432 Class 19 Slides

github.com/THOMASELOVE/432-2018

2018-03-27

Setup

```
library(skimr)
library(arm)
library(rms)
library(boot)
library(MASS)
library(HSAUR)
library(pscl)
library(lmtest)
library(sandwich)
library(broom)
library(tidyverse)
```

Project 2 Instructions

Project 2

Instructions are here. There are three deliverables.

1. 2018-04-17 Registration/Scheduling Form
    - If you can't do May 3, 7, and 8, here's the place to tell me.
2. Portfolio (R Markdown + HTML + data or pseudo-data)
    - 3 hours before your presentation
    - 2 template options, or go off on your own (carefully)
3. Presentation May 3, 7 or 8
    - in a few cases by special arrangement, before May 3

Overview

Today's Materials

Regression Models for Count Outcomes

- Poisson Regression model
- Negative Binomial Regression model
- Zero-inflated models

The `medicare` data

The `medicare` example

The data we will use come from the `NMES1988` data set in R's `AER` package, although I have built a cleaner version for you in the `medicare.csv` file on our web site. These are essentially the same data as are used in my main resource from the University of Virginia for hurdle models.

These data are a cross-section originating from the US National Medical Expenditure Survey (NMES) conducted in 1987 and 1988. The NMES is based upon a representative, national probability sample of the civilian non-institutionalized population and individuals admitted to long-term care facilities during 1987. The data are a subsample of individuals ages 66 and over all of whom are covered by Medicare (a public insurance program providing substantial protection against health-care costs), and some of whom also have private supplemental insurance.

```
medicare <- read.csv("medicare.csv") %>% tbl_df
```

The `medicare` code book

Variable	Description
`subject`	subject number
`visits`	outcome of interest: number of physician office visits
`hospital`	number of hospital stays
`health`	self-perceived health status (poor, average, excellent)
`chronic`	number of chronic conditions
`sex`	male or female
`school`	number of years of education
`insurance`	is the subject (also) covered by private insurance? (yes or no)

Today's Goal

Predict `visits` using some combination of these 6 predictors...

Predictor	Description
`hospital`	number of hospital stays
`health`	self-perceived health status (poor, average, excellent)
`chronic`	number of chronic conditions
`sex`	male or female
`school`	number of years of education
`insurance`	is the subject (also) covered by private insurance? (yes or no)

The `medicare` tibble

```
# A tibble: 4,406 x 8
   subject visits hospital health  chronic sex    school
     <int>  <int>    <int> <fct>     <int> <fct>   <int>
 1       1      5        1 average       2 male        6
 2       2      1        0 average       2 female     10
 3       3     13        3 poor          4 female     10
 4       4     16        1 poor          2 male        3
 5       5      3        0 average       2 female      6
 6       6     17        0 poor          5 female      7
 7       7      9        0 average       0 female      8
 8       8      3        0 average       0 female      8
 9       9      1        0 average       0 female      8
10      10      0        0 average       0 female      8
# ... with 4,396 more rows, and 1 more variable:
#   insurance <fct>
```

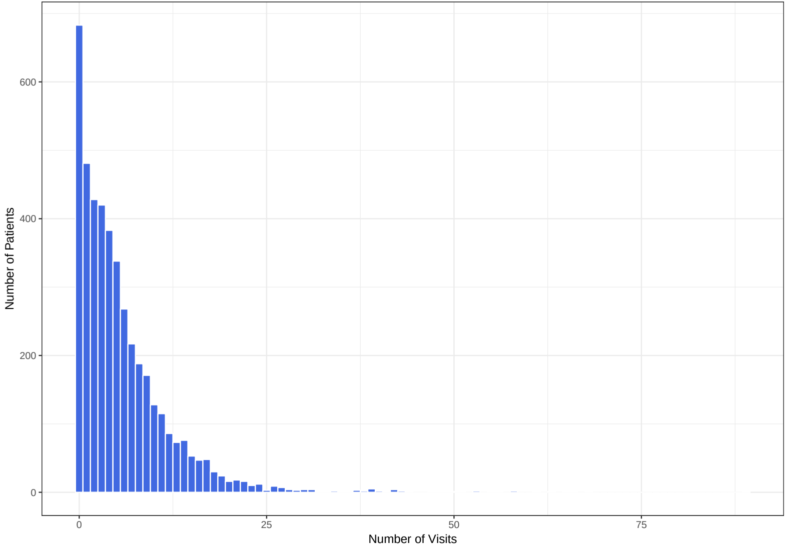
A skim of medicare

```
> skim(medicare)
Skim summary statistics
 n obs: 4406
 n variables: 8

Variable type: factor
  variable missing complete    n n_unique                           top_counts ordered
    health       0     4406 4406        3 ave: 3509, poo: 554, exc: 343, NA: 0   FALSE
 insurance       0     4406 4406        2          yes: 3421, no: 985, NA: 0   FALSE
       sex       0     4406 4406        2        fem: 2628, mal: 1778, NA: 0   FALSE

Variable type: integer
 variable missing complete    n    mean      sd p0     p25 median     p75 p100     hist
  chronic       0     4406 4406    1.54    1.35  0       1      1       2    8
 hospital       0     4406 4406     0.3    0.75  0       0      0       0    8
   school       0     4406 4406   10.29    3.74  0       8     11      12   18
  subject       0     4406 4406 2203.5  1272.05  1 1102.25 2203.5 3304.75 4406
   visits       0     4406 4406    5.77    6.76  0       1      4       8   89
```

Our outcome, `visits`



Counting the visits

```
medicare %>% count(visits)
```

```
# A tibble: 60 x 2
   visits     n
    <int> <int>
 1      0   683
 2      1   481
 3      2   428
 4      3   420
 5      4   383
 6      5   338
 7      6   268
 8      7   217
 9      8   188
10      9   171
# ... with 50 more rows
```

visits summary

```
describe(medicare$visits)
```

```
medicare$visits
       n  missing distinct     Info     Mean      Gmd
    4406        0       60    0.992    5.774    6.227
     .05      .10      .25      .50      .75      .90
       0        0        1        4        8       13
     .95
      17

lowest :  0  1  2  3  4, highest: 63 65 66 68 89
```

Reiterating the Goal

Predict `visits` using some combination of these 6 predictors...

Predictor	Description
`hospital`	number of hospital stays
`health`	self-perceived health status (poor, average, excellent)
`chronic`	number of chronic conditions
`sex`	male or female
`school`	number of years of education
`insurance`	is the subject (also) covered by private insurance? (yes or no)

Model 1: A Poisson Regression

Poisson Regression

Assume our count data (`visits`) follows a Poisson distribution with a mean conditional on our predictors.

```
mod_1 <- glm(visits ~ hospital + health + chronic +
                  sex + school + insurance,
              data = medicare, family = "poisson")
```

Remember the sample size here. Is statistical significance going to be our problem?

Model 1 (Poisson Regression)

```
mod_1
```

```
Call:  glm(formula = visits ~ hospital + health + chronic + se
    insurance, family = "poisson", data = medicare)

Coefficients:
    (Intercept)         hospital  healthexcellent  
        1.02887          0.16480         -0.36199  
     healthpoor          chronic          sexmale  
        0.24831          0.14664         -0.11232  
         school     insuranceyes  
        0.02614          0.20169  

Degrees of Freedom: 4405 Total (i.e. Null);  4398 Residual
Null Deviance:      26940 
Residual Deviance: 23170    AIC: 35960
```

`tidy(mod_1)` with rounding and p values

```
tidy(mod_1, conf.int = F) %>%
    kable(format = "pandoc", digits = 2)
```

term	estimate	std.error	statistic	p.value
(Intercept)	1.03	0.02	43.26	0
hospital	0.16	0.01	27.48	0
healthexcellent	-0.36	0.03	-11.95	0
healthpoor	0.25	0.02	13.92	0
chronic	0.15	0.00	32.02	0
sexmale	-0.11	0.01	-8.68	0
school	0.03	0.00	14.18	0
insuranceyes	0.20	0.02	11.96	0

tidy(mod_1) with rounding and CI

```
tidy(mod_1, conf.int = T) %>%
    select(-statistic, -p.value) %>%
    kable(format = "pandoc", digits = 2)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	1.03	0.02	0.98	1.08
hospital	0.16	0.01	0.15	0.18
healthexcellent	-0.36	0.03	-0.42	-0.30
healthpoor	0.25	0.02	0.21	0.28
chronic	0.15	0.00	0.14	0.16
sexmale	-0.11	0.01	-0.14	-0.09
school	0.03	0.00	0.02	0.03
insuranceyes	0.20	0.02	0.17	0.23

Interpret the `male` and `chronic` variables

We have an additive model in the `log(visits)` scale.

- Coefficient of `male` is -0.11, with 95% CI (-0.14, -0.09)
  - If Harry and Sally share the same values for all other variables in the model, but Harry is male and Sally is female, then log(visits) for Harry is estimated to be -0.11 smaller than log(visits) for Sally.
- Coefficient of `chronic` is 0.15, with 95% CI (0.14, 0.16)
  - If Harry and Steve share the same values for all other variables in the model, but Harry has one more chronic illness than Steve, then log(visits) for Harry is estimated to be 0.15 larger than log(visits) for Steve.

The Fitted Equation

$$
\begin{gathered}
log(\text{ visits }) = 1.03 + 0.16 \text{ hospital } - 0.36(\text{ health } = \text{ excellent })+ \\
0.25(\text{ health } = \text{ poor }) + 0.15 \text{ chronic } - 0.11(\text{ sex } = \text{ male }) \\
+0.03 \text{ school } + 0.20(\text{ insurance } = \text{ yes})
\end{gathered}
$$

So, the count of visits follows a Poisson distribution, with mean λ, where:

$$
\begin{gathered}
\lambda = exp[1.03 + 0.16 \text{ hospital } - 0.36(\text{ health } = \text{ excellent })+ \\
0.25(\text{ health } = \text{ poor }) + 0.15 \text{ chronic } - 0.11(\text{ sex } = \text{ male }) \\
+0.03 \text{ school } + 0.20(\text{ insurance } = \text{ yes})]
\end{gathered}
$$

Expressing the model differently

```
tidy(mod_1, exponentiate = T, conf.int = T) %>%
    select(-statistic, -p.value) %>%
    kable(format = "pandoc", digits = 2)
```

term	estimate	std.error	conf.low	conf.high
(Intercept)	2.80	0.02	2.67	2.93
hospital	1.18	0.01	1.17	1.19
healthexcellent	0.70	0.03	0.66	0.74
healthpoor	1.28	0.02	1.24	1.33
chronic	1.16	0.00	1.15	1.17
sexmale	0.89	0.01	0.87	0.92
school	1.03	0.00	1.02	1.03
insuranceyes	1.22	0.02	1.18	1.26

Interpret the `male` and `chronic` after exponentiation

Now, we have a multiplicative model in the `visits` scale.

- exp(`male`) is 0.89, with 95% CI (0.87, 0.92)
  - If Harry and Sally share the same values for all other variables in the model, but Harry is male and Sally is female, then visits for Harry is estimated to be 0.89 times the visits for Sally. Harry is expected to have 89% of the visits Sally has.
- exp(`chronic`) is 1.16, with 95% CI (1.15, 1.17)
  - If Harry and Steve share the same values for all other variables in the model, but Harry has one more chronic illness than Steve, then visits for Harry is estimated to be 1.16 times visits for Steve. Harry is expected to have 116% of the visits Steve has.

display(mod_1) from arm package

```
display(mod_1)
```

```
glm(formula = visits ~ hospital + health + chronic + sex + sch
    insurance, family = "poisson", data = medicare)
                 coef.est coef.se
(Intercept)       1.03     0.02  
hospital          0.16     0.01  
healthexcellent -0.36     0.03  
healthpoor        0.25     0.02  
chronic           0.15     0.00  
sexmale          -0.11     0.01  
school            0.03     0.00  
insuranceyes      0.20     0.02  
---
  n = 4406, k = 8
  residual deviance = 23167.8, null deviance = 26942.9 (differ
```

summary(mod_1)

```
> summary(mod_1)

Call:
glm(formula = visits ~ hospital + health + chronic + sex + school +
    insurance, family = "poisson", data = medicare)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4055  -1.9962  -0.6737   0.7049  16.3620

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.028874   0.023785  43.258   <2e-16 ***
hospital         0.164797   0.005997  27.478   <2e-16 ***
healthexcellent -0.361993   0.030304 -11.945   <2e-16 ***
healthpoor       0.248307   0.017845  13.915   <2e-16 ***
chronic          0.146639   0.004580  32.020   <2e-16 ***
sexmale         -0.112320   0.012945  -8.677   <2e-16 ***
school           0.026143   0.001843  14.182   <2e-16 ***
insuranceyes     0.201687   0.016860  11.963   <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: 26943  on 4405  degrees of freedom
Residual deviance: 23168  on 4398  degrees of freedom
AIC: 35959

Number of Fisher Scoring iterations: 5
```

confint(mod_1)

```r
confint(mod_1)
```

```
Waiting for profiling to be done...
```

```
                      2.5 %      97.5 %
(Intercept)       0.98214199  1.07537749
hospital          0.15296768  0.17647770
healthexcellent  -0.42189692 -0.30309508
healthpoor        0.21324851  0.28319940
chronic           0.13764952  0.15560166
sexmale          -0.13771836 -0.08697322
school            0.02253268  0.02975845
insuranceyes      0.16873364  0.23482518
```

Testing the Predictors

- Wald tests are provided with the Poisson regression summary.
- ANOVA approach lets us do sequential likelihood ratio tests.

```
> anova(mod_1, test = "Chisq")
Analysis of Deviance Table

Model: poisson, link: log

Response: visits

Terms added sequentially (first to last)


         Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                      4405      26943
hospital  1  1494.22      4404      25449 < 2.2e-16 ***
health    2   756.68      4402      24692 < 2.2e-16 ***
chronic   1   961.76      4401      23730 < 2.2e-16 ***
sex       1    61.27      4400      23669 4.981e-15 ***
school    1   353.21      4399      23316 < 2.2e-16 ***
```

Making Predictions

```r
medicare %>% head(2)
```

```
# A tibble: 2 x 8
  subject visits hospital health  chronic sex    school
    <int>  <int>    <int> <fct>     <int> <fct>   <int>
1       1      5        1 average       2 male        6
2       2      1        0 average       2 female     10
# ... with 1 more variable: insurance <fct>
```

Store Predictions

```
mod_1_aug <- augment(mod_1, medicare,
                     type.predict = "response",
                     type.residuals = "response")

mod_1_aug %>% select(visits, .fitted, .resid) %>% head(2)
```

```
  visits  .fitted     .resid
1      5 5.658592 -0.6585917
2      1 5.961186 -4.9611865
```

Calculating a Pseudo-R^2 for `mod_1`

```
(mod_1_r <- with(mod_1_aug, cor(visits, .fitted)))
```

```
[1] 0.3144637
```

```
(mod_1_r^2)
```

```
[1] 0.09888744
```

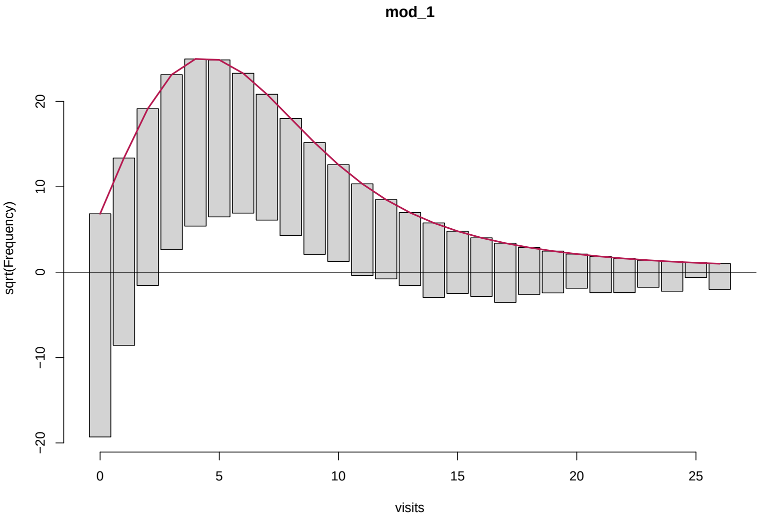
Summarizing the Model's Fit

```
glance(mod_1)
```

```
  null.deviance df.null    logLik      AIC      BIC
1      26942.92    4405 -17971.61 35959.23 36010.35
  deviance df.residual
1 23167.81        4398
```

Rootogram: See the Fit (using default choices)

```
countreg::rootogram(mod_1)
```

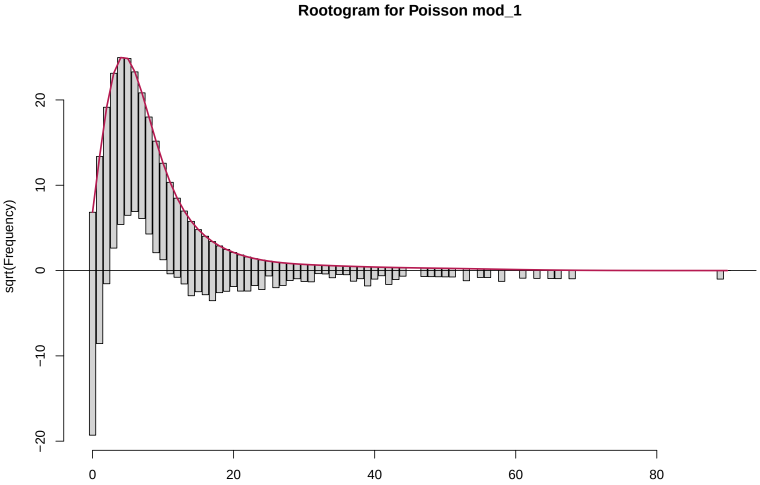


Interpreting the Hanging Rootogram

- The red curved line is the theoretical Poisson fit.
- "Hanging" from each point on the red line is a bar, the height of which represents the difference between expected and observed counts.
  - A bar hanging below 0 indicates underfitting. (In this case, this refers to when our predict fewer values than the data show.)
  - A bar hanging above 0 indicates overfitting. (In this case, this refers to when our model predicts more values than the data show.)
- The counts have been transformed with a square root transformation to prevent smaller counts from getting obscured and overwhelmed by larger counts.

The Complete Hanging Rootogram for Model 1

```
countreg::rootogram(mod_1, max = 90,
                    main = "Rootogram for Poisson mod_1")
```



Interpreting the Rootogram for Model 1

In `mod_1`, we see a great deal of underfitting for counts of 0 and 1, then overfitting for visit counts in the 3-10 range, with some underfitting again at more than a dozen or so visits.

- Our Poisson model (`mod_1`) doesn't fit enough zeros or ones, and fits too many 3-12 values, then not enough of the higher values.

How many zero counts does Model 1 predict?

```
lam <- predict(mod_1, type = "response") # exp. mean count
exp <- sum(dpois(x = 0, lambda = lam)) # sum the prob(0)
round(exp)
```

```
[1] 47
```

How many subjects with zero visits did we see?

```
medicare %>% count(visits == 0)
```

```
# A tibble: 2 x 2
  `visits == 0`     n
  <lgl>         <int>
1 FALSE          3723
2 TRUE            683
```

Do we have an overdispersion problem?

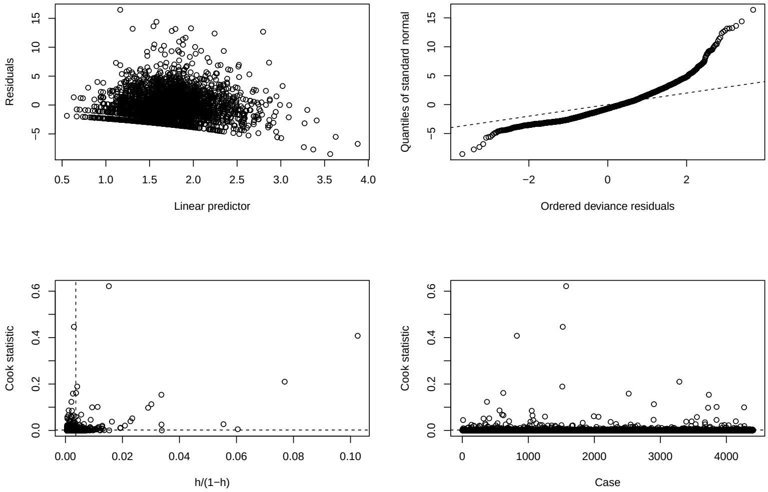
```
overdispersion ratio is  6.706136

p value of overdispersion test:  0
```

Code used on previous slide

```
yhat <- predict(mod_1, type = "response")
n <- 4406; k <- 8 # use display(mod_1) to see these
z <- (mod_1_aug$visits - mod_1_aug$.fitted) /
    sqrt(mod_1_aug$.fitted)
cat("overdispersion ratio is ", sum(z^2)/ (n - k), "\n")
cat("p value of overdispersion test: ",
    pchisq(sum(z^2)/(n-k), n-k), "\n")
```

`glm.diag.plots` from `boot` for Model 1



The Negative Binomial Model (mod_2)

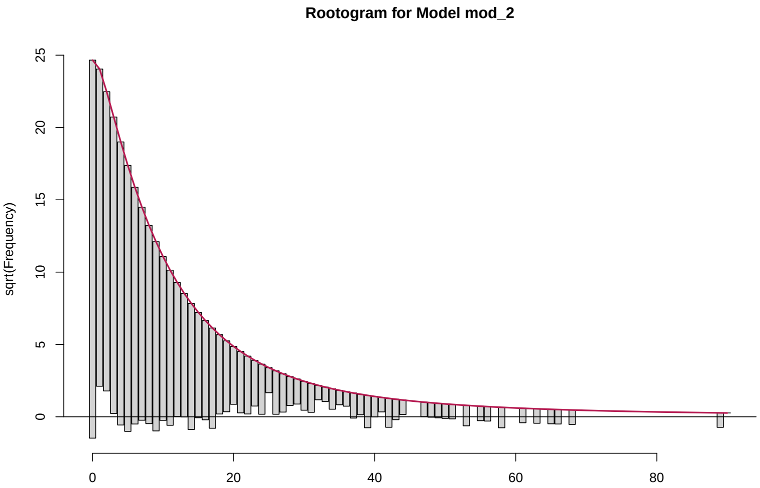
Fitting the Negative Binomial Model

Looks like our data are overdispersed compared to what a Poisson model expects.

```
mod_2 <- MASS::glm.nb(visits ~ hospital + health + chronic +
                      sex + school + insurance,
                 data = medicare)
```

Rootogram for Negative Binomial Model

```
countreg::rootogram(mod_2, max = 90,
                    main = "Rootogram for Model mod_2")
```



Save predicted values and residuals

```
mod_2_aug <- medicare %>%
    mutate(fitted = fitted(mod_2, type = "response"),
           resid = resid(mod_2, type = "response"))

mod_2_aug %>%
    dplyr::select(visits, fitted, resid) %>%
    head(2)
```

```
# A tibble: 2 x 3
  visits fitted  resid
   <int>  <dbl>  <dbl>
1      5   5.79 -0.787
2      1   5.88 -4.88
```

Pseudo-R^2 for Neg. Bin. model (mod_2)

We can calculate a proxy for R^2 as the squared correlation of the fitted values and the observed values.

```
mod2_r <- with(mod_2_aug, cor(visits, fitted))
mod2_r^2
```

```
[1] 0.08271151
```

What is a Zero-Inflated Model?

Zero-Inflated Poisson (ZIP) model

The zero-inflated Poisson or (ZIP) model is used to describe count data with an excess of zero counts.

The model posits that there are two processes involved:

- a logit model is used to predict excess zeros
- while a Poisson model is used to predict the counts

The `pscl` package is used here, which can conflict with the `countreg` package we used to fit rootograms.

Fitting the ZIP model (Model `mod_3`)

```
mod_3 <- zeroinfl(visits ~ hospital + health + chronic +
                    sex + school + insurance,
                  data = medicare)
```

summary(mod_3) (and see next 2 slides)

```
> summary(mod_3)

Call:
zeroinfl(formula = visits ~ hospital + health + chronic + sex + school + insurance, data = medicare)

Pearson residuals:
    Min      1Q  Median      3Q     Max 
-5.4092 -1.1579 -0.4769  0.5435 25.0380 

Count model coefficients (poisson with log link):
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)      1.405812   0.024175  58.152  < 2e-16 ***
hospital         0.159011   0.006060  26.239  < 2e-16 ***
healthexcellent -0.304134   0.031151  -9.763  < 2e-16 ***
healthpoor       0.253454   0.017705  14.315  < 2e-16 ***
chronic          0.101836   0.004721  21.571  < 2e-16 ***
sexmale         -0.062332   0.013054  -4.775 1.80e-06 ***
school           0.019144   0.001873  10.221  < 2e-16 ***
insuranceyes     0.080557   0.017145   4.699 2.62e-06 ***

Zero-inflation model coefficients (binomial with logit link):
                Estimate Std. Error z value Pr(>|z|)    
(Intercept)     -0.08102    0.14233  -0.569 0.569219    
hospital        -0.30330    0.09158  -3.312 0.000927 ***
healthexcellent  0.23786    0.14990   1.587 0.112550    
healthpoor       0.02166    0.16170   0.134 0.893431    
chronic         -0.53117    0.04601 -11.545  < 2e-16 ***
sexmale          0.41527    0.08919   4.656 3.22e-06 ***
school          -0.05677    0.01223  -4.640 3.49e-06 ***
insuranceyes    -0.75294    0.10257  -7.341 2.12e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Number of iterations in BFGS optimization: 24 
Log-likelihood: -1.613e+04 on 16 Df
```

Zero-inflation model coefficients in `mod_3`

```
Zero-inflation model coefficients (binomial with logit link):
               Estimate Std. Error z value Pr(>|z|)    
(Intercept)    -0.08102    0.14233  -0.569 0.569219    
hospital       -0.30330    0.09158  -3.312 0.000927 ***
healthexcellent  0.23786    0.14990   1.587 0.112550    
healthpoor      0.02166    0.16170   0.134 0.893431    
chronic        -0.53117    0.04601 -11.545  < 2e-16 ***
sexmale         0.41527    0.08919   4.656 3.22e-06 ***
school         -0.05677    0.01223  -4.640 3.49e-06 ***
insuranceyes   -0.75294    0.10257  -7.341 2.12e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Count model coefficients in `mod_3`

```
Count model coefficients (poisson with log link):
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.405812   0.024175  58.152  < 2e-16 ***
hospital         0.159011   0.006060  26.239  < 2e-16 ***
healthexcellent -0.304134   0.031151  -9.763  < 2e-16 ***
healthpoor       0.253454   0.017705  14.315  < 2e-16 ***
chronic          0.101836   0.004721  21.571  < 2e-16 ***
sexmale         -0.062332   0.013054  -4.775 1.80e-06 ***
school           0.019144   0.001873  10.221  < 2e-16 ***
insuranceyes     0.080557   0.017145   4.699 2.62e-06 ***
```

The Fitted Equation (part 1 of 2)

The form of the model equation for a zero-inflated Poisson regression requires us to take two separate models into account.

First, we have a logistic regression model to predict the log odds of zero visits. . .

```
logit(visits = 0) = -0.08 - 0.30 hospital +
             0.24 health = excellent + 0.21 health = poor -
             0.53 chronic + 0.42 sex = male - 0.06 school -
             0.75 insurance = yes
```

That takes care of the *extra* zeros.

The Fitted Equation (part 2 of 2)

The form of the model equation for a zero-inflated Poisson regression requires us to take two separate models into account.

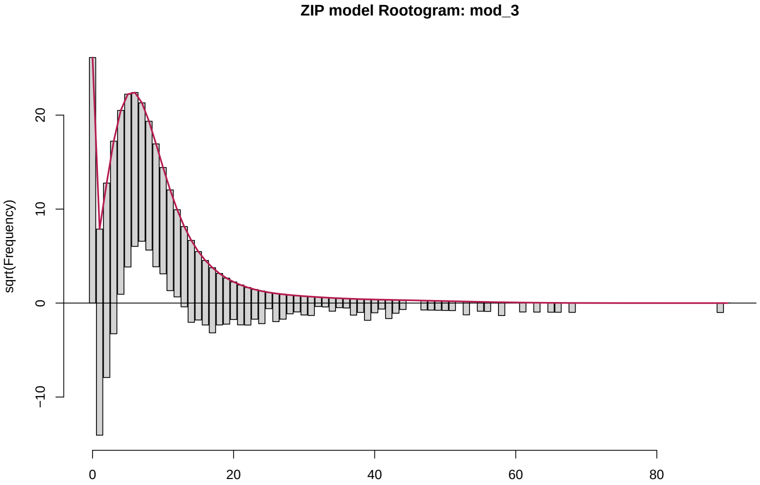
Second, we have a Poisson regression model to predict log(visits)...

```
log(visits) = 1.41 + 0.16 hospital -
             0.30 health = excellent + 0.25 health = poor +
             0.10 chronic - 0.06 sex = male + 0.02 school +
             0.08 insurance = yes
```

This may produce some additional zero count estimates.

Rootogram for ZIP model

```
countreg::rootogram(mod_3, max = 90,
                    main = "ZIP model Rootogram: mod_3")
```



Save predicted values and residuals

```
mod_3_aug <- medicare %>%
    mutate(fitted = fitted(mod_3, type = "response"),
           resid = resid(mod_3, type = "response"))

mod_3_aug %>%
    dplyr::select(visits, fitted, resid) %>%
    head(2)
```

```
# A tibble: 2 x 3
  visits fitted  resid
   <int>  <dbl>  <dbl>
1      5   5.98 -0.982
2      1   6.05 -5.05
```

Is ZIP significantly better than Poisson (Vuong test)

```r
vuong(mod_3, mod_1)
```

```
Vuong Non-Nested Hypothesis Test-Statistic:
(test-statistic is asymptotically distributed N(0,1) under the
 null that the models are indistinguishible)
-------------------------------------------------------------
              Vuong z-statistic             H_A    p-value
Raw                    17.13459 model1 > model2 < 2.22e-16
AIC-corrected          17.05999 model1 > model2 < 2.22e-16
BIC-corrected          16.82163 model1 > model2 < 2.22e-16
```

Pseudo-R^2 for ZIP model (`mod_3`)

We can calculate a proxy for R2 as the squared correlation of the fitted values and the observed values.

```
mod3_r <- with(mod_3_aug, cor(visits, fitted))
mod3_r^2
```

```
[1] 0.1073657
```

The Zero-Inflated Negative Binomial Model

Fitting the Zero-Inflated Negative Binomial (mod_4)

```
mod_4 <- zeroinfl(visits ~ hospital + health + chronic +
                    sex + school + insurance,
                dist = "negbin", data = medicare)
```

summary(mod_4) (and see next 2 slides)

```
> summary(mod_4)

Call:
zeroinfl(formula = visits ~ hospital + health + chronic + sex + school + insurance, data = medicare, dist = "negbin")

Pearson residuals:
    Min      1Q  Median      3Q     Max 
-1.1966 -0.7097 -0.2784  0.3256 17.7661 

Count model coefficients (negbin with log link):
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)      1.193466   0.056737  21.035  < 2e-16 ***
hospital         0.201214   0.020392   9.867  < 2e-16 ***
healthexcellent -0.313540   0.062977  -4.979 6.40e-07 ***
healthpoor       0.287190   0.045940   6.251 4.07e-10 ***
chronic          0.128955   0.011938  10.802  < 2e-16 ***
sexmale         -0.080093   0.031035  -2.581  0.00986 ** 
school           0.021338   0.004368   4.886 1.03e-06 ***
insuranceyes     0.126815   0.041687   3.042  0.00235 ** 
Log(theta)       0.394731   0.035145  11.231  < 2e-16 ***

Zero-inflation model coefficients (binomial with logit link):
                Estimate Std. Error z value Pr(>|z|)    
(Intercept)     -0.06354    0.27668  -0.230  0.81837    
hospital        -0.81760    0.43875  -1.863  0.06240 .  
healthexcellent  0.10488    0.30965   0.339  0.73484    
healthpoor       0.10178    0.44071   0.231  0.81735    
chronic         -1.24630    0.17918  -6.956 3.51e-12 ***
sexmale          0.64937    0.20046   3.239  0.00120 ** 
school          -0.08481    0.02676  -3.169  0.00153 ** 
insuranceyes    -1.15808    0.22436  -5.162 2.45e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Theta = 1.484 
Number of iterations in BFGS optimization: 31 
Log-likelihood: -1.209e+04 on 17 Df
```

Zero-inflation model coefficients in `mod_4`

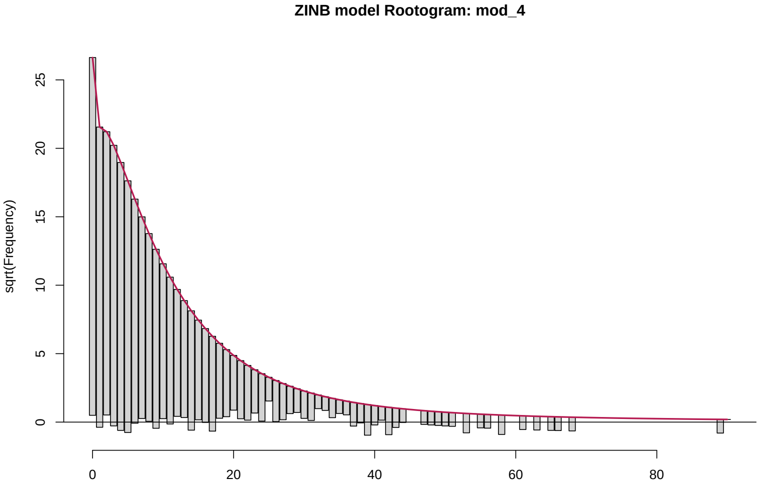
```
Zero-inflation model coefficients (binomial with logit link):
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     -0.06354    0.27668  -0.230  0.81837
hospital        -0.81760    0.43875  -1.863  0.06240 .
healthexcellent  0.10488    0.30965   0.339  0.73484
healthpoor       0.10178    0.44071   0.231  0.81735
chronic         -1.24630    0.17918  -6.956 3.51e-12 ***
sexmale          0.64937    0.20046   3.239  0.00120 **
school          -0.08481    0.02676  -3.169  0.00153 **
insuranceyes    -1.15808    0.22436  -5.162 2.45e-07 ***
```

Count model coefficients in `mod_4`

```
Count model coefficients (negbin with log link):
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      1.193466   0.056737  21.035  < 2e-16 ***
hospital         0.201214   0.020392   9.867  < 2e-16 ***
healthexcellent -0.313540   0.062977  -4.979 6.40e-07 ***
healthpoor       0.287190   0.045940   6.251 4.07e-10 ***
chronic          0.128955   0.011938  10.802  < 2e-16 ***
sexmale         -0.080093   0.031035  -2.581  0.00986 **
school           0.021338   0.004368   4.886 1.03e-06 ***
insuranceyes     0.126815   0.041687   3.042  0.00235 **
Log(theta)       0.394731   0.035145  11.231  < 2e-16 ***
```

Rootogram for ZINB model

```
countreg::rootogram(mod_4, max = 90,
                    main = "ZINB model Rootogram: mod_4")
```



Save predicted values and residuals

```
mod_4_aug <- medicare %>%
    mutate(fitted = fitted(mod_4, type = "response"),
           resid = resid(mod_4, type = "response"))

mod_4_aug %>%
    dplyr::select(visits, fitted, resid) %>%
    head(2)
```

```
# A tibble: 2 x 3
  visits fitted resid
   <int>  <dbl> <dbl>
1      5   6.14 -1.14
2      1   5.94 -4.94
```

Is ZINB significantly better than Negative Binomial?

```r
vuong(mod_4, mod_2)
```

```
Vuong Non-Nested Hypothesis Test-Statistic:
(test-statistic is asymptotically distributed N(0,1) under the
 null that the models are indistinguishible)
-------------------------------------------------------------
              Vuong z-statistic             H_A    p-value
Raw                    5.917202 model1 > model2 1.6373e-09
AIC-corrected          5.324799 model1 > model2 5.0532e-08
BIC-corrected          3.431859 model1 > model2 0.00029973
```

Pseudo-R^2 for ZINB model (`mod_4`)

We can calculate a proxy for R2 as the squared correlation of the fitted values and the observed values.

```
mod4_r <- with(mod_4_aug, cor(visits, fitted))
mod4_r^2
```

```
[1] 0.09620424
```

So Far . . .

Model	Pseudo-R^2	Rootogram?	Comments
Poisson	0.099	Many problems.	Data appear overdispersed.
Neg. Bin.	0.083	Better.	Still not enough zeros.
ZIP	0.107	All but 0 a problem.	Not enough 1-3.
ZINB	0.096	Better.	Zeros not a perfect fit.

Next Time - The Hurdle Model

The hurdle model is a two-part model that specifies one process for zero counts and another process for positive counts. The idea is that positive counts occur once a threshold is crossed, or put another way, a hurdle is cleared. If the hurdle is not cleared, then we have a count of 0.

- The first part of the model is typically a binary logit model. This models whether an observation takes a positive count or not.
- The second part of the model is usually a truncated Poisson or Negative Binomial model. Truncated means we're only fitting positive counts. If we were to fit a hurdle model to our [medicare] data, the interpretation would be that one process governs whether a patient visits a doctor or not, and another process governs how many visits are made.

Next Time - The Tobit (Censored Regression) Model

The idea of the tobit model (sometimes called a censored regression model) is to estimate associations for outcomes where we can see either left-censoring (censoring from below) or right-censoring (censoring from above.)

We'll look at a different example for the tobit, since we don't have an upper bound (technically) for the visit counts in the `medicare` data.