Modelling Count Variables

MA4128 - Spring 2016

Overview

1. Introduction to Modelling Count Variables
2. Poisson Regression
3. Negative Binomial Regression
4. Zero-Inflated Models and Vuong Tests
5. Zero Truncation

PART 1: Introduction to Modelling Count Variables

Introduction

- This presentation is about regression methods in which the dependent variable takes count (nonnegative integer) values.
- The dependent variable is usually the number of times an event occurs in a certain period of time.

- Linear regression is used to model and predict continuous measurement variables.
- Poisson regression is used to model and predict discrete count variables.

Poisson regression assumes the response variable Y has a Poisson distribution, and assumes the logarithm of its expected value can be modeled by a linear combination of unknown parameters. A Poisson regression model is sometimes known as a log-linear model, especially when used to model contingency tables.

Overview

Some examples of **event counts** are:

- number of claims per year on a particular car owners insurance policy,
- number of workdays missed due to sickness of a dependent in a one-year period,
- number of papers published per year by a researcher.

Poisson Distribution

- The number of persons killed by mule or horse kicks in the Prussian army per year.
- Ladislaus Bortkiewicz collected data from 20 volumes of Preussischen Statistik.
- These data were collected on 10 corps of the Prussian army in the late 1800s over the course of 20 years, giving a total of 200 observations of one corps for a one year period.
- The unit period of observation is thus one year.

Poisson Distribution: Prussian Cavalary

- The total deaths from horse kicks were 122, and the average number of deaths per year per corps was thus $122/200 = 0.61$.
- In any given year, we expect to observe, well, not exactly 0.61 deaths in one corps
- Here, then, is the classic Poisson situation: a rare event, whose average rate is small, with observations made over many small intervals of time.

Generating Random Numbers

```
> X <- rpois(200,lambda=0.61)
> X
[1]   1 2 0 1 0 3 0 0 1 0 0 4 0 0 0 1 0 1 0 2
[21]  0 0 0 2 2 0 0 0 1 0 0 0 0 1 0 0 0 1 2 0
[41]  0 0 1 0 1 0 1 0 0 1 1 0 1 0 0 1 0 0 3 1
.......
[141] 0 0 0 0 1 2 0 1 0 1 0 0 0 0 0 0 0 1 0 0
[161] 1 0 1 0 0 0 0 1 0 0 0 0 0 1 1 1 0 2 0 1
[181] 0 0 2 0 2 0 0 1 0 0 3 1 0 0 0 1 1 0 0 0
>
> mean(X) ;var(X)
[1] 0.53
[1] 0.5317588
```

Poisson Distribution Assumptions

- Poisson Regression is main technique used to model count variables.
- Assumption underlying Poisson Distribution : Mean and Variance are equal

$$\mathrm{E}(X) = \mathrm{Var}(X)$$

- Allow for a margin of error of about 5% . Simulation Studies can be used to determine the validity of this assumption. (see Next Slide)

Simulation Studies

```
> X=rpois(1000,lambda=1);mean(X);var(X)
[1] 1.001
[1] 1.028027
> X=rpois(5000,lambda=0.5);mean(X);var(X)
[1] 0.5074
[1] 0.5232499
> X=rpois(2500,lambda=0.7);mean(X);var(X)
[1] 0.7248
[1] 0.7317577
> X=rpois(500,lambda=3);mean(X);var(X)
[1] 3.076
[1] 2.851928
```

Simulation Studies

```
> Ratio = numeric()
> M = 10000
>
> for ( i in 1:M){
+       X=rpois(2500,lambda=5);
+       Ratio[i] = var(X)/mean(X)
+ }
>
> quantile(Ratio, c(0.025,0.975))
     2.5%     97.5%
0.9452617 1.0563994
```

Problem Areas

Over-Dispersion : Important Poisson Distributon assumption does not hold

$$\mathrm{E}(X) < \mathrm{Var}(X)$$

Zero-Inflation : More “Zeros” would occure than in conventional Poisson Process (This is actually ”overdispersion” also, but we will treat them separately).

Zero-Truncation : Process does not allow for a “Zero” outcome.

Over-Dispersion

- Overdispersion is the presence of greater variability in a data set than would be expected based on a given simple statistical model.
- Violation of Poisson Distribution Assumption:

$$\mathrm{Var}(X) > \mathrm{E}(X)$$

Zero-Inflation

- One common cause of over-dispersion is excess zeros, which in turn are generated by an additional data generating process.
- In this situation, zero-inflated model should be considered.
- If the data generating process does not allow for any 0s (such as the number of days spent in the hospital), then a zero-truncated model may be more appropriate.

Over-Dispersion

- When there seems to be an issue of dispersion, we should first check if our model is appropriately specified, such as omitted variables and functional forms.
- For example, if we omitted the predictor variable prog in the example above, our model would seem to have a problem with over-dispersion.
- In other words, a misspecified model could present a symptom like an over-dispersion problem.

Poisson Regression with R

- Assuming that the model is correctly specified, the assumption that the conditional variance is equal to the conditional mean should be checked.
- There are several tests including the likelihood ratio test of over-dispersion parameter alpha by running the same model using negative binomial distribution.

Generalized Linear Models

The `glm()` function

- In statistics, the problem of modelling count variables is an example of generalized linear modelling.
- Generalized linear models are fit using the `glm()` function.
- The form of the `glm` function is

```
glm( modelformula,
       family=familytype(link=linkfunction),
       data=dataname)
```

Generalized Linear Models

Family	Default Link Function
binomial	`(link = "logit")`
gaussian	`(link = "identity")`
Gamma	`(link = "inverse")`
inverse.gaussian	(link = "$1/mu^2$")
poisson	`(link = "log")`
quasibinomial	`(link = "logit")`
quasipoisson	`(link = "log")`

Generalized Linear Models

Texts on GLMs

- Dobson, A. J. (1990) An Introduction to Generalized Linear Models. (*London: Chapman and Hall.*)
- Hastie, T. J. and Pregibon, D. (1992) Generalized linear models. Chapter 6 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.
- McCullagh P. and Nelder, J. A. (1989) Generalized Linear Models. (*London: Chapman and Hall.*)
- Venables, W. N. and Ripley, B. D. (2002) Modern Applied Statistics with S. *New York: Springer.*

pscl: Political Science Computational Laboratory

Author(s): Simon Jackman et al (Stanford University)
URL: http://pscl.stanford.edu/

Description
Bayesian analysis of item-response theory (IRT) models, roll call analysis; computing highest density regions; maximum likelihood estimation of zero-inflated and hurdle models for count data; goodness-of-fit measures for GLMs; data sets used in writing and teaching at the Political Science Computational Laboratory; seats-votes curves.

glm2: Fitting Generalized Linear Models

Author(s): Ian Marschner

Fits generalized linear models using the same model specification as glm in the stats package, but with a modified default fitting method that provides greater stability for models that may fail to converge using glm

VGAM: Vector Generalized Linear and Additive Models

Author(s): Thomas W. Yee (t.yee@auckland.ac.nz)

URL: http://www.stat.auckland.ac.nz/$\sim$ yee/VGAM

Vector generalized linear and additive models, and associated models (Reduced-Rank VGLMs, Quadratic RR-VGLMs, Reduced-Rank VGAMs).

This package fits many models and distribution by maximum likelihood estimation (MLE) or penalized MLE. Also fits constrained ordination models in ecology.

MA4128 - Review Questions

(i) Describe Event Counts / Count Variables.

(ii) Poisson Distribution : Asummption of Parameter Equality what is it? how to check?

(iii) State the three cases where assumption does not hold

You are not required to know anything about `R` implementation.

PART 2: Poisson Regression

- Poisson regression is used to model count variables.
- Poisson regression has a number of extensions useful for count models.

Conventional OLS regression

- Count outcome variables are sometimes log-transformed and analyzed using OLS regression.
- Many issues arise with this approach, including loss of data due to undefined values generated by taking the log of zero (which is undefined) and biased estimates.

Poisson Regression with R

If $\mathbf{x} \in \mathbb{R}^n$ is a vector of independent variables, then the model takes the form

$$\log_e(\mathrm{E}(Y \mid \mathbf{x})) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

$$\mathrm{E}(Y \mid \mathbf{x}) = e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n}$$

$$\mathrm{E}(Y \mid \mathbf{x}) = e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n}$$

$$\mathrm{E}(Y \mid \mathbf{x}) = e^{\beta_0} \times e^{\beta_1 x_1} \times \ldots \times e^{\beta_n x_n}$$

Poisson Regression : Crabs Example

The Crabs Data Set

The crabs data set is derived from Agresti *(2007, Table 3.2, pp.76-77)*. It gives 4 variables for each of 173 female horseshoe crabs.

- **Satellites** number of male partners in addition to the female's primary partner
- **Width** width of the female in centimeters
- **Dark** a binary factor indicating whether the female has dark coloring (yes or no)
- **GoodSpine** a binary factor indicating whether the female has good spine condition (yes or no)

Let the first variable be a response variable, with the other three as predictors.

Poisson Regression : Crabs Example

The data is containted in the `R` package **glm2**

```
require(glm2)

data(crabs)
head(crabs)

summary(crabs[,1:4])
```

Poisson Regression : Crabs Example

```
> head(crabs)
  Satellites  Width  Dark GoodSpine  Rep1 Rep2
1          8   28.3   no         no     2    2
2          0   22.5  yes         no     4    5
3          9   26.0   no        yes     5    6
4          0   24.8  yes         no     6    6
5          4   26.0  yes         no     6    8
....
```

Poisson Regression : Crabs Example

```
> summary(crabs[,1:4])
 Satellites          Width           Dark      GoodSpine
 Min.   : 0.000   Min.   :21.0    no :107    no :121
 1st Qu.: 0.000   1st Qu.:24.9    yes: 66    yes: 52
 Median : 2.000   Median :26.1
 Mean   : 2.919   Mean   :26.3
 3rd Qu.: 5.000   3rd Qu.:27.7
 Max.   :15.000   Max.   :33.5
```

Poisson Regression : Crabs Example

- Fit a Poisson regression model with the number of Satellites as the outcome and the width of the female as the covariate.
- What is the multiplicative change in the expected number of crabs for each additional centimeter of width?

```
crabs.pois <- glm2(Satellites ~ Width,
data=crabs, family="poisson")
summary(crabs.pois)

exp(0.164)
```

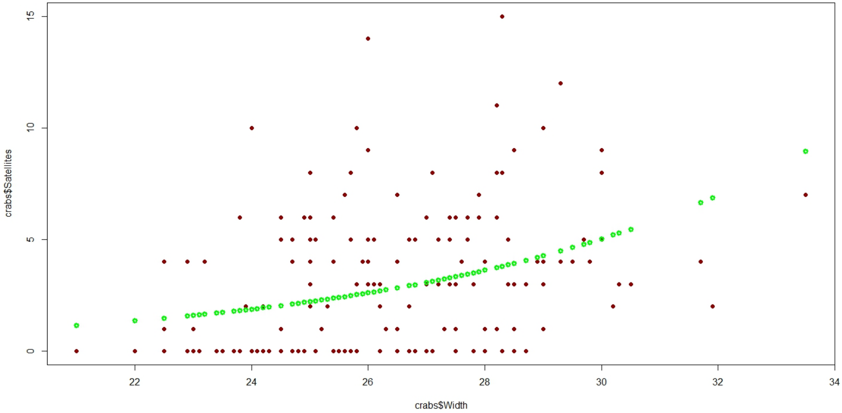
Poisson Regression : Crabs Example

```
> summary(crabs.pois)

Call:
glm2(formula = Satellites ~ Width,
family = "poisson", data = crabs)

......
......
Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.30476    0.54224  -6.095  1.1e-09 ***
Width        0.16405    0.01997   8.216  < 2e-16 ***
---
......
```

Poisson Regression : Crabs Example



Poisson Regression : Crabs Example

Code for Crabs Data Plot

```
plot(crabs$Width, crabs$Satellites,
   pch=16, col="darkred")
   points(crabs$Width, crabs.pois$fitted.values,
   col="green", lwd=3)
```

Other Examples of Poisson regression

- The number of awards earned by students at a secondary or high school.
- Predictors of the number of awards earned include the type of program in which the student was enrolled (e.g., vocational, general or academic) and the score on their final exam in math.

Description of the data

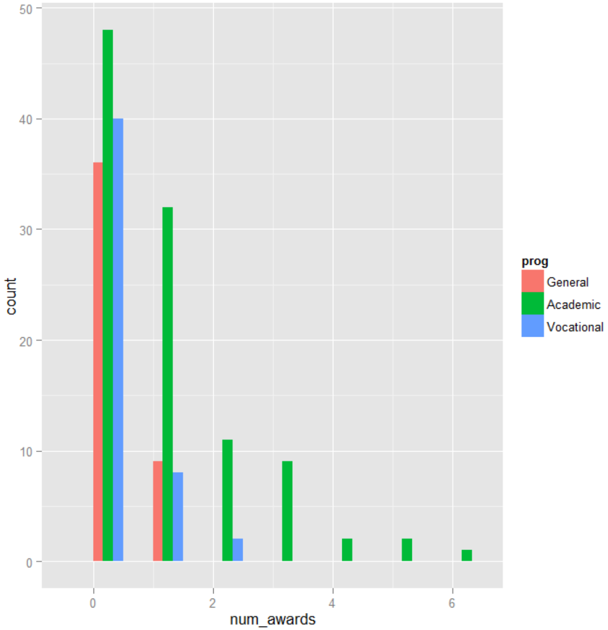
- For the purpose of illustration, we have simulated a data set for the last example.
- The data set is called *poisreg.csv*
- In this example, **num_awards** is the outcome variable and indicates the number of awards earned by students at a high school in a year.

Predictor Variables

- **math** is a continuous predictor variable and represents students' scores on their math final exam,
- **prog** is a categorical predictor variable with three levels indicating the type of program in which the students were enrolled.
- **prog** is coded as $1 = $ "*General*", $2 = $ "*Academic*" and $3 = $ "*Vocational*".

Poisson Regression with R

```
       id          num_awards            prog           math
 1      :  1   Min.   :0.00   General   : 45   Min.   :33.0
 2      :  1   1st Qu.:0.00   Academic  :105   1st Qu.:45.0
 3      :  1   Median :0.00   Vocational: 50   Median :52.0
 4      :  1   Mean   :0.63                    Mean   :52.6
 5      :  1   3rd Qu.:1.00                    3rd Qu.:59.0
 6      :  1   Max.   :6.00                    Max.   :75.0
 (Other):194
```



50
40
30
20
10
0
count
0
2
4
6
num_awards
prog
General
Academic
Vocational

Poisson Regression with R

- Each variable has 200 valid observations and their distributions seem quite reasonable.
- The mean and variance of our outcome variable are more or less the same.
- Our model assumes that these values, conditioned on the predictor variables, will be equal (or at least roughly so).

Poisson regression

- At this point, we are ready to perform our Poisson regression model analysis using the `glm()` function.
- We fit the model and save it in the object `model1` and get a summary of the model.

Poisson Regression with R

```
model1 <- glm(num_awards ~ prog + math,
family="poisson", data=poisreg)

summary(model1)
```

Poisson Regression with R

```
Call:
glm(formula = num_awards ~ prog + math,
      family = "poisson",
      data = poisreg)

Deviance Residuals:
Min        1Q   Median       3Q      Max
-2.204   -0.844   -0.511   0.256   2.680
```

Poisson Regression with R

```
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     -5.2471     0.6585   -7.97  1.6e-15 ***
progAcademic     1.0839     0.3583    3.03   0.0025 **
progVocational   0.3698     0.4411    0.84   0.4018
math             0.0702     0.0106    6.62  3.6e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Poisson Regression with R

```
(Dispersion parameter for poisson family taken to be 1)

Null deviance: 287.67  on 199  degrees of freedom
Residual deviance: 189.45  on 196  degrees of freedom
AIC: 373.5

Number of Fisher Scoring iterations: 6
```

Poisson Regression with R

Regression Coefficients

- Intercept $\beta_0 = -5.2471$
- progAcademic $\beta_1 = 1.0839$
- progVocational $\beta_2 = 0.3698$
- math $\beta_3 = 0.0702$

Exercise

Predict number of awards for Vocational Student with a maths mark of 70.

$$\hat{Y} = e^{-5.2471} \times e^{1.0839 \times 0} \times e^{0.3698 \times 1} \times e^{0.0702 \times 70} = e^{0.0367} = 1.0373$$

We expect the student to win 1 award.

MA4128 Review

(i) Based on `R` output, be able to carry out calculations similar to that in previous slide.

PART 3: Negative Binomial Regression

Negative Binomial Regression is for modeling count outcome variables, when over-dispersion is detected.

Negative Binomial regression with R

- Negative binomial regression can be used for over-dispersed count data, that is when the conditional variance exceeds the conditional mean.

$$\mathrm{Var}(X) > \mathrm{E}(X)$$

- It can be considered as a generalization of Poisson regression since it has the same mean structure as Poisson regression and it has an extra parameter (r) to model the over-dispersion.

$$\Pr(X = k) = \binom{k + r - 1}{k} p^k (1 - p)^r \quad \text{for } k = 0, 1, 2, \ldots$$

Negative Binomial Regression with R

Examples of negative binomial regression

- **Example 1** School administrators study the attendance behavior of high school juniors at two schools. Predictors of the number of days of absence include the type of program in which the student is enrolled and a standardized test in math.
- **Example 2** A health-related researcher is studying the number of hospital visits in past 12 months by senior citizens in a community based on the characteristics of the individuals and the types of health plans under which each one is covered.

Description of the data

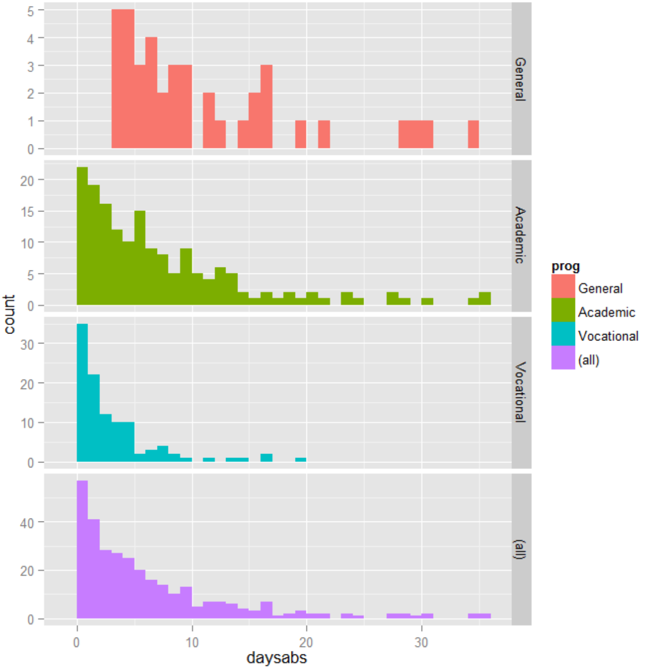
Let's pursue Example 1 from above.

- We have attendance data on 314 high school juniors from two urban high schools in the file **negbin.csv** .
- The response variable of interest is days absent, **daysabs**.
- The variable **math** gives the standardized math score for each student.
- The variable **prog** is a three-level nominal variable indicating the type of instructional program in which the student is enrolled.

Negative Binomial Regression with R

Exploratory Data Analysis

```
summary(dat)
        id            gender           math           daysabs      
 1001    :  1   female:160   Min.   : 1.0   Min.   : 0.00  
 1002    :  1   male  :154   1st Qu.:28.0   1st Qu.: 1.00  
 1003    :  1                Median :48.0   Median : 4.00  
 1004    :  1                Mean   :48.3   Mean   : 5.96  
 1005    :  1                3rd Qu.:70.0   3rd Qu.: 8.00  
 1006    :  1                Max.   :99.0   Max.   :35.00  
 (Other):308                                               
         prog    
 General   : 40  
 Academic  :167  
 Vocational:107  
```



count
daysabs
0
10
20
30
General
Academic
Vocational
(all)
prog
General
Academic
Vocational
(all)

Data Set

- The table on next slide shows the average numbers of days absent by program type and seems to suggest that program type is a good candidate for predicting the number of days absent, our outcome variable, because the mean value of the outcome appears to vary by **prog**.

Negative Binomial Regression with R

Programme	Mean Days Absent	Std.Deviation Days Absent
General	10.65	8.20
Academic	6.95	7.45
Vocational	2.67	3.73

Data Set

- The variances within each level of **prog** are higher than the means within some of the levels.
- These are the conditional means and variances. These differences suggest that over-dispersion is present and that a Negative Binomial model would be appropriate.

Negative Binomial Regression with R

Negative binomial regression analysis

We will use the glm.nb function from the MASS package to estimate a negative binomial regression.

```
summary(m1 <- glm.nb(daysabs ~ math + prog,
 data = negbinom))
##
## Call:
## glm.nb(formula = daysabs ~ math + prog,
        data = dat, init.theta = 1.032713156,
##      link = log)
##
```

Negative Binomial Regression with R

- R first displays the call and the deviance residuals.
- Next, we see the regression coefficients for each of the variables, along with standard errors, z-scores, and p-values.

Negative Binomial Regression with R

```
Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     2.61527    0.19746   13.24  < 2e-16 ***
math           -0.00599    0.00251   -2.39    0.017 *
progAcademic   -0.44076    0.18261   -2.41    0.016 *
progVocational -1.27865    0.20072   -6.37  1.9e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

Negative Binomial Regression with R

- The variable **math** has a coefficient of -0.006, which is statistically significant.
- This means that for each one-unit increase in **math**, the expected log count of the number of days absent decreases by 0.006.
- The indicator variable shown as **progAcademic** is the expected difference in log count between group 2 and the reference group (prog=1).

Negative Binomial Regression with R

- The expected log count for level 2 of prog is 0.44 lower than the expected log count for level 1.
- The indicator variable for **progVocational** is the expected difference in log count between group 3 and the reference group.

Negative Binomial Regression with R

- The expected log count for level 3 of prog is 1.28 lower than the expected log count for level 1.
- To determine if prog itself, overall, is statistically significant, we can compare a model with and without **prog**.
- The reason it is important to fit separate models, is that unless we do, the overdispersion parameter is held constant.

Negative Binomial Regression with R

```
m2 <- update(m1, . ~ . - prog)
anova(m1, m2)
## Likelihood ratio tests of Negative Binomial Models
##
## Response: daysabs
##          Model  theta Resid. df    2 x log-lik.   Test
## 1         math 0.8559       312           -1776
## 2 math + prog 1.0327       310           -1731 1 vs 2
##     Pr(Chi)
## 1
## 2 1.652e-10
```

Negative Binomial Regression with R

- The two degree-of-freedom chi-square test indicates that prog is a statistically significant predictor of daysabs.
- The null deviance is calculated from an intercept-only model with 313 degrees of freedom.
- Then we see the residual deviance, the deviance from the full model.
- We are also shown the AIC and 2*log likelihood.

Negative Binomial Regression with R

The theta parameter shown is the dispersion parameter.

```
## (Dispersion parameter for Negative Binomial(1.033)
## family taken to be 1)
##
##     Null deviance: 427.54  on 313  degrees of freedom
## Residual deviance: 358.52  on 310  degrees of freedom
## AIC: 1741
##
## Number of Fisher Scoring iterations: 1
##
##               Theta:  1.033
##           Std. Err.:  0.106
##
##  2 x log-likelihood:  -1731.258
```

Confidence Intervals

We can get the confidence intervals for the coefficients by profiling the likelihood function.

```
(est <- cbind(Estimate = coef(m1), confint(m1)))
## Waiting for profiling to be done...
##                  Estimate   2.5 %    97.5 %
## (Intercept)      2.615265  2.2421  3.012936
## math            -0.005993 -0.0109 -0.001067
## progAcademic    -0.440760 -0.8101 -0.092643
## progVocational  -1.278651 -1.6835 -0.890078
```

Incidence Rate Ratios

- We might be interested in looking at incident rate ratios rather than coefficients.
- To do this, we can exponentiate our model coefficients.
- The same applies to the confidence intervals.

Negative Binomial Regression with R

```
exp(est)
##                  Estimate  2.5 %  97.5 %
## (Intercept)       13.6708 9.4127 20.3470
## math               0.9940 0.9892  0.9989
## progAcademic       0.6435 0.4448  0.9115
## progVocational     0.2784 0.1857  0.4106
```

Negative Binomial Regression with R

- The output above indicates that the incident rate for prog $= 2$ is 0.64 times the incident rate for the reference group (prog $= 1$).
- Likewise, the incident rate for prog $= 3$ is 0.28 times the incident rate for the reference group holding the other variables constant.
- The percent change in the incident rate of **daysabs** is a 1% decrease for every unit increase in math.

Negative Binomial Regression with R

- The form of the model equation for negative binomial regression is the same as that for Poisson regression.
- The log of the outcome is predicted with a linear combination of the predictors:

Negative Binomial Regression with R

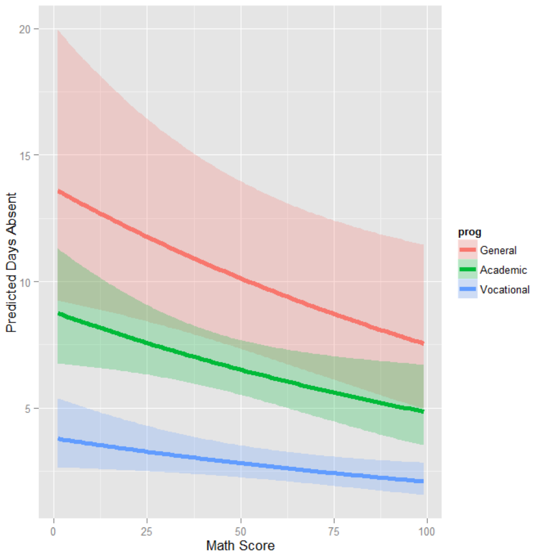
$$\log(\widehat{daysabs_i}) = Intercept + b_1(prog_i = 2) + b_2(prog_i = 3) +$$

$$\widehat{daysabs_i} = e^{Intercept + b_1(prog_i = 2) + b_2(prog_i = 3) + b_3 math_i}$$

$$= e^{Intercept} e^{b_1(prog_i = 2)} e^{b_2(prog_i = 3)} e^{b_3 math_i}$$

Negative Binomial Regression with R

- The coefficients have an additive effect in the $ln(y)$ scale and the IRR have a multiplicative effect in the y scale.
- The dispersion parameter in negative binomial regression does not effect the expected counts, but it does effect the estimated variance of the expected counts.



20
15
10
5
Predicted Days Absent
0
25
50
75
100
Math Score
prog
General
Academic
Vocational

Negative Binomial Regression with R

Things to consider

- It is not recommended that negative binomial models be applied to small samples.
- One common cause of over-dispersion is excess zeros by an additional data generating process. In this situation, **zero-inflated model** should be considered.
- If the data generating process does not allow for any 0s (such as the number of days spent in the hospital), then a **zero-truncated model** may be more appropriate.

Negative Binomial Regression with R

- Count data often have an exposure variable, which indicates the number of times the event could have happened.
- This variable should be incorporated into your negative binomial regression model with the use of the `offset` option.

Negative Binomial Regression with R

- The outcome variable in a negative binomial regression cannot have negative numbers.
- You will need to use the `m1$resid` command to obtain the residuals from our model to check other assumptions of the negative binomial model

Zero-inflated Regression models - Summary

- Zero-inflated models attempt to account for excess zeros.
- In other words, two kinds of zeros are thought to exist in the data, "**true zeros**" and "**excess zeros**".

Zero-inflated Regression models

Two Distinct Processes

- The two parts of the a zero-inflated model are a binary model, usually a logit model to model which of the two processes the zero outcome is associated with and a count model (either a Poisson or negative binomial model) to model the count process.
- In other words, the excess zeros are generated by a separate process from the count values and that the excess zeros can be modelled independently.
- Zero-inflated models estimate two equations simultaneously, one for the count model and one for the excess zeros.
- The expected count is expressed as a combination of the two processes.

Zero-inflated Regression models

Fishing Data Set

- We have data on 250 groups that went to a park.
- Each group was questioned about how many fish they caught (**count**), how many children were in the group (**child**), how many people were in the group (**persons**), and whether or not they brought a camper to the park (**camper**).
- In addition to predicting the number of fish caught, there is interest in predicting the existence of excess zeros, i.e., the probability that a group caught zero fish.

Zero-inflated Regression models

Fishing Data Set

- We will use the variables child, persons, and camper in our model.
- In addition to predicting the number of fish caught, there is interest in predicting the existence of excess zeros, i.e., the probability that a group caught zero fish.
- We will use the variables child, persons, and camper in our model.

```
> head(fish)
  nofish livebait camper persons child         xb
1      1        0      0       1     0 -0.8963146
2      0        1      1       1     0 -0.5583450
3      0        1      0       1     0 -0.4017310
4      0        1      1       2     1 -0.9562981
5      0        1      0       1     0  0.4368910
6      0        1      1       4     2  1.3944855
          zg count
1  3.0504048     0
2  1.7461489     0
3  0.2799389     0
4 -0.6015257     0
5  0.5277091     1
6 -0.7075348     0
```

What is a Zero-Inflated Model?

The Fishing Example

- Recall: A zero-inflated model assumes that zero outcome is due to two different processes.
- For instance, in the example of fishing presented here, the two processes are that a subject has gone 1. *fishing* vs. 2. *not fishing*.
- If not gone fishing, the only outcome possible is zero.
- If gone fishing, it is then a count process.

$$\boxed{\begin{gathered} E(\text{no. fish caught} = k) \\ = P(\text{not fishing}) \times 0 + P(\text{fishing}) \times E(y = k|\text{fishing}) \end{gathered}}$$

The `zeroinfl()` function

The arguments of `zeroinfl()` are given by

```
zeroinfl(formula, data, subset, na.action,
    weights, offset,
 dist = "poisson", link = "logit",
 control = zeroinfl.control(...),
 model = TRUE,
 y = TRUE, x = FALSE, ...)
```

Zero-inflated Poisson regression

Though we can run a Poisson regression in `R` using the `glm` function in one of the core packages, we need another package to run the zero-inflated poisson model. We use the **pscl** package.

```
summary(m1 <- zeroinfl(count ~ child + camper |
     persons, data = zinb))
```

Zero-inflated Poisson regression

```
##
## Call:
## zeroinfl(formula = count ~ child + camper | persons, dat
##
## Pearson residuals:
##     Min      1Q Median      3Q     Max
## -1.237 -0.754 -0.608 -0.192 24.085
```

Zero-inflated Poisson regression

```
## Count model coefficients (poisson with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.5979     0.0855   18.68   <2e-16 ***
## child        -1.0428     0.1000  -10.43   <2e-16 ***
## camper1       0.8340     0.0936    8.91   <2e-16 ***
```

Zero-inflated Poisson regression

```
## Zero-inflation model coefficients (binomial with logit l
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)        1.297      0.374    3.47  0.00052 ***
## persons           -0.564      0.163   -3.46  0.00053 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
## Number of iterations in BFGS optimization: 12
## Log-likelihood: -1.03e+03 on 5 Df
```

Zero-inflated Poisson regression

- Below the model call, you will find a block of output containing Poisson regression coefficients for each of the variables along with standard errors, z-scores, and p-values for the coefficients.
- A second block follows that corresponds to the inflation model.
- This includes logit coefficients for predicting excess zeros along with their standard errors, z-scores, and p-values.

Zero-Inflation

```
confint(m1)
##                         2.5 %  97.5 %
## count_(Intercept)  1.4302  1.7655
## count_child       -1.2388 -0.8469
## count_camper1      0.6505  1.0175
## zero_(Intercept)   0.5647  2.0302
## zero_persons      -0.8838 -0.2449
```

Zero-Inflation

- All of the predictors in both the count and inflation portions of the model are statistically significant.
- We can use other methods (for example non parametric methods) to compute CIs (probably more accurate)

Vuong Testing

- Note that the model output above does not indicate in any way if our zero-inflated model is an improvement over a standard Poisson regression.
- We can determine this by running the corresponding standard Poisson model and then performing a Vuong test of the two models.

```
summary(p1 <- glm(count ~ child + camper,
family = poisson, data = fishing))
```

Vuong Testing

- The Vuong test compares the zero-inflated model with an ordinary Poisson regression model.
- In this example, we can see that our test statistic is significant, indicating that the zero-inflated model is superior to the standard Poisson model.

```
vuong(p1, m1)

## Vuong Non-Nested Hypothesis Test-Statistic: -3.574
##
## (test-statistic is asymptotically distributed N(0,1)
## under the null hypothesis that the models are
## indistinguishible)
## in this case:
##
## model2 > model1, with p-value 0.0001756
```

PART 4B :

Zero-Inflated Negative Binomial Regression

Zero-Inflated Negative Binomial regression

- Zero-inflated negative binomial regression is for modeling count variables with excessive zeros and it is usually for overdispersed count outcome variables.
- Furthermore, theory suggests that the excess zeros are generated by a separate process from the count values and that the excess zeros can be modeled independently.

Zero-Inflated Negative Binomial regression

- We are going to use the variables: **child** and **camper** to model the count in the part of negative binomial model and the variable **persons** in the logit part of the model.
- We use the **pscl** to run a zero-inflated negative binomial regression.
- We begin by estimating the model (called `m1`) with the variables of interest.

Zero-Inflated Negative Binomial regression

```
m1 <- zeroinfl(count ~ child + camper | persons,
  data = fishing, dist = "negbin",
  EM = TRUE)

summary(m1)
```

```
## Call:
## zeroinfl(formula = count ~ child + camper | persons,
##     data = fishing,
##     dist = "negbin", EM = TRUE)
##
## Pearson residuals:
##    Min     1Q Median     3Q    Max
## -0.586 -0.462 -0.389 -0.197 18.013
```

Zero-Inflated Negative Binomial regression

- Below the model call, you will find a block of output containing negative binomial regression coefficients for each of the variables along with standard errors, z-scores, and p-values for the coefficients.
- A second block follows that corresponds to the inflation model.
- This includes logit coefficients for predicting excess zeros along with their standard errors, z-scores, and p-values.

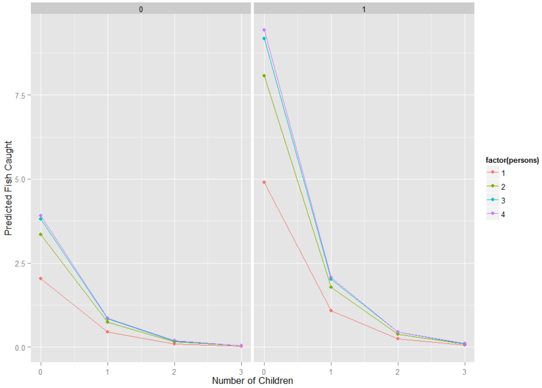
```
## Count model coefficients (negbin with log link):
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.371      0.256    5.35  8.6e-08 ***
## child         -1.515      0.196   -7.75  9.4e-15 ***
## camper1        0.879      0.269    3.26   0.0011 **
## Log(theta)    -0.985      0.176   -5.60  2.1e-08 ***
```

```
## Zero-inflation model coefficients (binomial with logit l
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.603      0.836    1.92    0.055 .
## persons        -1.666      0.679   -2.45    0.014 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
## Theta = 0.373
## Number of iterations in BFGS optimization: 2
## Log-likelihood: -433 on 6 Df
```

- The predictors child and camper in the part of the negative binomial regression model predicting number of fish caught (count) are both significant predictors.
- The predictor person in the part of the logit model predicting excessive zeros is statistically significant.
- For these data, the expected change in log(count) for a one-unit increase in child is -1.515255 holding other variables constant.
- A camper (camper $= 1$) has an expected log(count) of 0.879051 higher than that of a non-camper (camper $= 0$) holding other variables constant.

Tests of Significance

- All of the predictors in both the count and inflation portions of the model are statistically significant.


0
1
Predicted Fish Caught
7.5
5.0
2.5
0.0
0
1
2
3
Number of Children
factor(persons)
1
2
3
4

- The log odds of being an excessive zero would decrease by 1.67 for every additional person in the group.
- In other words, the more people in the group the less likely that the zero would be due to not gone fishing.
- Put plainly, the larger the group the person was in, the more likely that the person went fishing.
- The Vuong test suggests that the zero-inflated negative binomial model is a significant improvement over a standard negative binomial model.

Vuong Testing

- Note that the model output above does not indicate in any way if our zero-inflated model is an improvement over a standard negative binomial regression.
- We can determine this by running the corresponding standard negative binomial model and then performing a Vuong test of the two models.
- We use the MASS package to run the standard negative binomial regression.

Vuong Testing

```
library(MASS)
summary(m2 <- glm.nb(count ~ child + camper, data = zinb))
.....
```

Vuong Testing

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.073      0.242    4.42  9.7e-06 ***
## child         -1.375      0.196   -7.03  2.1e-12 ***
## camper1        0.909      0.284    3.21   0.0013 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

Vuong Testing

```
vuong(m1, m2)

## Vuong Non-Nested Hypothesis Test-Statistic: 1.702
## (test-statistic is asymptotically distributed N(0,1) und
##  null that the models are indistinguishible)
## in this case:
## model1 > model2, with p-value 0.0444
```

- The log odds of being an excessive zero would decrease by 1.67 for every additional person in the group.
- In other words, the more people in the group the less likely that the zero would be due to not gone fishing.
- Put plainly, the larger the group the person was in, the more likely that the person went fishing.
- The Vuong test suggests that the zero-inflated negative binomial model is a significant improvement over a standard negative binomial model.

Zero Truncated Poisson Distribution

Zero-Truncated Poisson Regression

- Zero-truncated Modelling is used to model count data for which the value zero cannot occur.
- Zero Truncated Poisson Model
- Zero Truncated Negative Binomial Model (Over Dispersion)

Examples of Zero-Truncated Model

Example 1.

- A study of length of hospital stay, in days, as a function of age, kind of health insurance and whether or not the patient died while in the hospital.
- Length of hospital stay is recorded as a minimum of at least one day.

Example 2.

- A study of the number of journal articles published by tenured faculty as a function of discipline (fine arts, science, social science, humanities, medical, etc).
- To get tenure faculty must publish, therefore, there are no tenured faculty with zero publications.

Examples of Zero-Truncated Model

Example 3.

- A study by the county traffic court on the number of tickets received by teenagers as predicted by school performance, amount of driver training and gender.
- Only individuals who have received at least one citation are in the traffic court files.

Example 4.

- Consider for example the random variable of the number of items in a shopper's basket at a supermarket checkout line.
- Presumably a shopper does not stand in line with nothing to buy (i.e. the minimum purchase is 1 item), so this phenomenon may follow a ZTP distribution

Zero-Truncated Poisson regression

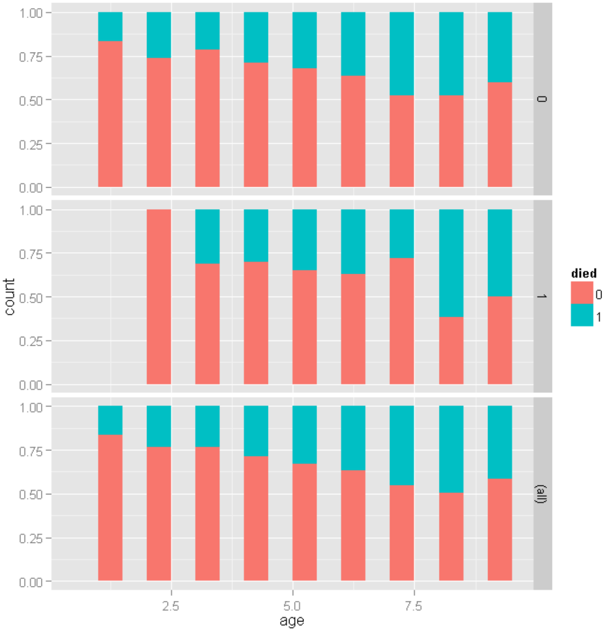
Data Set : hospitalstay

- We have a hypothetical data file, **hospitalstay** with 1,493 observations.
- The length of hospital stay variable is **stay**.
- The variable **age** gives the age group from 1 to 9 which will be treated as interval in this example.
- The variables **hmo** and **died** are binary indicator variables for HMO insured patients and patients who died while in the hospital, respectively.

Zero-Truncated Poisson regression

Data Set : hospitalstay

```
##       stay              age          hmo        died
##  Min.   : 1.00   Min.   :1.00   0:1254   0:981
##  1st Qu.: 4.00   1st Qu.:4.00   1: 239   1:512
##  Median : 8.00   Median :5.00
##  Mean   : 9.73   Mean   :5.23
##  3rd Qu.:13.00   3rd Qu.:6.00
##  Max.   :74.00   Max.   :9.00
```


1.00
0.75
0.50
0.25
0.00
count
2.5
5.0
7.5
age
0
1
(all)
died
0
1

Zero-Truncated Poisson regression

Data Set : hospitalstay

- For the lowest ages, a smaller proportion of people in HMOs died, but for higher ages, there does not seem to be a huge difference, with a slightly higher proportion in HMOs dying if anything.
- Overall, as age group increases, the proportion of those dying increases, as expected.

Zero-Truncated Poisson regression

- To fit the zero-truncated Poisson model, we use the `vglm` function in the VGAM package.
- This function fits a very flexible class of models called **vector generalized linear models** to a wide range of assumed distributions.
- In our case, we believe the data are Poisson, but without zeros.
- Thus the values are strictly positive Poisson, for which we use the positive Poisson family via the `pospoisson` function passed to `vglm`.

Zero-Truncated Poisson regression

Fitting the Model with `R`

We will use the *hospitalstay* data.

```
m1 <- vglm(stay ~ age + hmo + died,
    family = pospoisson(),
    data = hospitalstay)
summary(m1)
```

Zero-Truncated Poisson regression

Fitting the Model with R

Model Summary

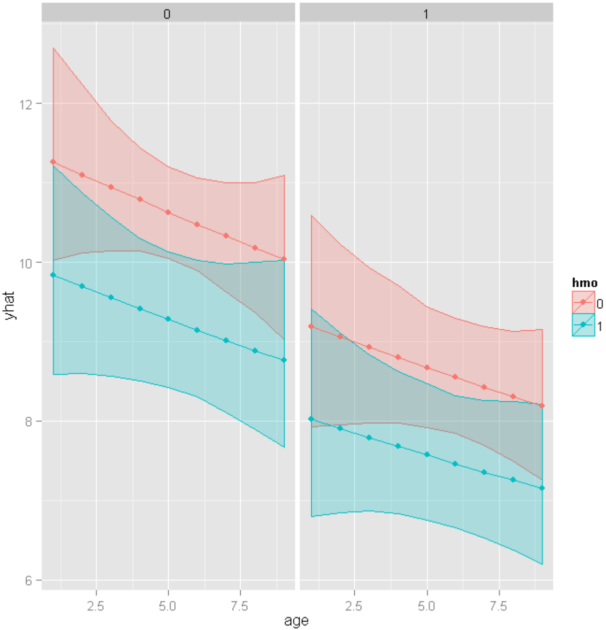
```
## Coefficients:
##               Estimate Std. Error z value
## (Intercept)      2.436      0.027    89.1
## age             -0.014      0.005    -2.9
## hmo1            -0.136      0.024    -5.7
## died1           -0.204      0.018   -11.1
```

Zero-Truncated Poisson regression

- The value of the coefficient for age, -0.0144 suggests that the log count of stay decreases by 0.0144 for each year increase in age.
- The coefficient for hmo, -0.1359 indicates that the log count of stay for HMO patient is 0.1359 less than for non-HMO patients.
- The log count of stay for patients who died while in the hospital was 0.2038 less than those patients who did not die.
- Finally, the value of the constant 2.4358 is the log count of the stay when all of the predictors equal zero.

Zero-Truncated Poisson regression

- Can compute CIs using **boot** package
- Age does not have a significant effect, but hmo and died both do.



0
1
12
10
8
6
2.5
5.0
7.5
2.5
5.0
7.5
yhat
age
hmo
0
1

Zero-truncated negative binomial regression

- Zero-truncated negative binomial regression is used to model count data for which the value zero cannot occur and for which over dispersion exists.

Zero-truncated negative binomial regression

- To fit the zero-truncated negative binomial model, we use the `vglm` function in the VGAM package.
- This function fits a very flexible class of models called vector generalized linear models to a wide range of assumed distributions.
- In our case, we believe the data come from the negative binomial distribution, but without zeros.
- Thus the values are strictly positive poisson, for which we use the positive negative binomial family via the `posnegbinomial` function passed to `vglm`.

Zero-truncated negative binomial regression

Fitting the Model with R

We will use the *hospitalstay* data again.

```
m1 <- vglm(stay ~ age + hmo + died,
   family = posnegbinomial(),
   data = hospitalstay)
```

Zero Truncated Negative Binomial Regression

```
summary(m1)
##
## Call:
## vglm(formula = stay ~ age + hmo + died,
##     family = posnegbinomial(),
##     data = hospitalstay)
##
## Pearson Residuals:
##                Min    1Q Median   3Q Max
## log(munb)   -1.4 -0.70  -0.23 0.45 9.8
## log(size) -14.1 -0.27   0.45 0.76 1.0
```

Zero Truncated Negative Binomial Regression

```
## Coefficients:
##                Estimate Std. Error z value
## (Intercept):1     2.408      0.072    33.6
## (Intercept):2     0.569      0.055    10.4
## age              -0.016      0.013    -1.2
## hmo1             -0.147      0.059    -2.5
## died1            -0.218      0.046    -4.7
```

Zero Truncated Negative Binomial Regression

-
- The first intercept is what we know as the typical intercept.
- The second is the **over dispersion parameter**, α.
- The number of linear predictors is two, one for the expected mean λ and one for the over dispersion.
- Next the dispersion parameter is printed, assumed to be one after accounting for overdispersion.

Zero Truncated Negative Binomial Regression

- The value of the coefficient for age, -0.0157 suggests that the log count of stay decreases by 0.0157 for each year increase in age.
- The coefficient for hmo, -0.1471 indicates that the log count of stay for HMO patient is 0.1471 less than for non-HMO patients.
- The log count of stay for patients who died while in the hospital was 0.2178 less than those patients who did not die.

Zero Truncated Negative Binomial Regression

- The value of the constant 2.4083 is the log count of the stay when all of the predictors equal zero.
- The value of the second intercept, the over dispersion parameter, α is 0.5686.
- To test whether we need to estimate over dispersion, we could fit a zero-truncated Poisson model and compare the two. (Not Covered).