

432 Class 20 Slides

github.com/THOMASELOVE/432-2018

2018-03-29

Setup

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

Overview

Today's Materials

Regression Models for Count Outcomes

- Poisson Regression model
- Negative Binomial Regression model
- Zero-inflated models
 - ZIP (Zero-inflated Poisson)
 - ZINB (Zero-inflated Neg. Binomial)
- Hurdle models
- Tobit (censored) regression models

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

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)

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")
```

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-R² 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			

Building a Rootogram

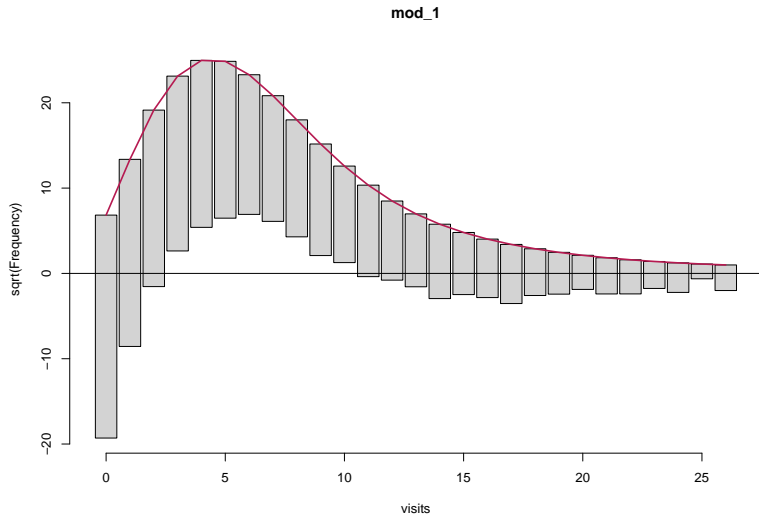
To build a rootogram, you need to load the `countreg` package. This package is housed on R-Forge, rather than CRAN, so you need to install it with ...

```
install.packages("countreg",  
                 repos="http://R-Forge.R-project.org")
```

in order to use it.

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 that the model under-predicts that value. (Model predicts fewer values than the data show.)
 - A bar hanging above 0 indicates over-prediction of that value. (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.

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
```

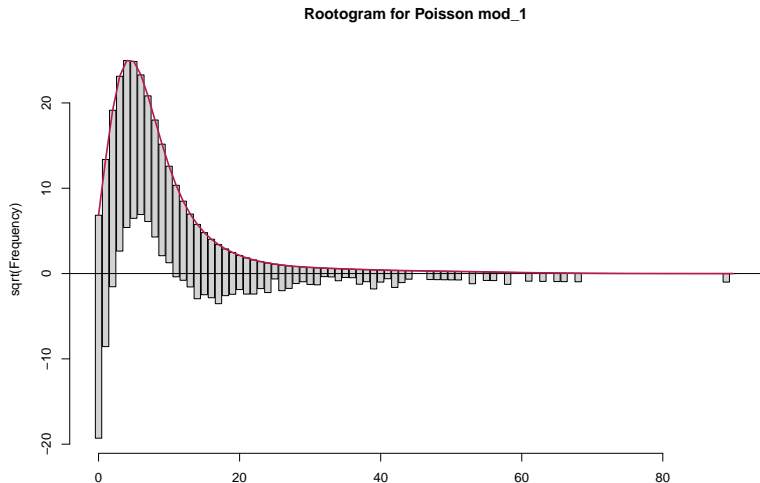
How many subjects with zero visits did we see?

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

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

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.

Do we have an overdispersion problem?

overdispersion ratio is 6.706136

p value of overdispersion test: 0

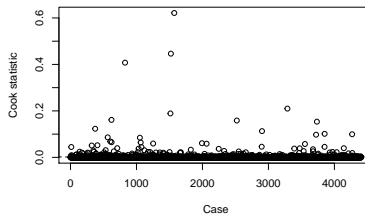
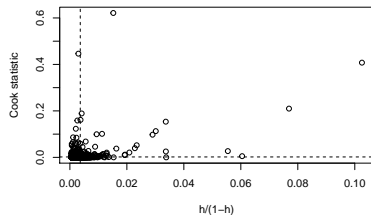
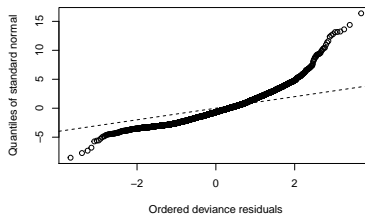
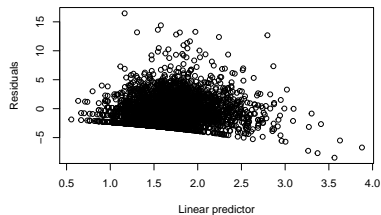
Dealing with Overdispersion?

To address the overdispersion, we'll adopt a negative binomial approach, in part because the rootogram tool we're using doesn't handle the quasipoisson model.

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



Model 2: A Negative Binomial Model

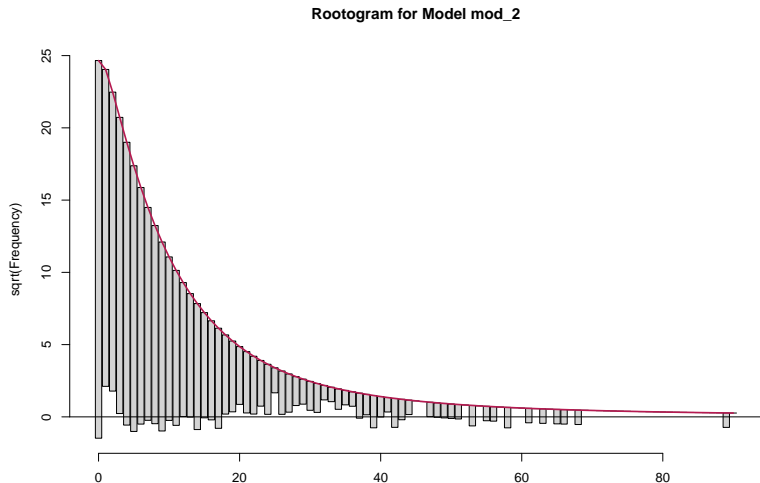
Fitting the Negative Binomial Model

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
```

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.

Model 3: Zero-Inflated Poisson (ZIP) 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 to fit these zero-inflated models.

```
mod_3 <- pscl::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
```

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. . .

$$\begin{aligned}\text{logit}(\text{visits} = 0) = & -0.08 - 0.30 \text{ hospital} + \\ & 0.24 \text{ health} = \text{excellent} + 0.21 \text{ health} = \text{poor} - \\ & 0.53 \text{ chronic} + 0.42 \text{ sex} = \text{male} - 0.06 \text{ school} - \\ & 0.75 \text{ insurance} = \text{yes}\end{aligned}$$

That takes care of the *extra* zeros.

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
```

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 $\log(\text{visits})$...

$$\begin{aligned}\log(\text{visits}) = & 1.41 + 0.16 \text{ hospital} - \\ & 0.30 \text{ health} = \text{excellent} + 0.25 \text{ health} = \text{poor} + \\ & 0.10 \text{ chronic} - 0.06 \text{ sex} = \text{male} + 0.02 \text{ school} + \\ & 0.08 \text{ insurance} = \text{yes}\end{aligned}$$

This may produce some additional zero count estimates.

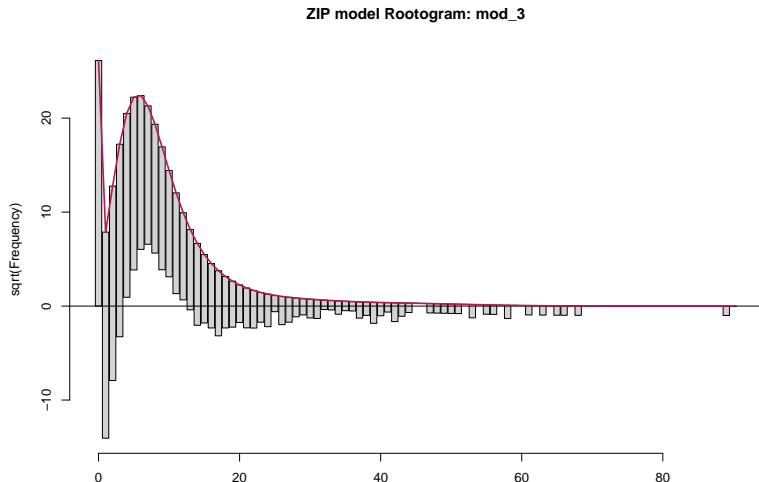
Count model coefficients in mod_3

```
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	***

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 = predict(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)

```
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 indistinguishable)

	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

- Conclusion: ZIP model shows evidence of superiority over Poisson.
- Vuong QH (1989) Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica* 57:307-333.

Pseudo- R^2 for ZIP model (mod_3)

We can calculate a proxy for R^2 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
```

Model 4: 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 ***
healthexcell -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 .
healthexcell  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

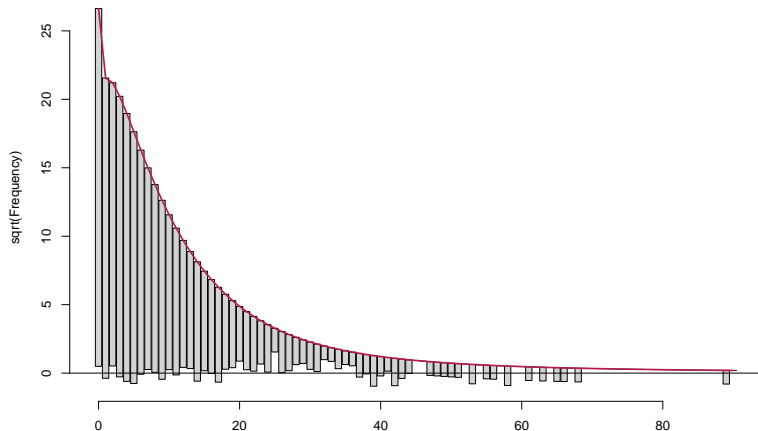
```
Count model coefficients (negbin with log link):
```

	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")
```

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?

```
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 indistinguishable)

	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 R^2 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.

Model 5: The Hurdle Model (Poisson)

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 logistic regression** 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, and not zeros.

In fitting 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.

Fitting a Hurdle Model / Poisson-Logistic

```
mod_5 <- hurdle(visits ~ hospital + health + chronic +  
                sex + school + insurance,  
                dist = "poisson", zero.dist = "binomial",  
                data = medicare)
```


Summary of Hurdle Model / Poisson-Logistic

```
> summary(mod_5)

Call:
hurdle(formula = visits ~ hospital + health + chronic + sex + school +
  insurance, data = medicare, dist = "poisson", zero.dist = "binomial")

Pearson residuals:
      Min       1Q   Median       3Q      Max
-5.4144 -1.1565 -0.4770  0.5432 25.0228

Count model coefficients (truncated poisson with log link):
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.406459   0.024180  58.167 < 2e-16 ***
hospital     0.158967   0.006061  26.228 < 2e-16 ***
healthexcellent -0.303677 0.031150  -9.749 < 2e-16 ***
healthpoor    0.253521   0.017708  14.317 < 2e-16 ***
chronic       0.101720   0.004719  21.557 < 2e-16 ***
sexmale      -0.062247   0.013055  -4.768 1.86e-06 ***
school       0.019078   0.001872   10.194 < 2e-16 ***
insuranceyes  0.080879   0.017139   4.719 2.37e-06 ***

Zero hurdle model coefficients (binomial with logit link):
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.043147   0.139852   0.309 0.757688
hospital     0.312449   0.091437   3.417 0.000633 ***
healthexcellent -0.289570 0.142682  -2.029 0.042409 *
healthpoor   -0.008716  0.161024  -0.054 0.956833
chronic       0.535213   0.045378  11.794 < 2e-16 ***
sexmale      -0.415658   0.087608  -4.745 2.09e-06 ***
school       0.058541   0.011989   4.883 1.05e-06 ***
insuranceyes  0.747120   0.100880   7.406 1.30e-13 ***

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 14
Log-likelihood: -1.613e+04 on 16 Df
```

Logistic Regression Model to predict zeros

```
Zero hurdle model coefficients (binomial with logit link):
```

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	0.043147	0.139852	0.309	0.757688	
hospital	0.312449	0.091437	3.417	0.000633	***
healthexcellent	-0.289570	0.142682	-2.029	0.042409	*
healthpoor	-0.008716	0.161024	-0.054	0.956833	
chronic	0.535213	0.045378	11.794	< 2e-16	***
sexmale	-0.415658	0.087608	-4.745	2.09e-06	***
school	0.058541	0.011989	4.883	1.05e-06	***
insuranceyes	0.747120	0.100880	7.406	1.30e-13	***

Truncated Poisson to predict non-zero counts

```
Count model coefficients (truncated poisson with log link):
```

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	1.406459	0.024180	58.167	< 2e-16	***
hospital	0.158967	0.006061	26.228	< 2e-16	***
healthexcellent	-0.303677	0.031150	-9.749	< 2e-16	***
healthpoor	0.253521	0.017708	14.317	< 2e-16	***
chronic	0.101720	0.004719	21.557	< 2e-16	***
sexmale	-0.062247	0.013055	-4.768	1.86e-06	***
school	0.019078	0.001872	10.194	< 2e-16	***
insuranceyes	0.080879	0.017139	4.719	2.37e-06	***

The Fitted Equation

Logistic Regression to predict log odds of zero visits...

$$\begin{aligned}\text{logit}(\text{visits} = 0) = & .04 + .31 \text{ hospital} - .29 \text{ health} = \text{Exc} \\ & - .01 \text{ health} = \text{Poor} + .54 \text{ chronic} - .42 \text{ sex} = \text{male} \\ & + .06 \text{ school} + .75 \text{ insurance} = \text{yes}\end{aligned}$$

Truncated¹ Poisson model to predict $\log(\text{visits})$

$$\begin{aligned}\log(\text{visits}) = \max(0, & -1.4 + .15 \text{ hospital} - .30 \text{ health} = \text{Exc} \\ & + .25 \text{ health} = \text{Poor} + .10 \text{ chronic} - .06 \text{ sex} = \text{male} \\ & + .02 \text{ school} + .08 \text{ insurance} = \text{yes})\end{aligned}$$

¹to produce only estimates greater than 0

Confidence Intervals around coefficients

```
> confint(mod_5)
```

	2.5 %	97.5 %
count_(Intercept)	1.35906781	1.453849989
count_hospital	0.14708813	0.170846838
count_healthexcellent	-0.36473054	-0.242623743
count_healthpoor	0.21881501	0.288227575
count_chronic	0.09247176	0.110968673
count_sexmale	-0.08783320	-0.036660471
count_school	0.01541018	0.022746684
count_insuranceyes	0.04728775	0.114470242
zero_(Intercept)	-0.23095748	0.317251001
zero_hospital	0.13323624	0.491660923
zero_healthexcellent	-0.56922139	-0.009919046
zero_healthpoor	-0.32431694	0.306885255
zero_chronic	0.44627272	0.624152555
zero_sexmale	-0.58736685	-0.243949220
zero_school	0.03504270	0.082039766
zero_insuranceyes	0.54939962	0.944840005

Exponentiated Coefficients

```
> exp(coef(mod_5))  
      count_(Intercept)      count_hospital  
      4.0814769          1.1722998  
count_healthexcellent count_healthpoor  
      0.7380991          1.2885548  
      count_chronic      count_sexmale  
      1.1070737          0.9396509  
      count_school count_insuranceyes  
      1.0192616          1.0842397  
      zero_(Intercept)      zero_hospital  
      1.0440911          1.3667677  
zero_healthexcellent zero_healthpoor  
      0.7485852          0.9913220  
      zero_chronic      zero_sexmale  
      1.7078113          0.6599059  
      zero_school zero_insuranceyes  
      1.0602887          2.1109114
```

Exponentiated Confidence Intervals

```
> exp(confint(mod_5))
```

	2.5 %	97.5 %
count_(Intercept)	3.8925630	4.2795591
count_hospital	1.1584561	1.1863090
count_healthexcellent	0.6943837	0.7845667
count_healthpoor	1.2446010	1.3340609
count_chronic	1.0968822	1.1173599
count_sexmale	0.9159136	0.9640034
count_school	1.0155295	1.0230074
count_insuranceyes	1.0484236	1.1212793
zero_(Intercept)	0.7937732	1.3733472
zero_hospital	1.1425199	1.6350296
zero_healthexcellent	0.5659659	0.9901300
zero_healthpoor	0.7230211	1.3591850
zero_chronic	1.5624775	1.8666634
zero_sexmale	0.5557888	0.7835274
zero_school	1.0356639	1.0854990
zero_insuranceyes	1.7322127	2.5724018

Two Specific Variables

after exponentiation...

Coefficient	Logistic	Truncated Poisson
chronic	1.71 (1.56, 1.87)	1.11 (1.10, 1.12)
sex = male	0.66 (0.56, 0.78)	0.94 (0.92, 0.96)

Comparison to ZIP model

```
vuong(mod_3, mod_5)
```

Vuong Non-Nested Hypothesis Test-Statistic:

(test-statistic is asymptotically distributed $N(0,1)$ under the null that the models are indistinguishable)

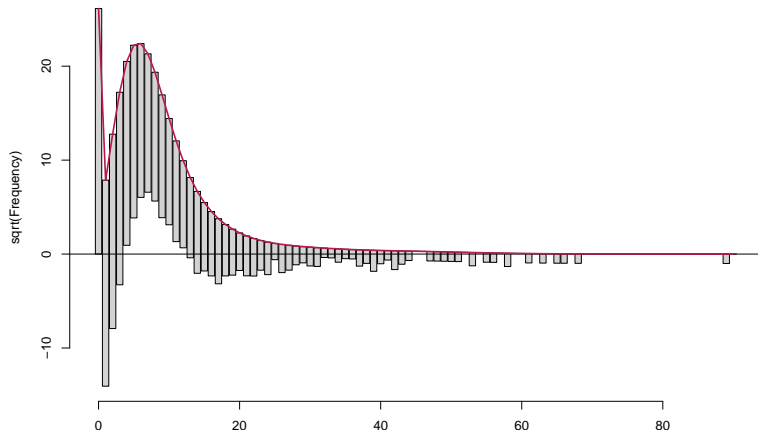
	Vuong z-statistic	H_A	p-value
Raw	2.190241 model1 > model2		0.014253
AIC-corrected	2.190241 model1 > model2		0.014253
BIC-corrected	2.190241 model1 > model2		0.014253

- Looks like the ZIP model may fit a little better than this Hurdle model.

Rootogram for Hurdle/Poisson

```
countreg::rootogram(mod_5, max = 90,  
  main = "Hurdle/Poisson Rootogram: mod_5")
```

Hurdle/Poisson Rootogram: mod_5



Save Fitted Values and Residuals

```
mod_5_aug <- medicare %>%  
  mutate(fitted = fitted(mod_5, type = "response"),  
         resid = resid(mod_5, type = "response"))
```

```
mod_5_aug %>%  
  dplyr::select(visits, fitted, resid) %>%  
  head(2)
```

```
# A tibble: 2 x 3  
  visits fitted  resid  
  <int>   <dbl> <dbl>  
1      5    5.98 -0.981  
2      1    6.05 -5.05
```

Pseudo- R^2 for Hurdle/Poisson model (mod_5)

Squared correlation of the fitted values and the observed values.

```
mod5_r <- with(mod_5_aug, cor(visits, fitted))  
mod5_r^2
```

```
[1] 0.1073668
```

Model 6: The Hurdle Model (Negative Binomial)

Hurdle Model (Negative Binomial-Logistic)

```
mod_6 <- hurdle(visits ~ hospital + health + chronic +  
                sex + school + insurance,  
                dist = "negbin", zero.dist = "binomial",  
                data = medicare)
```

Comparison to ZINB model

```
vuong(mod_4, mod_6)
```

Vuong Non-Nested Hypothesis Test-Statistic:

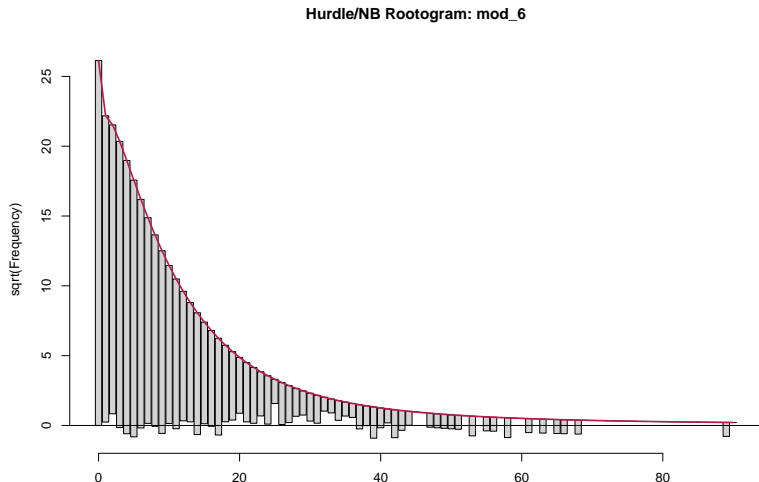
(test-statistic is asymptotically distributed $N(0,1)$ under the null that the models are indistinguishable)

	Vuong z-statistic	H_A	p-value
Raw	-0.391012	model2 > model1	0.34789
AIC-corrected	-0.391012	model2 > model1	0.34789
BIC-corrected	-0.391012	model2 > model1	0.34789

- No significant difference between ZINB and this Hurdle model.

Rootogram for Hurdle/Negative Binomial

```
countreg::rootogram(mod_6, max = 90,  
  main = "Hurdle/NB Rootogram: mod_6")
```



Save Fitted Values and Residuals

```
mod_6_aug <- medicare %>%  
  mutate(fitted = fitted(mod_6, type = "response"),  
         resid = resid(mod_6, type = "response"))
```

```
mod_6_aug %>%  
  dplyr::select(visits, fitted, resid) %>%  
  head(2)
```

```
# A tibble: 2 x 3  
  visits fitted resid  
  <int>   <dbl> <dbl>  
1      5    6.07 -1.07  
2      1    5.94 -4.94
```

Pseudo- R^2 for Hurdle/NB model (mod_6)

Squared correlation of the fitted values and the observed values.

```
mod6_r <- with(mod_6_aug, cor(visits, fitted))  
mod6_r^2
```

```
[1] 0.09223186
```

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	Not good.	0's fine, not enough 1-3.
ZINB	0.096	Better.	Zeros not a perfect fit.
Hurdle (P)	0.107	Like ZIP	Not enough 1-3.
Hurdle (NB)	0.092	Like ZINB	Exact on 0.

Model 7: The Tobit (Censored Regression) Model

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.)

- Here, we might think of a broader latent (unobserved) variable that describes good health.
- We have censoring from below (at 0) where a person with good health (or better) has value 0.
- All of the people with better-than-good health take the same value (0) for visits.

The tobit model postulates that the value 0 in our model is just the lower limit of the underlying measure of poor physical health that we would actually observe in the population if we had a stronger measure.

Fitting the Tobit Model (uses VGAM::vglm)

```
mod_7 <- vglm(visits ~ hospital + health + chronic +  
              sex + school + insurance,  
              tobit(Lower = 0),  
              type.fitted = "censored",  
              data = medicare)
```

Summary of Model 7

```
> summary(mod_7)

Call:
vglm(formula = visits ~ hospital + health + chronic + sex + school +
      insurance, family = tobit(Lower = 0), data = medicare, type.fitted = "censored")

Pearson residuals:
      Min       1Q   Median       3Q      Max
mu      -12.188  -0.4716   0.04917  0.53947   9.892
loge(sd)  -1.013  -0.8205  -0.66567  0.08536  74.700

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1  -0.33163    0.39892  -0.831   0.406
(Intercept):2   1.96495    0.01306 150.484 < 2e-16 ***
hospital        1.73811    0.15121  11.495 < 2e-16 ***
healthexcellent -1.73968    0.43529  -3.997 6.42e-05 ***
healthpoor      1.91404    0.35897   5.332 9.71e-08 ***
chronic         1.22040    0.08913  13.693 < 2e-16 ***
sexmale        -0.93035    0.22770  -4.086 4.39e-05 ***
school          0.18937    0.03203   5.912 3.38e-09 ***
insuranceyes    1.68072    0.28739   5.848 4.97e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of linear predictors: 2

Names of linear predictors: mu, loge(sd)

Log-likelihood: -13198.82 on 8803 degrees of freedom

Number of iterations: 7

No Hauck-Donner effect found in any of the estimates
```

Detailed Coefficient Summary

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept):1	-0.33163	0.39892	-0.831	0.406	
(Intercept):2	1.96495	0.01306	150.484	< 2e-16	***
hospital	1.73811	0.15121	11.495	< 2e-16	***
healthexcellent	-1.73968	0.43529	-3.997	6.42e-05	***
healthpoor	1.91404	0.35897	5.332	9.71e-08	***
chronic	1.22040	0.08913	13.693	< 2e-16	***
sexmale	-0.93035	0.22770	-4.086	4.39e-05	***
school	0.18937	0.03203	5.912	3.38e-09	***
insuranceyes	1.68072	0.28739	5.848	4.97e-09	***

Confidence Intervals for Coefficients

```
> confint(mod_7)
```

	2.5 %	97.5 %
(Intercept):1	-1.113509	0.4502458
(Intercept):2	1.939359	1.9905434
hospital	1.441756	2.0344698
healthexcellent	-2.592820	-0.8865337
healthpoor	1.210463	2.6176118
chronic	1.045715	1.3950864
sexmale	-1.376642	-0.4840553
school	0.126586	0.2521463
insuranceyes	1.117443	2.2440013

Fitted Equation

Using the `type.fitted = "censored"` approach, we'll get predictions limited to visit counts of 0 and larger. If the model below yields predicted visits < 0 , we will fit 0. The model equation is:

$$\begin{aligned} \text{visits} = & -0.33 + 1.74 \text{ hospital} - 1.74 \text{ health} = \text{Excellent} \\ & + 1.91 \text{ health} = \text{Poor} + 1.22 \text{ chronic} - 0.93 \text{ sex} = \text{M} \\ & + 0.19 \text{ school} + 1.68 \text{ insurance} = \text{yes} \end{aligned}$$

Tobit model regression coefficients are interpreted as we would a set of OLS coefficients, except that the linear effect is on the **uncensored latent variable**, rather than on the observed outcome.

Save Fitted Values and Residuals

```
mod_7_aug <- medicare %>%  
  mutate(fitted = fitted(mod_7, type.fitted = "censored"),  
         resid = visits - fitted)
```

```
mod_7_aug %>%  
  dplyr::select(visits, fitted, resid) %>%  
  head(2)
```

```
# A tibble: 2 x 3  
  visits fitted  resid  
  <int>   <dbl>  <dbl>  
1      5    5.73 -0.734  
2      1    5.68 -4.68
```

Rootogram? Not really. Table?

```
table(mod_7_aug$visits, round(mod_7_aug$fitted,0))
```

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	67	71	104	136	117	88	42	20	15	7	7	4	0
1	21	29	59	85	111	77	42	26	14	3	4	2	2
2	10	20	52	68	113	66	43	30	9	7	4	2	3
3	7	25	40	75	82	66	55	29	14	11	6	5	2
4	7	10	32	57	69	65	55	34	22	19	8	0	2
5	4	3	23	41	63	78	42	40	14	9	7	7	3
6	2	3	15	34	54	48	43	27	15	13	7	4	1
7	3	1	7	16	30	40	46	18	22	15	5	3	6
8	4	6	8	12	26	28	38	22	15	6	4	5	4
9	1	2	4	16	19	32	27	21	18	15	7	7	1
10	0	2	9	9	19	24	19	9	13	11	1	6	2
11	2	1	4	17	14	19	16	15	5	6	6	4	1
12	0	1	5	6	9	14	14	15	9	2	1	4	3

Tables of Observed and Fitted visits from Tobit

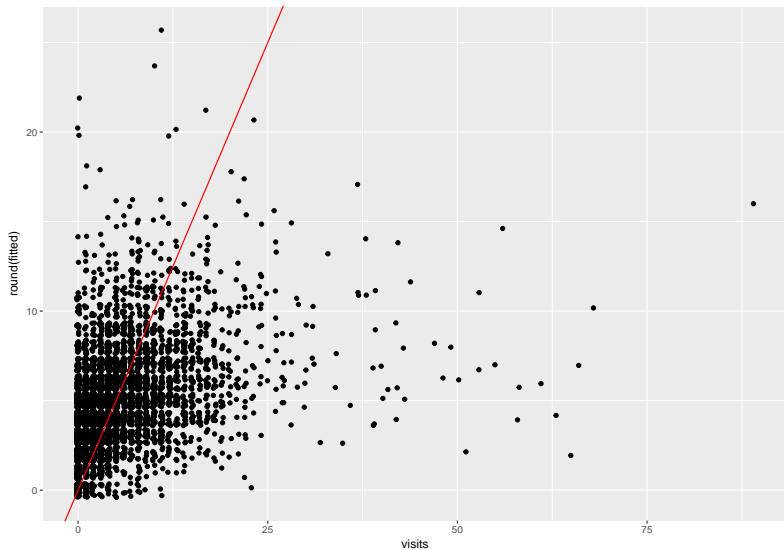
```
addmargins(table(round(mod_7_aug$fitted,0)))
```

0	1	2	3	4	5	6	7	8	9	10	11
129	179	381	593	784	704	563	368	235	162	101	79
12	13	14	15	16	17	18	20	21	22	24	26
44	28	18	15	8	3	3	4	2	1	1	1
Sum											
4406											

```
addmargins(table(mod_7_aug$visits))
```

0	1	2	3	4	5	6	7	8	9	10	11
683	481	428	420	383	338	268	217	188	171	128	115
12	13	14	15	16	17	18	19	20	21	22	23
86	73	76	53	47	48	30	24	16	18	16	10

Plot?



Pseudo- R^2 for Tobit model (mod_7)

Squared correlation of the fitted values and the observed values.

```
mod7_r <- with(mod_7_aug, cor(visits, fitted))  
mod7_r2
```

```
      [,1]  
[1,] 0.1242918
```