PPOL 503-03, PPOL 503-04, Fall 2016
Course Notes #11: Poisson Regression

Poisson regression is for modeling count variables with the assumption that the conditional mean equal the conditional variance.

Examples of Poisson regression

Example 1. 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.

Example 2. A researcher in education is interested in the association between the number of awards earned by students at one high school and the students' performance in math and the type of program (e.g., vocational, general or academic) in which students were enrolled.

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, **math** is a continuous predictor variable and represents students' scores on their math final exam, and **prog** is a categorical predictor variable with three levels indicating the type of program in which the students were enrolled.

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

```
sum num_awards
```

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
  num_awards |       200         .63    1.052921          0          6
```

```
dis r(Var)
1.1086432
```

The mean of **num_awards** is 0.63 and that the variance is approximately 1.12. The distribution assumption of a Poisson regression is that the conditional mean is equal to the conditional variance. Even though the above output is for the unconditional mean and variance it can be informative because it gives us some indication of whether a Poisson regression should be used. In this example, **num_awards** appears to be slightly over dispersed, as the variance is larger than the mean. However, the variance is not substantially larger than the mean and the

predictor variables should help, so it may be reasonable to fit a Poisson regression model.

tabstat num_awards, by(prog) stats(mean sd n)

```
Summary for variables: num_awards
     by categories of: prog (type of program)

    prog |      mean        sd         N
---------+------------------------------
 general |        .2  .4045199        45
academic |         1  1.278521       105
vocation |       .24  .5174506        50
---------+------------------------------
   Total |       .63  1.052921       200
```

.sum math

```
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        math |       200      52.645    9.368448         33         75
```

Poisson regression analysis

Below we use the **poisson** command to estimate a Poisson regression model. The **i.** before **prog** indicates that it is a factor variable (i.e., categorical variable), and that it should be included in the model as a series of indicator variables.

We use the **vce(robust)** option to obtain robust standard errors for the parameter estimates. Cameron and Trivedi (2009) recommend the use of robust standard errors when estimating a Poisson model.

poisson num_awards i.prog math, vce(robust)

```
Iteration 0:   log pseudolikelihood = -182.75759
Iteration 1:   log pseudolikelihood = -182.75225
Iteration 2:   log pseudolikelihood = -182.75225

Poisson regression                                Number of obs   =  200
                                                  Wald chi2(3)    =  80.15
                                                  Prob > chi2     = 0.0000
Log pseudolikelihood = -182.75225                 Pseudo R2       = 0.2118
------------------------------------------------------------------------------
-
             |               Robust
num_awards |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
prog |
          2  |   1.083859   .3218538     3.37   0.001     .4530373    1.714681
          3  |   .3698092   .4014221     0.92   0.357    -.4169637    1.156582
                |
```

```
        math |   .0701524   .0104614     6.71   0.000     .0496485    .0906563
       _cons |  -5.247124   .6476195    -8.10   0.000    -6.516435   -3.977814
------------------------------------------------------------------------------
```

- The last value in the iteration log is the final value of the log pseudolikelihood for the full model and is displayed again. Because we asked for robust standard errors, the likelihood is a pseudolikelihood. The estimates are still the maximum likelihood estimates. But the robust standard errors are estimated using the sandwich estimator.
- The Wald chi-square statistic with three degrees of freedom for the full model, followed by the p-value for the chi-square. The model, as a whole, is statistically significant. The pseudo-R^2 is 0.21.
- Below the header you will find the Poisson regression coefficients for each of the variables along with robust standard errors, z-scores, p-values and 95% confidence intervals for the coefficients. The coefficient for **math** is .07 and is statistically significant. This means that the expected change in the log count for a one-unit increase in **math** is .07. The indicator variable **2.prog** is also statistically significant. Compared to level 1 of **prog**, the expected log count for level 2 of **prog** increases by about 1.1. The indicator variable **3.prog** is not statistically significant. To determine if **prog** itself, overall, is statistically significant, we can use the **test** command to obtain the two degree-of-freedom test of this variable.

```
test 2.prog 3.prog
```

```
 ( 1)  [num_awards]2.prog = 0
 ( 2)  [num_awards]3.prog = 0

           chi2(  2) =   14.76
         Prob > chi2 =    0.0006
```

The two degree-of-freedom chi-square test indicates that **prog**, taken together, is a statistically significant predictor of **num_awards**.

To help assess the fit of the model, the **estat gof** command can be used to obtain the goodness-of-fit chi-squared test.

```
estat gof
```

```
         Goodness-of-fit chi2  =  189.4496
         Prob > chi2(196)      =    0.6182
```

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. In that

situation, we may try to determine if there are omitted predictor variables, or if a negative binomial model may be more appropriate for over-dispersion.

If you would like the results displayed as incident rate ratios, you can use the **irr** option.

```
poisson, irr

Poisson regression                                      Number of obs   =  200
                                                        Wald chi2(3)    =  80.15
                                                        Prob > chi2     =  0.0000
Log pseudolikelihood = -182.75225                       Pseudo R2       =  0.2118

------------------------------------------------------------------------------
             |               Robust
  num_awards |        IRR   Std. Err.      z    P>|z|     [95% Conf.Interval]
-------------+----------------------------------------------------------------
-
        prog |
          2  |   2.956065   .9514208     3.37   0.001     1.573083   5.554903
          3  |   1.447458   .5810418     0.92   0.357     .6590449   3.179049
             |
        math |   1.072672   .0112216     6.71   0.000     1.050902   1.094893
------------------------------------------------------------------------------
```

The output above indicates that the expected incident rate for **2.prog** is 2.96 times the expected incident rate for the reference group (**1.prog**). Likewise, the expected incident rate for **3.prog** is 1.45 times the expected incident rate for the reference group. The percent change in the incident rate of **num_awards** increases by 7% for every unit increase in **math**.

Below we use the **margins** command to calculate the predicted counts at each level of **prog**, holding all other variables (in this example, **math**) in the model at their means.

```
margins prog, atmeans
Adjusted predictions                                    Number of obs   = 200
      Model VCE    : Robust

        Expression   : Predicted number of events, predict()
      at           : 1.prog          =        .225 (mean)
                     2.prog          =        .525 (mean)
                     3.prog          =         .25 (mean)
                     math            =      52.645 (mean)
------------------------------------------------------------------------------
-
      |            Delta-method
      |     Margin   Std. Err.      z    P>|z|     [95% Conf.Interval]
------------+-----------------------------------------------------------------
 prog |
```

```
   1  |     .211411   .0627844     3.37   0.001     .0883558    .3344661
    2  |   .6249446   .0887008     7.05   0.000     .4510943    .7987949
    3  |   .3060086   .0828648     3.69   0.000     .1435966    .4684205
------------------------------------------------------------------------------
```

The predicted number of events for level 1 of **prog** is about .21, holding **math** at its mean. The predicted number of events for level 2 of **prog** is higher at .62, and the predicted number of events for level 3 of **prog** is about .31.

Below we will obtain the predicted counts for values of **math** that range from 35 to 75 in increments of 10.

```
margins, at(math=(35(10)75)) vsquish

Predictive margins                                 Number of obs   = 200
      Model VCE    : Robust

      Expression   : Predicted number of events, predict()
      1._at        : math            =          35
      2._at        : math            =          45
      3._at        : math            =          55
      4._at        : math            =          65
      5._at        : math            =          75
------------------------------------------------------------------------------
|            Delta-method
|        Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
_at |
1  |   .1311326   .0358696     3.66   0.000     .0608295    .2014358
2  |   .2644714    .047518     5.57   0.000     .1713379    .3576049
3  |   .5333923   .0575203     9.27   0.000     .4206546      .64613
4  |   1.075758   .1220143     8.82   0.000     .8366147    1.314902
5  |   2.169615   .4115856     5.27   0.000     1.362922    2.976308
------------------------------------------------------------------------------
```

The table above shows that with **prog** at its observed values and **math** held at 35 for all observations, the average predicted count (or average number of awards) is about .13; when **math** = 75, the average predicted count is about 2.17.

To obtain a test of the over-dispersion parameter alpha, you can run the same model as a negative binomial regression. We have included the **nolog** option to suppress the display of the iteration log.

nbreg num_awards i.prog math, nolog

```
Negative binomial regression                      Number of obs   =  200
                                                  LR chi2(3)      = 70.93
Dispersion     = mean                             Prob > chi2     = 0.0000
Log likelihood = -181.90545                       Pseudo R2       = 0.1632

------------------------------------------------------------------------------
 num_awards |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
------------+----------------------------------------------------------------
       prog |
         2  |   1.075071   .3676128     2.92   0.003     .3545633    1.795579
         3  |   .3669593   .4526298     0.81   0.418    -.5201788    1.254097
            |
       math |   .0710688   .0116342     6.11   0.000     .0482661    .0938714
       cons |  -5.293312   .7088893    -7.47   0.000     -6.68271   -3.903915
------------+----------------------------------------------------------------
   /lnalpha |  -1.810689   .9133063                     -3.600737    -.020642
------------+----------------------------------------------------------------
      alpha |   .1635413   .1493633                      .0273036    .9795696
------------------------------------------------------------------------------

Likelihood-ratio test of alpha=0:  chibar2(01) =  1.69 Prob>=chibar2 = 0.097
```

The likelihood-ratio test of the over-dispersion parameter alpha is not statistically significant (prob >= chibar2 = 0.097), which suggests that Poisson regression is an appropriate model.

Points to Remember

- Poisson models should not be applied to small samples.
- The assumption of the Poisson model that the conditional mean is equal to the conditional variance needs to be checked. There are several tests of the over-dispersion parameter alpha, including the likelihood ratio test provided with the output of **nbreg**. Failing to reject the null hypothesis that there is no over dispersion usually is an indication of a problem of model specification, related to omitted predictor variables and the functional form of the predictor variables.
- One common cause of over-dispersion is excess zeros generated by an additional data generating process. In this situation, zero-inflated models 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.