Overdispersion and Excess Zeroes

Count Models: From Poisson to Hurdle Models

Josemari Feliciano

July 17, 2020 @ Medidata

Poisson Regression

Key Assumptions

- Outcome variable consists of count data {0,1,2,…}
- Independence of observations, this means each observation is independent of the other observations; that is, one observation cannot provide any information on another observation.
- The mean and variance of the model are identical.

Formal Notation:

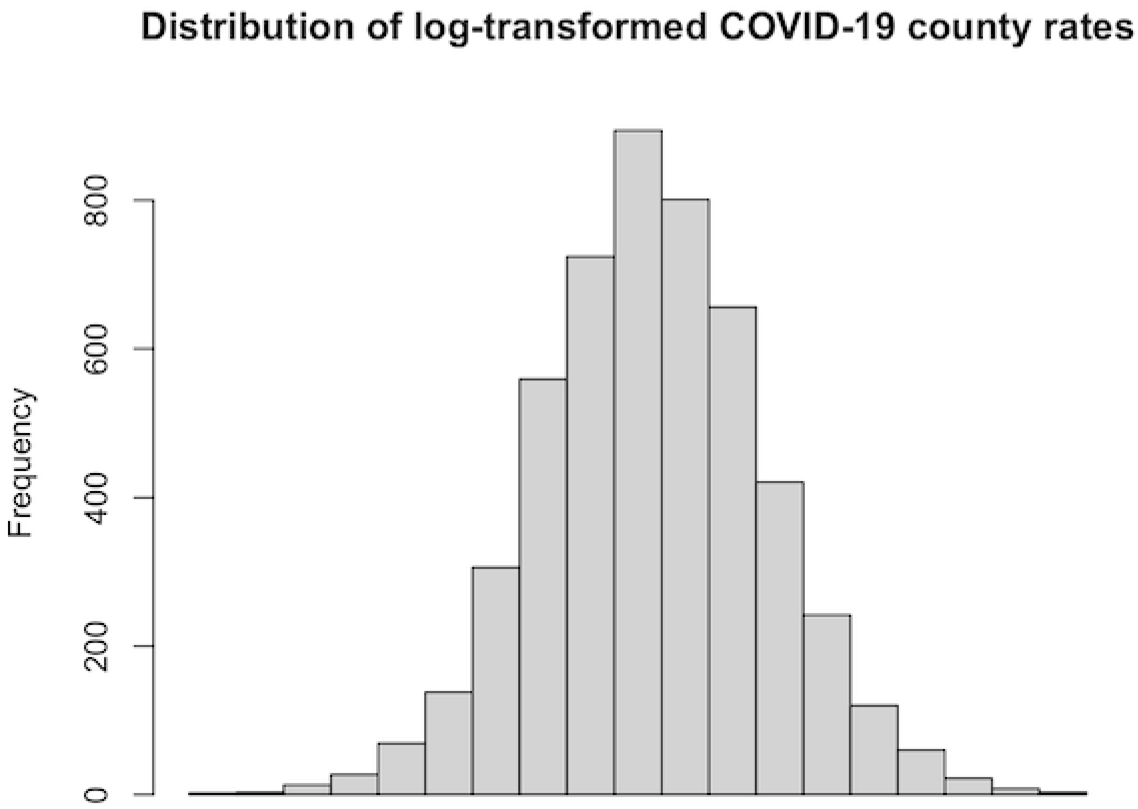
$$log(\mu_i) = log(E(Y_i)) = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi}$$

Let us keep it simple:

$$log(\mu) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

Working Around Poisson

It is a common practice to log transform count/rate data then see if it is approximately normal to use common linear models (OLS, WLS, LME, ANOVA).



Applied Example: Pharmacy-Level Data

Here's a test dataset I created for 915 individuals for this presentation.

```
head(drug_df)
```

```
##   abandoned discount female num_other_meds final_copay pharmacy_distance
## 1         0        0      1              0          68                 7
## 2         0        1      0              0          60                 6
## 3         0        1      0              0          98                 6
## 4         0        0      1              1          22                 3
## 5         0        1      0              0          90                26
## 6         0        1      1              2          86                 2
```

Data Descriptors:

- abandoned: the number of times the patient has abandoned their medication.
- discount: 1 if patient offered e-voucher discount by pharmacy tech, 0 if not.
- female: 1 if female, 0 otherwise.
- num_other_meds: number of other medications the patients have to pick up.
- final_copay: final price for the medication of interest.
- pharmacy_distance: distance of pharmacy from patients.

Poisson Regression in R

For simplicity and comparability to later models, let us argue that it is important to control for all variables described earlier.

Here's a script to create the initial Poisson model in R.

```
poi_model <- glm(abandoned ~ discount + female + num_other_meds +
                         final_copay + pharmacy_distance,
                       family = poisson, data = drug_df)
```

Poisson Mean and Variance Assumption

Let us compare the mean and variance of the number of abandoned prescriptions.

```
data.frame(mean= mean(drug_df$abandoned),
           variance = var(drug_df$abandoned),
           ratio = var(drug_df$abandoned)/mean(drug_df$abandoned))
```

```
##       mean variance    ratio
## 1 1.692896 3.709742 2.191358
```

Earlier: For Poisson models, the mean and variance of the outcome are identical.

A common soft-rule in medicine: Not a problem if ratio of mean and variance is less than 2. Flexibility on this.

Overdispersion: $Var[Y_i] > E[Y_i]$ (common)

Underdispersion: $Var[Y_i] < E[Y_i]$ (rare)

Model Output

```
round(summary(poi_model)$coefficients, 4)
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.2767     0.1088  2.5426   0.0110
## discount           -0.2245     0.0546 -4.1119   0.0000
## female              0.1559     0.0612  2.5452   0.0109
## num_other_meds     -0.1850     0.0401 -4.6116   0.0000
## final_copay         0.0009     0.0012  0.7560   0.4496
## pharmacy_distance   0.0254     0.0020 12.5835   0.0000
```

When overdispersion is present, the model underestimates the standard errors.

Two solutions: scale your standard errors or use a different model such as negative binomial.

Scale Parameter Method

We can scale our standard errors with a parameter (σ) where:

$$\sigma = \frac{model_{deviance}}{model_{df}}$$

To avoid manual work, set the `family` parameter from poisson to quasipoisson:

```
quasi_model <- glm(abandoned ~ discount + female + num_other_meds +
                     final_copay + pharmacy_distance,
                   family = quasipoisson, data = drug_df)
```

Now let us compare the outputs from poisson and quasipoisson models.

Overdispersion: Standard Errors and Significance

For Poisson:

```
round(summary(poi_model)$coefficients, 4)
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.2767     0.1088  2.5426   0.0110
## discount           -0.2245     0.0546 -4.1119   0.0000
## female              0.1559     0.0612  2.5452   0.0109
## num_other_meds     -0.1850     0.0401 -4.6116   0.0000
## final_copay         0.0009     0.0012  0.7560   0.4496
## pharmacy_distance   0.0254     0.0020 12.5835   0.0000
```

For Quasipoisson:

```
round(summary(quasi_model)$coefficients, 4)
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.2767     0.1472  1.8800   0.0604
## discount           -0.2245     0.0738 -3.0404   0.0024
## female              0.1559     0.0828  1.8820   0.0602
## num_other_meds     -0.1850     0.0543 -3.4098   0.0007
## final_copay         0.0009     0.0017  0.5590   0.5763
## pharmacy_distance   0.0254     0.0027  9.3043   0.0000
```

Negative Binomial in R

Overdispersion could indicate that the sample comes from a different but Poisson-like distribution.

Negative Binomial is often referred to as Poisson-Gamma Mixture where the variance is greater than the mean.

In R, we can use the MASS package to run negative binomial models.

```r
library(MASS)
nb_model <- glm.nb(abandoned ~ discount + female + num_other_meds +
                         final_copay + pharmacy_distance,  data = drug_df)
```

Negative Binomial in R

Overdispersion could indicate that the sample comes from a different but Poisson-like distribution.

Negative Binomial is often referred to as Poisson-Gamma Mixture where the variance is greater than the mean.

In R, we can use the MASS package to run negative binomial models.

```
round(summary(nb_model)$coefficients, 4)
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.2282     0.1451  1.5724   0.1159
## discount           -0.2158     0.0726 -2.9717   0.0030
## female              0.1510     0.0819  1.8428   0.0654
## num_other_meds     -0.1766     0.0528 -3.3438   0.0008
## final_copay         0.0010     0.0017  0.6216   0.5342
## pharmacy_distance   0.0289     0.0032  8.9634   0.0000
```

Poisson vs Negative Binomial

Since we used the same set of variables, the model with the lowest AIC is the better model.

```
summary(poi_model)$aic
```

```
## [1] 3313.777
```

```
summary(nb_model)$aic
```

```
## [1] 3135.716
```

Since negative binomial provides the lowest AIC, this is the better model.

The other common issue: zero inflation.

- Suppose we are starting with a Poisson model from scratch.
- The number of zeroes in the data are unusually high.

Multiple Data Generating Processes

- Instead of having one data generating process, there could be other factors that could be impacting the outcome distribution.
- Zero inflated poisson (ZIP) models remedy the excess zeroes by attributing a portion of the zero counts to another data generating process.
- In effect, ZIP models attribute a portion of the zero counts by fitting a binary classifier such as logistic regression model (default).
- Classic example on modeling counts of insurance claims: areas with lower uninsured rates are likely to have excess zeroes in claim counts because area residents are not able to file for claims due to lack of insurance.

Running ZIP Models in R

We can use `zeroinfl` from the pscl package.

Here, I am specifying to attribute the excess zeroes to pharmacy distance.

```r
library(pscl)
zip_model <-  zeroinfl(abandoned ~ discount + female + num_other_meds
                       final_copay + pharmacy_distance | pharmacy_distanc
                       data = drug_df)
```

ZIP Model - Binomial Portion

```
summary(zip_model)$coefficient$zero
```

```
##                      Estimate Std. Error   z value     Pr(>|z|)
## (Intercept)       -0.6860090  0.2055132 -3.338029 0.0008437506
## pharmacy_distance -0.1300346  0.0402349 -3.231886 0.0012297610
```

Inspect the significance of pharmacy distance in the binomial portion of the ZIP model.

ZIP Model - Poisson Portion

```r
round(summary(zip_model)$coefficient$count, 4)
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.6084     0.1202  5.0600   0.0000
## discount           -0.2183     0.0588 -3.7116   0.0002
## female              0.1349     0.0660  2.0426   0.0411
## num_other_meds     -0.1628     0.0434 -3.7532   0.0002
## final_copay         0.0000     0.0013  0.0111   0.9911
## pharmacy_distance   0.0182     0.0023  7.9770   0.0000
```

```r
round(summary(poi_model)$coefficients, 4)
```

```
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.2767     0.1088  2.5426   0.0110
## discount           -0.2245     0.0546 -4.1119   0.0000
## female              0.1559     0.0612  2.5452   0.0109
## num_other_meds     -0.1850     0.0401 -4.6116   0.0000
## final_copay         0.0009     0.0012  0.7560   0.4496
## pharmacy_distance   0.0254     0.0020 12.5835   0.0000
```

Improvement vs significant improvement

Is there a significant improvement in using ZIP model over a regular Poisson model?

```r
library(lmtest)
lmtest::lrtest(poi_model, zip_model)
```

```
## Likelihood ratio test
## 
## Model 1: abandoned ~ discount + female + num_other_meds + final_copay + 
##     pharmacy_distance
## Model 2: abandoned ~ discount + female + num_other_meds + final_copay + 
##     pharmacy_distance | pharmacy_distance
##   #Df  LogLik Df  Chisq Pr(>Chisq)    
## 1   6 -1650.9                         
## 2   8 -1605.8  2 90.259  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Perform a likelihood ratio test and inspect significance of improvement. The chi-square statistic indicates an improved fit.

Comparison of Three Models

Compare BIC (instead of AIC) to penalize higher degrees of freedom more:

```
BIC(poi_model, nb_model, zip_model)
```

```
##            df      BIC
## poi_model  6 3342.690
## nb_model   7 3169.448
## zip_model  8 3266.069
```

Other Noteable Count Models

- Zero-Inflated Negative Binomial
- Zero-Truncated
- Hurdle Models

Concluding Thoughts

How I see these models as a Biostatistician.

Thank you for attending my talk.

A copy of my data and slides will be available on: https://github.com/neonseri