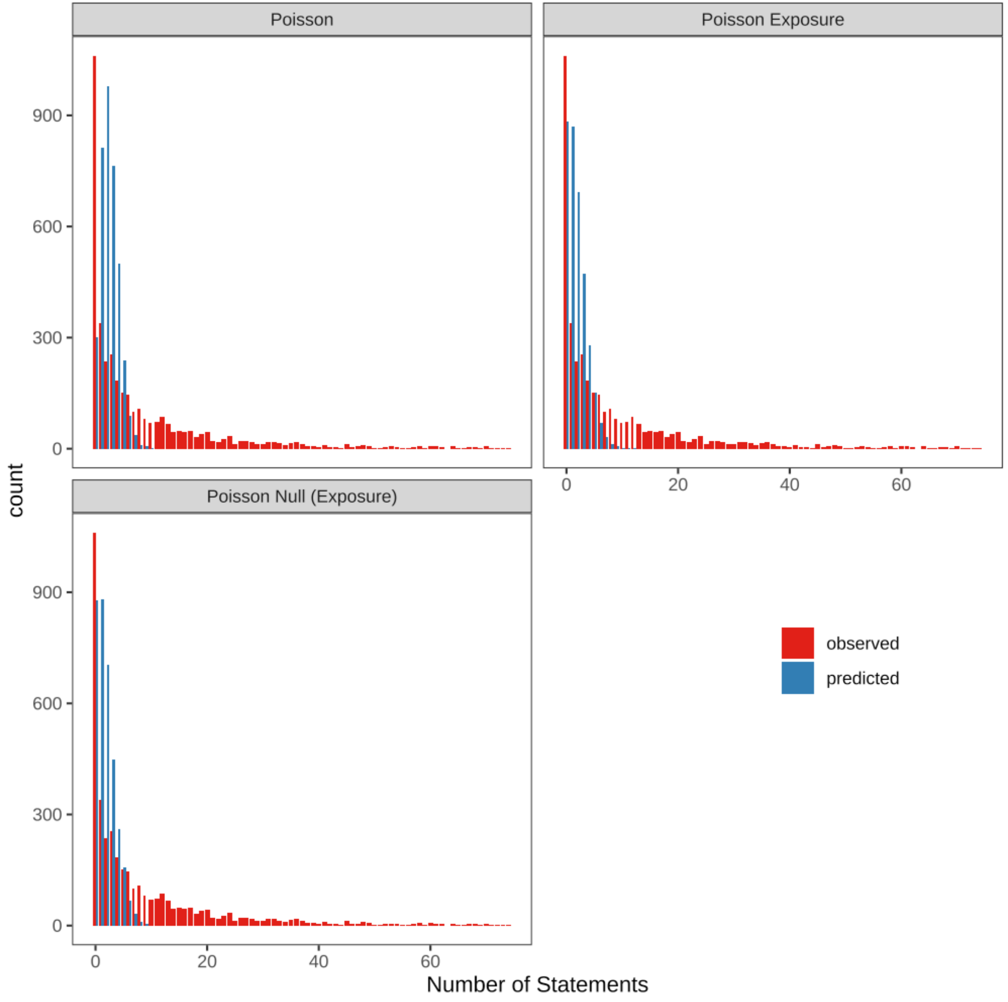
With an exposure term:

$$\begin{aligned}
\log\left(\frac{E(Y|X)}{\text{Exopsure}}\right) &= Xb \\
\log(E(Y|X)) - \log(\text{Exposure}) &= Xb \\
\log(E(Y|X)) &= \log(Xb) + \log(\text{Exposure})
\end{aligned}$$

Poisson Model

```
# Without Exposure
tmp <- na.omit(man[,c("num201", "total", "rile")])
mod <- glm(num201 ~ rile,
           data=tmp,
           family=poisson)
mode0 <- glm(num201 ~ 1 + offset(log(total)),
             data=tmp,
             family=poisson)
mode <- glm(num201 ~ 1 + rile + offset(log(total)),
             data=tmp,
             family=poisson)
AIC(mod, mode0, mode)
```

```
##       df       AIC
## mod    2 109874.28
## mode0  1  53591.16
## mode   2  52566.84
```

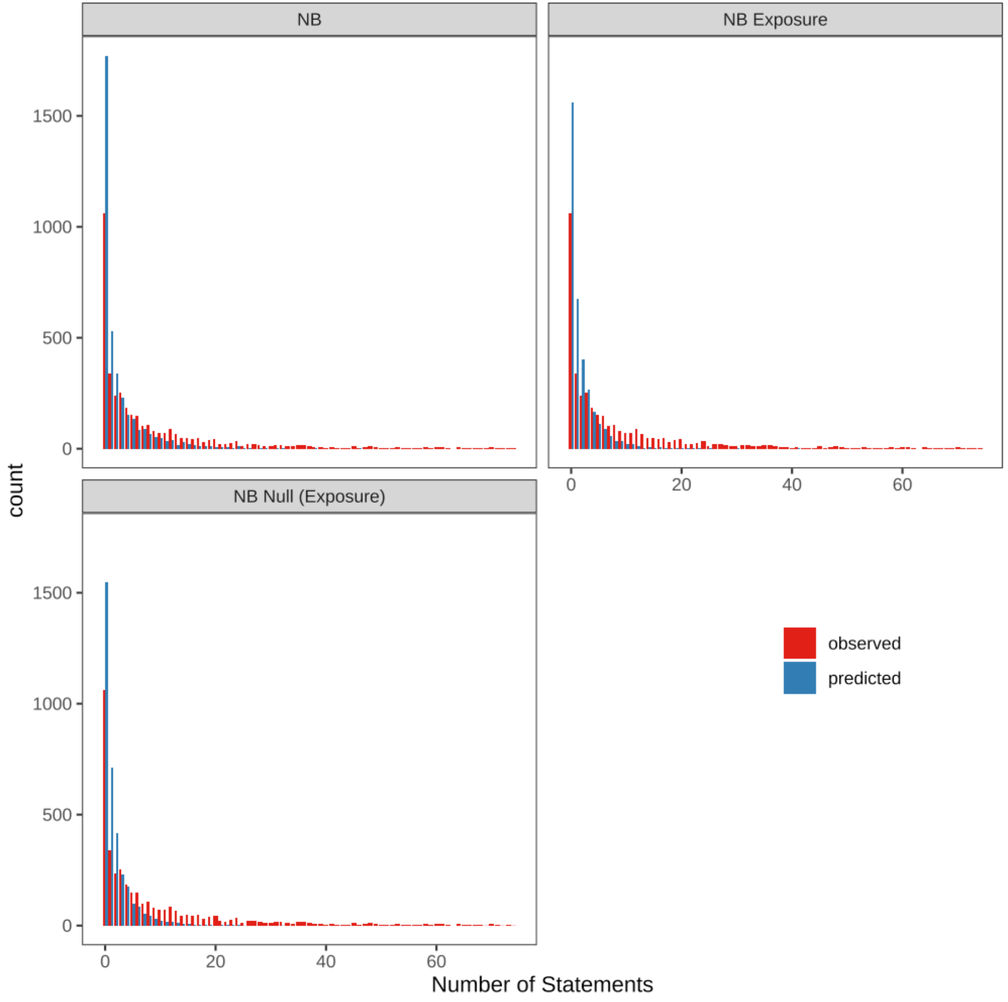


Negative Binomial Model

```
# Without exposure
mod2 <- MASS::glm.nb(num201 ~ rile, data=tmp)
mod2e0 <- MASS::glm.nb(num201 ~ 1 +
                 offset(log(total)),
               data=tmp)

mod2e <- MASS::glm.nb(num201 ~ rile +
                 offset(log(total)),
               data=tmp)
AIC(mod2, mod2e0, mod2e)
```

```
##         df      AIC
## mod2     3 24763.78
## mod2e0   2 22361.96
## mod2e    3 22256.06
```



Binomial Model

When we know the number of possibilities for each count (in this case, the number of sentences in the party's manifesto), then we can use that information. Recall the Binomial distribution.

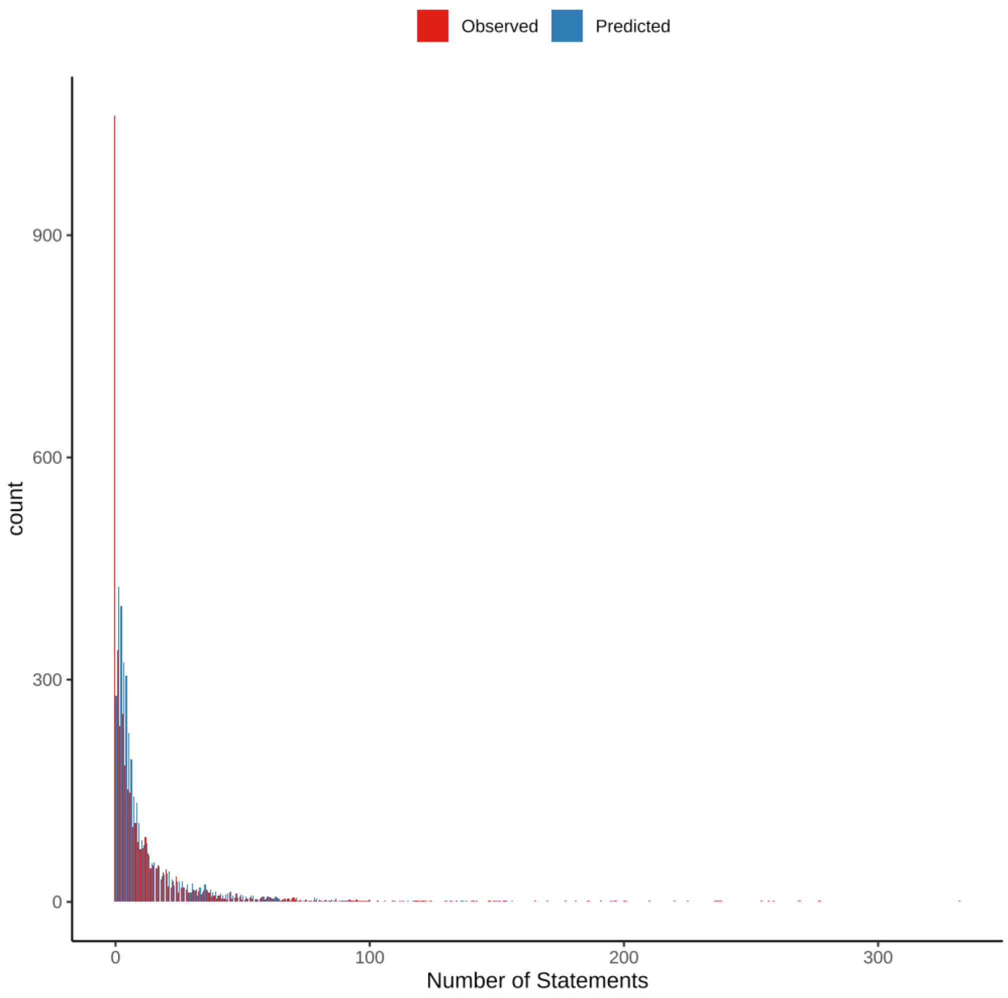
$$Pr(y = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Here, we're using a regression model where we parameterize p.

$$Pr(y_i = k_i) = \binom{n_i}{k_i} p_i^{k_i} (1 - p_i)^{n_i - k_i} \mathrm{logit}(p_i) \qquad = \mathbf{x}_i \beta$$

Binomial Model Example

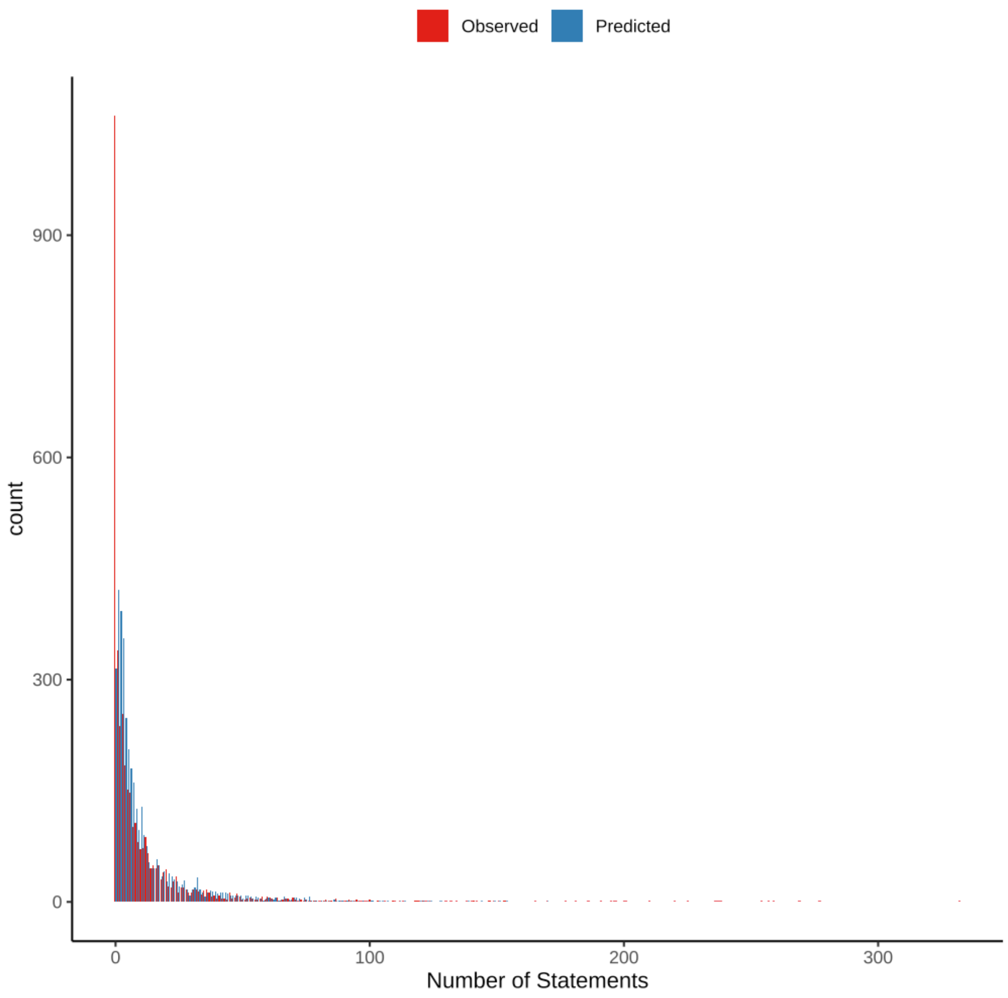
```
tmp$other <- floor(tmp$total - tmp$num201)
mod3 <- glm(cbind(num201, other) ~ rile, data=tmp, family=binomia
```



Quasi-Binomial Model Example

The `quasibinomial` link accounts for overdispersion in the binomial data. With binomial data, $E(y_i) = n_i p_i$ and the variance is $\text{var}(y_i) = n_i p_i(1 - p_i)$. The quasibinomial adds a dispersion parameter, like the negative binomial does.

```
mod3q <- glm(cbind(num201, other) ~ rile, data=tmp, family=quasib
```



Effects

```
s <- seq(-75, 90, length=100)
mode <- glm(num201 ~ 1 + rile + offset(log(total)),
            data=tmp,
            family=poisson)
mod2e <- MASS::glm.nb(num201 ~ rile +
                offset(log(total)),
              data=tmp)
mod3 <- glm(cbind(num201, other) ~ rile, data=tmp, family=quasibinomial)
ap1 <- avg_predictions(mode, variables=list(rile = s))
ap2 <- avg_predictions(mod2e, variables=list(rile = s))
ap3 <- avg_predictions(mod3, variables=list(rile = s))
ap3q <- avg_predictions(mod3q, variables=list(rile = s))
plot.dat <- ap1 %>%
  as_tibble() %>%
  mutate(method = "Poisson (E)") %>%
  bind_rows(ap2 %>% as_tibble() %>% mutate(method="NB (E)"),
            ap3 %>% as_tibble() %>%
              mutate(method="Binom",
                     across(c(estimate, conf.low, conf.high),
                            ~.x * median(tmp$total))),
            ap3q %>% as_tibble() %>%
              mutate(method="Quasi-Binom",
                     across(c(estimate, conf.low, conf.high),
                            ~.x * median(tmp$total))))

ggplot(plot.dat, aes(x=rile,y=estimate)) +
  geom_ribbon(aes(ymin = conf.low, ymax=conf.high, fill
                  =method),
              alpha=.25) +
  geom_line(aes(color = method)) +
  theme_classic()
```

Recap

1. Develop Count models - poisson, negative binomial, binomial
2. Effects and Effect Displays
3. Model Fit and Evaluation
4. Discuss Overdispersion
5. Zero-inflation and hurdle models

Exercise

A while back, I collected some data on scientific literacy in the US (along with a bunch of other stuff). We asked 12 True-False questions about science and recorded peoples' answers. In the `data/science.dta` file, you'll find the answers to those questions, along with the number of correct answers each respondent gave and the respondent's age, education, income, race and region of residence. Using the data, do the following.

```
library(rio)
library(dplyr)
sci <- import("data/science.dta")
sci <- sci %>%
  mutate(across(age_group:income, factorize))
```

Estimate these models:

1. Poisson without offset
2. Poisson with offset of `log(n_ans)`
3. Binomial with `n=n_ans
4. OLS where y is the number of right answers.

Variables

- age_group - What is your Age
- education - What is the highest level of education you have attained?
- race - With which race do you most closely identify?
- region - In what region do you live?
- income - In what range does your gross household income fall?
- Q47 - The Sun goes around the Earth
- Q48 - The center of the Earth is very hot
- Q49 - The oxygen we breathe comes from plants
- Q50 - Radioactive milk can be made safe by boiling it
- Q51 - The continents on which we live have been moving for millions of years and will continue to move
- Q52 - It is the mother's genes that decide whether the baby is a boy or a girl
- Q53 - The earliest humans lived at the same time as the dinosaurs
- Q54 - Antibiotics kill viruses as well as bacteria
- Q55 - Lasers work by focusing sound waves
- Q56 - All radioactivity is man-made
- Q57 - Human beings, as we know them today, developed from earlier species of animals
- Q58 - It takes one month for the Earth go to around the Sun
- n_right - Number of correct answers
- n_ans - Number of questions answered
- n_asked - Number of questions asked
- n_wrong - Number of questions answered incorrectly