Negative Binomial Regression with R

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

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 If you do not have a package installed, run: install.packages("packagename"), or if you see the version is out of date, run: update.packages().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(MASS)
```

Negative Binomial Regression with R

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 checking, verification of assumptions, model diagnostics or potential follow-up analyses.

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.

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 **nb_data**. 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

Let's look at the data. It is always a good idea to start with descriptive statistics and plots.

```
dat <- read.dta("http://www.ats.ucla.edu/sta
dat <- within(dat, {
prog <- factor(prog, levels = 1:3, labels =
id <- factor(id)
})
```

Negative Binomial Regression with R

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

Negative Binomial Regression with R

```
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.

Negative Binomial Regression with R

```
newdata1 <- data.frame(math = mean(dat$math)
labels = levels(dat$prog)))
newdata1$phat <- predict(m1, newdata1, type
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.

Negative Binomial Regression with R

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.

Negative Binomial Regression with R

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

Negative Binomial Regression with R

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

Negative Binomial Regression with R

Negative Binomial Regression with R

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

Negative Binomial Regression with `R`

Negative Binomial Regression with R

Plot of the model predicted days absent with confidence intervals 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.

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.

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. See the glm documentation for details.

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 (see Cameron and Trivedi (1998) and Dupont (2002) for more information).