Negative Binomial Regression with R

Let's continue with our description of the variables in this dataset. 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. The variances within each level of prog are higher than the means within each level. 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                Academic
## "M (SD) = 10.65 (8.20)"  "M (SD) = 6.93 (7.45)"  "M
```

Analysis methods you might consider Below is a list of some analysis methods you may have encountered. Some of the methods listed are quite reasonable, while others have either fallen out of favor or have limitations.

Negative Binomial Regression with R

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

Negative Binomial Regression with R

Poisson regression - Poisson regression is often used for modeling count data. Poisson regression has a number of extensions useful for count models. Zero-inflated regression model - 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.

Negative Binomial Regression with R

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), as well as the lack of capacity to model the dispersion.

Negative Binomial Regression with R

Negative binomial regression analysis Below we use the glm.nb function from the MASS package to estimate a negative binomial regression.

```
summary(m1 <- glm.nb(daysabs ~ math + prog,
##
## Call:
## glm.nb(formula = daysabs ~ math + prog, d
##     link = log)
##
## Deviance Residuals:
##    Min       1Q   Median       3Q      Max
```

Negative Binomial Regression with R

```
## Coefficients:
##                  Estimate Std. Error z valu
## (Intercept)       2.61527    0.19746   13.2
## math             -0.00599    0.00251   -2.3
## progAcademic     -0.44076    0.18261   -2.4
## progVocational   -1.27865    0.20072   -6.3
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’
```

Negative Binomial Regression with R

```
## (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).

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.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 Binomi
##
## Response: daysabs
##         Model  theta Resid. df    2 x log
## 1        math 0.8559       312
## 2 math + prog 1.0327       310
##      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. 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 $1/0.968 = 1.033$.

Negative Binomial Regression with R

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 = "po
pchisq(2 * (logLik(m1) - logLik(m3)), df = 1,
## 'log Lik.' 2.157e-203 (df=5)
```

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.

Negative Binomial Regression with R

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
```

Negative Binomial Regression with R

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

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

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

$$\therefore$$

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

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. 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).

Negative Binomial Regression with R

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).

Negative Binomial Regression with R

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.