- This talks is about regression methods in which the dependent variable takes nonnegative integer or count values.
- The dependent variable is usually the number of times an event occurs.

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.

Overview

- Poisson regression is used to model count variables.
- Negative Binomial regression is for modelling count variables, usually for over-dispersed count outcome variables.

Poisson Regression

► Poisson regression is used to model count variables.

► Poisson regression has a number of extensions useful for count models.

Examples of Poisson regression

- 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.

Examples of Poisson regression

- The number of people in line in front of you at the grocery store.
- Predictors may include the number of items currently offered at a special discounted price and whether a special event (e.g., a holiday, a big sporting event) is three or fewer days away.

Examples of Poisson regression

- The number of awards earned by students at one 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.

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.

Description of the data

- For the purpose of illustration, we have simulated a data set for the last example.

- The data set is called *poissonreg.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
```

Poisson Regression with R

- ▶ Each variable has 200 valid observations and their distributions seem quite reasonable.
- ▶ The unconditional mean and variance of our outcome variable are not extremely different.
- ▶ Our model assumes that these values, conditioned on the predictor variables, will be equal (or at least roughly so).

Poisson Regression with R

- Additionally, the means and variances within each level of prog–*the conditional means and variances*–are similar.

- A conditional histogram separated out by program type is plotted to show the distribution.

Poisson regression

- ▶ At this point, we are ready to perform our Poisson model analysis using the glm function.
- ▶ We fit the model and store it in the object m1 and get a summary of the model.

Poisson Regression with R

```
m1 <- glm(num_awards ~ prog + math,
  family="poisson", data=p)

summary(m1)
```

Poisson Regression with R

```
Call:
glm(formula = num_awards ~ prog + math, family = "po

 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 '.'
```

Poisson Regression with R

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

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

Number of Fisher Scoring iterations: 6
```

Poisson Regression with R

- ▶ It is recommended using robust standard errors for the parameter estimates to control for mild violation of the distribution assumption that the variance equals the mean.
- ▶ The R package **sandwich** can be used to obtain the robust standard errors and calculated the p-values accordingly.
- ▶ Together with the p-values, we have also calculated the 95% confidence interval using the parameter estimates and their robust standard errors.

Poisson Regression with R

sandwich R Package

- ▶ Robust Covariance Matrix Estimators
- ▶ Model-robust standard error estimators for cross-sectional, time series, and longitudinal data.

Poisson Regression with R

Robust Standard Errors

```
cov.m1 <- vcovHC(m1, type="HC0")
std.err <- sqrt(diag(cov.m1))

r.est <- cbind(Estimate= coef(m1),
  "Robust SE" = std.err,
  "Pr(>|z|)" = 2 * pnorm(abs(coef(m1)/std.err),
  lower.tail=FALSE),
LL = coef(m1) - 1.96 * std.err,
UL = coef(m1) + 1.96 * std.err)
```

Poisson Regression with R

```
r.est

               Estimate Robust SE  Pr(>|z|)        LL
 (Intercept)   -5.24712    0.64600 4.567e-16 -6.5133
 progAcademic   1.08386    0.32105 7.355e-04  0.4546
 progVocational 0.36981    0.40042 3.557e-01 -0.4150
 math           0.07015    0.01044 1.784e-11  0.0497
```

Poisson Regression with R

glm function output

- ▶ The output begins with echoing the function call. The information on deviance residuals is displayed next.

- ▶ Deviance residuals are approximately normally distributed if the model is specified correctly.

- ▶ Here it shows a little bit of skeweness since median is not quite zero.

glm function output

- ▶ The Poisson regression coefficients for each of the variables along with the standard errors, z-scores, p-values and 95% confidence intervals for the coefficients.

- ▶ The coefficient for math is 0.07.

- ▶ This means that the expected log count for a one-unit increase in math is 0.07.

glm function output

- ► The indicator variable **progAcademic** compares between **prog = Academic** and **prog = "General"**, the expected log count for **prog = Academic** increases by about 1.1.

- ► The indicator variable **prog.Vocational** is the expected difference in log count (≈ 0.37) between **prog = "Vocational"** and the reference group (**prog = "General"**).

Deviance

- In statistics, deviance is a quality of fit statistic for a model that is often used for statistical hypothesis testing.
- It is a generalization of the idea of using the sum of squares of residuals in ordinary least squares to cases where model-fitting is achieved by maximum likelihood.

Poisson Regression with R

glm function output

- ▶ The information on deviance is also provided.
- ▶ We can use the residual deviance to perform a goodness of fit test for the overall model.
- ▶ The residual deviance is the difference between the deviance of the current model and the maximum deviance of the ideal model where the predicted values are identical to the observed.

Poisson Regression with R

glm function output

- ▶ Therefore, if the residual difference is small enough, the goodness of fit test will not be significant, indicating that the model fits the data. We conclude that the model fits reasonably well because the goodness-of-fit chi-squared test is not statistically significant.

- ▶ If the test had been statistically significant, it would indicate that the data do not fit the model well.

- ▶ We could try to determine if there are omitted predictor variables, if our linearity assumption holds and/or if there is an issue of

Poisson Regression with R

```
with(m1, cbind(res.deviance = deviance,
df = df.residual,
        p = pchisq(deviance, df.residual,
            lower.tail=FALSE)))

      res.deviance  df       p
 [1,]        189.4 196 0.6182
```

Poisson Regression with R

- We can also test the overall effect of prog by comparing the deviance of the full model with the deviance of the model excluding prog.
- The two degree-of-freedom chi-square test indicates that prog, taken together, is a statistically significant predictor of num_awards.

Poisson Regression with R

```
# update m1 model dropping prog
m2 <- update(m1, . ~ . - prog)
# test model differences with chi square test
anova(m2, m1, test="Chisq")
```

Poisson Regression with R

```
Analysis of Deviance Table

Model 1: num_awards ~ math
Model 2: num_awards ~ prog + math
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       198        204
2       196        189  2     14.6  0.00069 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
```

Incident Rate Ratios

- Sometimes, we might want to present the regression results as **incident rate ratios** (IRRs) and their standard errors, together with the confidence interval.

- To compute the standard error for the incident rate ratios, we will use the **Delta method** (Numerical Computation Method).

- To this end, we make use the function deltamethod implemented in R package **msm**.

Incident Rate Ratios

A rate ratio (sometimes called an incidence density ratio) in epidemiology, is a relative difference measure used to compare the incidence rates of events occurring at any given point in time. A common application for this measure in analytic epidemiologic studies is in the search for a causal association between a certain risk factor and an outcome.[1]

$$\text{Incidence Rate Ratio} = \frac{\text{Incidence Rate 1}}{\text{Incidence Rate 2}}$$

Incident Rate Ratios

Incidence rate is the occurrence of an event over person-time, for example person-years.

$$\text{Incidence Rate} = \frac{\text{events}}{\text{Person Time}}$$

Note: the same time intervals must be used for both incidence rates.

```
s <- deltamethod(list(~ exp(x1), ~ exp(x2),
#exponentiate old estimates dropping the p v
rexp.est <- exp(r.est[, -3])

# replace SEs with estimates for exponentiat
rexp.est[, "Robust SE"] <- s
```

Poisson Regression with R

```
rexp.est

              Estimate Robust SE       LL       UL
 (Intercept)  0.005263   0.00340 0.001484 0.01867
 progAcademic 2.956065   0.94904 1.575551 5.54620
 progVocational 1.447458 0.57959 0.660335 3.17284
 math         1.072672   0.01119 1.050955 1.09484
```

Poisson Regression with R

- ▶ The output above indicates that the incident rate for prog = "Academic" is 2.96 times the incident rate for the reference group (prog = "General").

- ▶ Likewise, the incident rate for prog = "Vocational" is 1.45 times the incident rate for the reference group holding the other variables at constant.

Poisson Regression with R

- The percent change in the incident rate of num_awards is by 7% for every unit increase in math.

Poisson Regression with R

- ▶ Sometimes, we might want to look at the expected marginal means.
- ▶ For example, what are the expected counts for each program type holding math score at its overall mean?
- ▶ To answer this question, we can make use of the predict function.
- ▶ First off, we will make a small data set to apply the predict function to it.

Poisson Regression with R

```
(s1 <- data.frame(math = mean(p$math),
 prog = factor(1:3, levels = 1:3,
 labels = levels(p$prog))))

    math      prog
1 52.65   General
2 52.65  Academic
3 52.65 Vocational
```

Poisson Regression with R

```
predict(m1, s1, type="response", se.fit=TRUE)

 $fit
     1      2      3
0.2114 0.6249 0.3060

 $se.fit
     1      2      3
0.07050 0.08628 0.08834

 $residual.scale
 [1] 1
```
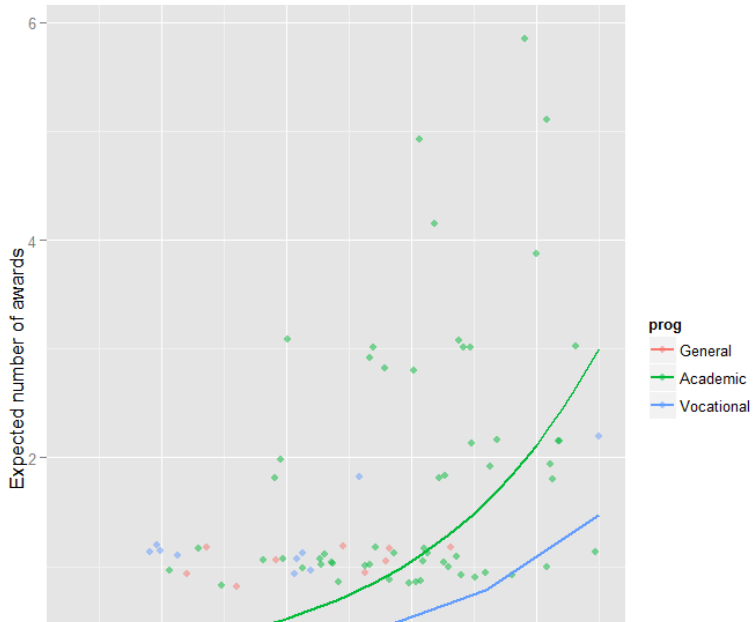
- In the output above, we see that the predicted number of events for level 1 of prog is about 0.21, holding math at its mean.
- The predicted number of events for level 2 of prog is higher at 0.62, and the predicted number of events for level 3 of prog is about .31.
- The ratios of these predicted counts ($\frac{0.625}{0.211} = 2.96$, $\frac{0.306}{0.211} = 1.45$) match what we saw looking at the IRR.

- ► We can also graph the predicted number of events with the commands below.
- ► The graph indicates that the most awards are predicted for those in the academic program (prog $= 2$), especially if the student has a high math score.
- ► The lowest number of predicted awards is for those students in the general program (prog $=$ 1).
- ► The graph overlays the lines of expected values onto the actual points, although a small amount of random noise was added vertically to lessen

Poisson Regression with R

```
# Calculate and store predicted values
p$phat <- predict(m1, type="response")

# order by program and then by math
p <- p[with(p, order(prog, math)), ]
```

Poisson Regression with R

```r
ggplot(p, aes(x = math, y = phat, colour = prog)) +
  geom_point(aes(y = num_awards), alpha=.5, position=
  geom_line(size = 1) +
  labs(x = "Math Score", y = "Expected number of awar
```

Poisson Regression with R

Over-Dispersion Overdispersion is the presence of greater variability (statistical dispersion) in a data set than would be expected based on a given simple statistical model.

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.
- ▶ The R package pscl (Political Science Computational Laboratory, Stanford University) provides many functions for binomial and count data including odTest for testing over-dispersion.

Poisson Regression with R

- 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.

Poisson 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 a Poisson model with the use of the offset option.

- ▶ The outcome variable in a Poisson regression cannot have negative numbers, and the exposure cannot have 0s.

Poisson Regression with R

- ▶ Many different measures of pseudo-R-squared exist. They all attempt to provide information similar to that provided by R-squared in OLS regression, even though none of them can be interpreted exactly as R-squared in OLS regression is interpreted.

- ▶ Poisson regression is estimated via maximum likelihood estimation. It usually requires a large sample size.

Negative Binomial Regression with R

Introduction Negative binomial regression is for modeling count variables, usually for over-dispersed count outcome variables.

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.

- ▶ 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 to model the over-dispersion.

Negative Binomial regression with `R`

- ▸ If the conditional distribution of the outcome variable is over-dispersed, the confidence intervals for Negative binomial regression are likely to be narrower as compared to those from a Poisson regression.

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.

Negative Binomial Regression with `R`

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.

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
```
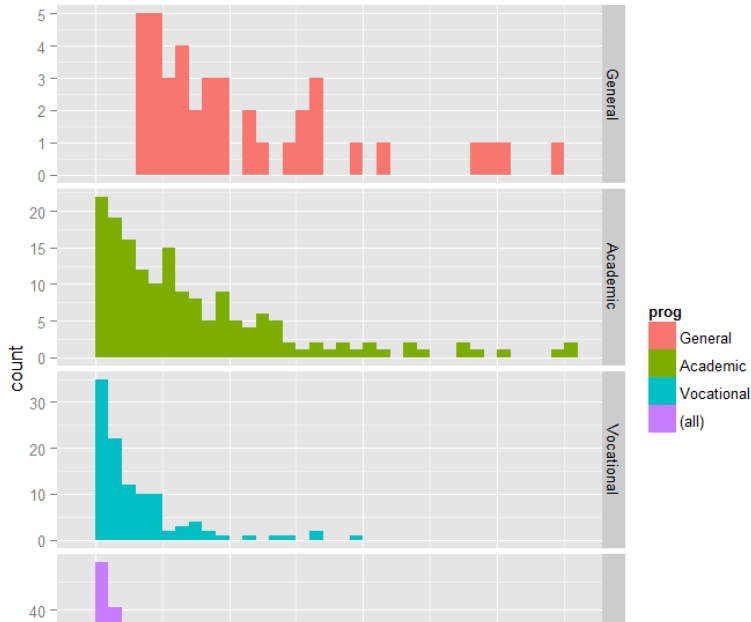
```
ggplot(dat, aes(daysabs, fill = prog)) + geom_
., margins = TRUE, scales = "free")
```

Histogram plots showing distribution of the data
Each variable has 314 valid observations and their
distributions seem quite reasonable. The
unconditional mean of our outcome variable is much
lower than its variance.

Data Set

▶ The table below 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.

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

```
with(dat, tapply(daysabs, prog, function(x) {
sprintf("M (SD) = %1.2f (%1.2f)", mean(x), sd(x))
}))
##                     General
## "M (SD) = 10.65 (8.20)"
##                    Academic
## "M (SD) = 6.93 (7.45)"
##                   Vocational
## "M (SD) = 2.67 (3.73)"
```

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 = da
##
## Call:
## glm.nb(formula = daysabs ~ math + prog,
        data = dat, init.theta = 1.032713156,
##      link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.155   -1.019   -0.369    0.229    2.527
```

```
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 '.'
```

```
## (Dispersion parameter for Negative Binomi
##
##      Null deviance: 427.54  on 313  degree
## Residual deviance: 358.52  on 310  degree
## AIC: 1741
##
## Number of Fisher Scoring iterations: 1
##
##
##                   Theta:  1.033
##              Std. Err.:  0.106
```

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.

- 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).

- 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.

```
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
```

- 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.
- ▶ Note that R parameterizes this differently from SAS, Stata, and SPSS.
- ▶ The R parameter (theta) is equal to the inverse of the dispersion parameter (alpha) estimated in these other software packages. Thus, the theta value of 1.033 seen here is equivalent to the 0.968 value seen in the Stata Negative Binomial Data Analysis Example because $1/0.968 = 1.033$.

Checking model assumption

- As we mentioned earlier, negative binomial models assume the conditional means are not equal to the conditional variances.
- This inequality is captured by estimating a dispersion parameter (not shown in the output) that is held constant in a Poisson model.
- Thus, the Poisson model is actually nested in the negative binomial model.
- We can then use a likelihood ratio test to compare these two and test this model assumption.
- To do this, we will run our model as a Poisson.

Negative Binomial Regression with R

```
m3 <- glm(daysabs ~ math + prog,
        family = "poisson", data = dat)
pchisq(2 * (logLik(m1) - logLik(m3)),
        df = 1, lower.tail = FALSE)
## 'log Lik.' 2.157e-203 (df=5)
```

Negative Binomial Regression with R

- In this example the associated chi-squared value is 926.03 with one degree of freedom.
- This strongly suggests the negative binomial model, estimating the dispersion parameter, is more appropriate than the Poisson model.

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.

```
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
```

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

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:

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

$$\therefore$$

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

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.

Negative Binomial Regression with R

- ► More details can be found in the Modern Applied Statistics with S by W.N. Venables and B.D. Ripley (the book companion of the MASS package).

- ► For additional information on the various metrics in which the results can be presented, and the interpretation of such, please see Regression Models for Categorical Dependent Variables Using Stata, Second Edition by J. Scott Long and Jeremy Freese (2006).

Predicted values

▶ For assistance in further understanding the model, we can look at predicted counts for various levels of our predictors. Below we create new datasets with values of math and prog and then use the predict command to calculate the predicted number of events.

Negative Binomial Regression with R

- First, we can look at predicted counts for each value of prog while holding math at its mean. To do this, we create a new dataset with the combinations of prog and math for which we would like to find predicted values, then use the predict command.

```
newdata1 <- data.frame(math = mean(dat$math), prog =
labels = levels(dat$prog)))
newdata1$phat <- predict(m1, newdata1, type = "respon
newdata1
    math      prog   phat
 1 48.27    General 10.237
 2 48.27   Academic  6.588
 3 48.27 Vocational  2.850
```

Negative Binomial Regression with R

- In the output above, we see that the predicted number of events (e.g., days absent) for a general program is about 10.24, holding math at its mean.

- The predicted number of events for an academic program is lower at 6.59, and the predicted number of events for a vocational program is about 2.85.

Below we will obtain the mean predicted number of events for values of math across its entire range for each level of prog and graph these.

```
newdata2 <- data.frame(
math = rep(seq(from = min(dat$math),
       to = max(dat$math), length.out = 100), 3),
prog = factor(rep(1:3, each = 100), levels = 1:3, lab
levels(dat$prog)))
```

```
newdata2 <- cbind(newdata2, predict(m1, newdata2, type
newdata2 <- within(newdata2, {
DaysAbsent <- exp(fit)
LL <- exp(fit - 1.96 * se.fit)
UL <- exp(fit + 1.96 * se.fit)
})
```
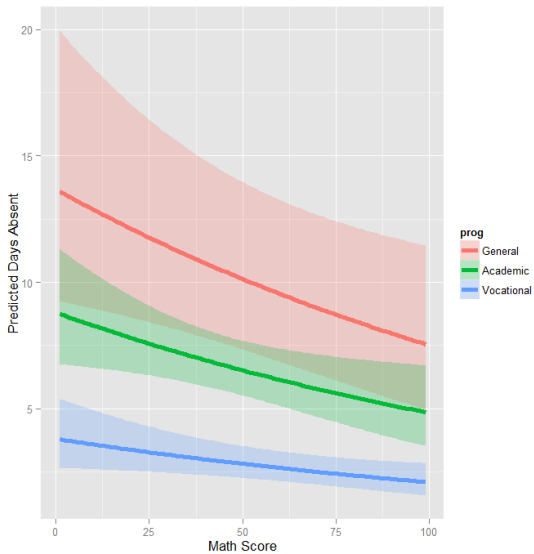
Figure:

Negative Binomial Regression with R

```
ggplot(newdata2, aes(math, DaysAbsent)) +
geom_ribbon(aes(ymin = LL, ymax = UL, fill = prog), a
geom_line(aes(colour = prog), size = 2) +
labs(x = "Math Score", y = "Predicted Days Absent")
```

Negative Binomial Regression with R

The graph shows the expected count across the range of math scores, for each type of program along with 95 percent confidence intervals. Note that the lines are not straight because this is a log linear model, and what is plotted are the expected values, not the log of the expected values.

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.

Negative Binomial Regression with R

- 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

Poisson Regression "Exposure" and offset

- ▶ Poisson regression may also be appropriate for rate data, where the rate is a count of events occurring to a particular unit of observation, divided by some measure of that unit's exposure.

- ▶ For example, biologists may count the number of tree species in a forest, and the rate would be the number of species per square kilometre.

- ▶ Demographers may model death rates in geographic areas as the count of deaths divided by personyears.

- ▶ More generally, event rates can be calculated as events per unit time, which allows the observation window to vary for each unit.

Poisson Regression : Exposure and Offset

In these examples, exposure is respectively unit area, personyears and unit time. In Poisson regression this is handled as an offset, where the exposure variable enters on the right-hand side of the equation, but with a parameter estimate (for log(exposure)) constrained to 1.

$$\log\left(\mathrm{E}(Y \mid x)\right) = \log\left(\textsf{exposure}\right) + \theta' x$$

which implies

$$\log\left(\mathrm{E}(Y \mid x)\right) - \log\left(\textsf{exposure}\right) = \log\left(\frac{\mathrm{E}(Y \mid x)}{\textsf{exposure}}\right) = \theta' x$$

Poisson Regression : Exposure and Offset

Offset in the case of a GLM in R can be achieved
using the offset() function:

```
glm(y ~ offset(log(exposure)) + x, family=po
```

Zero-inflated Regression models

- 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 models estimate two equations simultaneously, one for the count model and one for the excess zeros.

Zero-inflated Regression models

- ▶ Zero-inflated poisson regression is used to model count data that has an excess of zero counts.
- ▶ Further, 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.
- ▶ Thus, the zip model has two parts, a poisson count model and the logit model for predicting excess zeros.
- ▶ You may want to review these Data Analysis Example pages, Poisson Regression and Logit Regression.

Description of the data Let's pursue Example 2 from above.
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. We will use the variables
child, persons, and camper in our model. Let's look at the data.

```
zinb <- read.csv("http://www.ats.ucla.edu/stat/data/fish.csv")
zinb <- within(zinb, nofish <- factor(nofish) livebait <-
factor(livebait) camper <- factor(camper) )
summary(zinb)
```

Zero Truncated Poisson Distribution

R Data Analysis Examples: Zero-Truncated Poisson Regression

Zero-truncated poisson regression is used to model count data for which the value zero cannot occur. This page uses the following packages. Make sure that you can load them before trying to run the examples on this page. If you do not have a package installed, run: install.packages("packagename"), or if you see the version is out of date, run: update.packages(). require(foreign) require(ggplot2) require(VGAM) require(boot) Version info: Code for this page was tested in R Under development (unstable)

Zero Truncated Poisson Distribution

The zero-truncated Poisson (ZTP) distribution is a certain discrete probability distribution whose support is the set of positive integers. This distribution is also known as the conditional Poisson distribution[1] or the positive Poisson distribution.[2] It is the conditional probability distribution of a Poisson-distributed random variable, given that the value of the random variable is not zero. Thus it is impossible for a ZTP random variable to be zero. 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

Zero Truncated Poisson Distribution

Since the ZTP is a truncated distribution with the truncation stipulated as k ¿ 0, one can derive the probability mass function g(k;) from a standard Poisson distribution f(k;) as follows: [4]

$$g(k; \lambda) = P(X = k \mid k > 0) = \frac{f(k; \lambda)}{1 - f(0; \lambda)} = \frac{\lambda^k e^{-\lambda}}{k! \, (1 - e^{-\lambda}}$$

Distribution Parameters

The mean is

$$\mathrm{E}[X] = \frac{\lambda}{1 - e^{-\lambda}} = \frac{\lambda e^{\lambda}}{e^{\lambda} - 1}$$

and the variance is

$$\mathrm{Var}[X] = \frac{\lambda}{1 - e^{-\lambda}} - \frac{\lambda^2 e^{-\lambda}}{(1 - e^{-\lambda})^2}$$

$$= \frac{\lambda e^{\lambda}}{e^{\lambda} - 1} \left[1 - \frac{\lambda}{e^{\lambda} - 1} \right]$$

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

This page uses the following packages. Make sure that you can load them before trying to run the examples on this page. If you do not have a package installed, run: install.packages("packagename"), or if you see the version is out of date, run: update.packages().

```
require(foreign)
require(ggplot2)
require(VGAM)
require(boot)
```

Please Note: The purpose of this page is to show how to use various data analysis commands. It does not cover all aspects of the research process which researchers are expected to do. In particular, it does not cover data cleaning and verification, verification of assumptions, model diagnostics and potential follow-up analyses.

```
## melt data set to long for ggplot2
lpp <- melt(pp.write, id.vars = c("ses", "write"), value.
head(lpp)  # view first few rows

##   ses write variable probability
## 1 low   30 academic    0.09844
## 2 low   31 academic    0.10717
## 3 low   32 academic    0.11650
## 4 low   33 academic    0.12646
## 5 low   34 academic    0.13705
## 6 low   35 academic    0.14828
```

```
## plot predicted probabilities across write values for e
## facetted by program type
ggplot(lpp, aes(x = write, y = probability, colour = ses)
    ., scales = "free")
```

Ordinal Logistic Regression with R

```
ggplot(dat, aes(x = apply, y = gpa)) +
  geom_boxplot(size = .75) +
  geom_jitter(alpha = .5) +
  facet_grid(pared ~ public, margins = TRUE) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1,
```

Poisson Regression with R

```
with(p, tapply(num_awards, prog, function(x) {
  sprintf("M (SD) = %1.2f (%1.2f)", mean(x), sd(x))
}))
```

Poisson Regression with R

```
##                  General              Academic
## "M (SD) = 0.20 (0.40)" "M (SD) = 1.00 (1.28)" "M (SD)

ggplot(p, aes(num_awards, fill = prog)) +
  geom_histogram(binwidth=.5, position="dodge")
```

```
ggplot(dat, aes(daysabs, fill = prog)) + geom_histogram(b
    ., margins = TRUE, scales = "free")
```

```
with(dat, tapply(daysabs, prog, function(x) {
    sprintf("M (SD) = %1.2f (%1.2f)", mean(x), sd(x))
}))

##                    General              Academic
## "M (SD) = 10.65 (8.20)"  "M (SD) = 6.93 (7.45)"  "M (S
```

Poisson Regression

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

Poisson Regression with R

Examples of Poisson regression

- ▸ 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.

Examples of Poisson regression

- The number of people in line in front of you at the grocery store.
- Predictors may include the number of items currently offered at a special discounted price and whether a special event (e.g., a holiday, a big sporting event) is three or fewer days away.

Examples of Poisson regression

- The number of awards earned by students at one 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.

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

Description of the data

- ▶ For the purpose of illustration, we have simulated a data set for the last example.
- ▶ The data set is called *poissonreg.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
```

Poisson Regression with R

- Each variable has 200 valid observations and their distributions seem quite reasonable.
- The unconditional mean and variance of our outcome variable are not extremely different.
- Our model assumes that these values, conditioned on the predictor variables, will be equal (or at least roughly so).

- Additionally, the means and variances within each level of prog–*the conditional means and variances*–are similar.
- A conditional histogram separated out by program type is plotted to show the distribution.