Count Models in R

Clay Ford

Spring 2020

Motivation

- We sometimes have count data, i.e., non-negative values: 0, 1, 2, 3, ...
- We want to understand the variability in the count data as well as estimate an expected count given some other information
- Example: count of physician office visits for people over age 60 given predictors such as education, gender, number of chronic conditions, and insurance status
- Does knowing insurance status, level of education, etc, help us estimate an expected count of office visits?
- Count models can help us answer that question

Probability distributions for counts

Two common probability distributions for counts are the Poisson and negative binomial.

- The Poisson distribution is parameterized by its mean. The mean and variance are equal.
  - $E(X) = \mu$ and $Var(X) = \mu$
- The negative binomial is parameterized by a mean and a dispersion parameter ($\theta > 0$). The variance is greater than the mean.
  - $E(X) = \mu$ and $Var(X) = \mu + \mu^2/\theta$

Simulated poisson distribution

Use the `rpois` function to generate random data from a Poisson distribution with a specified mean.

```
# lambda is the mean
d <- rpois(n = 1000, lambda = 3)
table(d)
```

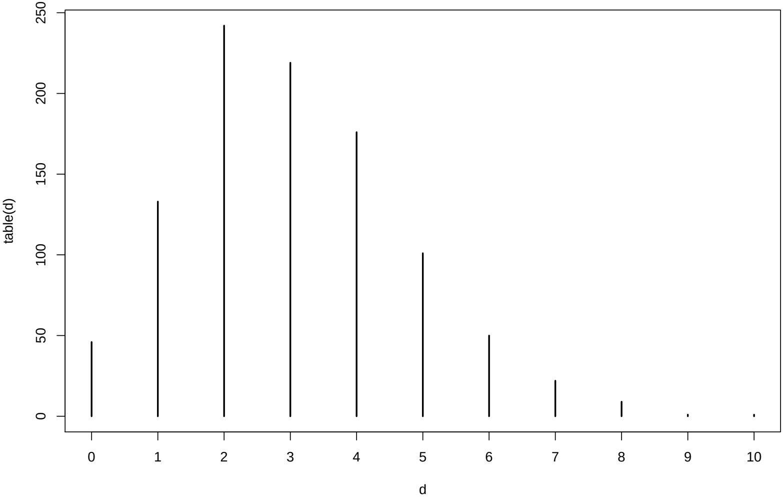
```
## d
##   0   1   2   3   4   5   6   7   8   9  10
##  46 133 242 219 176 101  50  22   9   1   1
```

```
c(mean(d), var(d))
```

```
## [1] 3.028000 2.882098
```

Notice the mean and variance about equal.

Simulated poisson distribution



Simulated negative binomial distribution

Use the `rnbinom` function to generate random data from a negative binomial distribution with a specified mean and dispersion parameter (θ).

```
# size is the dispersion parameter
d <- rnbinom(n = 1000, mu = 3, size = 2)
table(d)
```

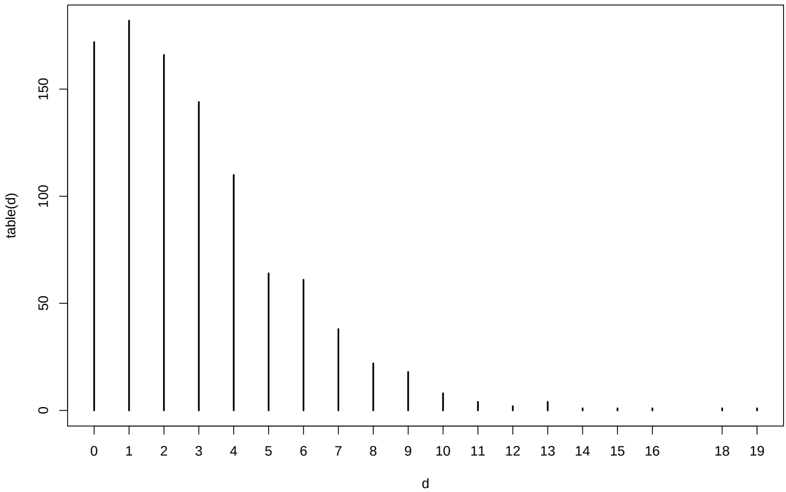
```
## d
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13
## 172 182 166 144 110  64  61  38  22  18   8   4   2   4
```

```
c(mean(d), var(d))
```

```
## [1] 2.95800 7.29353
```

Notice the variance is larger than the mean. As θ gets bigger, the negative binomial approaches a Poisson.

Simulated negative binomial distribution



Basic count models - Poisson

A Poisson count model assumes our count data came from a Poisson distribution with a mean *conditional on predictors*.

```
trt <- sample(0:1, size = 10000, replace = T)
d <- rpois(n = 10000,
           lambda = exp(3 + 2*trt))
log(c(mean(d[trt==0]), var(d[trt==0])))
```

```
## [1] 2.997627 3.008872
```

```
log(c(mean(d[trt==1]), var(d[trt==1])))
```

```
## [1] 4.998798 4.996098
```

The mean and variance of `d` is *conditional* on `trt`. We use `exp` (exponentiate) to ensure lambda is positive, which is what a count model assumes.

Basic count models - Poisson

A Poisson count model assumes our count data came from a Poisson distribution and attempts to recover the mean *conditional on predictors*.

The `glm` function with `family = poisson` fits a count model assuming a Poisson distribution.

```
m <- glm(d ~ trt, family = poisson)
coef(m)
```

```
## (Intercept)         trt
##    2.997627    2.001171
```

The `(Intercept)` is the estimated mean when `trt==0`. Adding `(Intercept)` and `trt` gives the estimated mean when `trt==1`.

Basic count models - negative binomial

A negative binomial count model assumes our count data came from a negative binomial distribution with a mean *conditional on predictors* and a fixed theta.

```
trt <- sample(0:1, size = 10000, replace = T)
d <- rnbinom(n = 10000, mu = exp(3 + 2*trt),
             size = 1.2)
log(c(mean(d[trt==0]), var(d[trt==0])))
```

```
## [1] 2.995722 5.865118
```

```
log(c(mean(d[trt==1]), var(d[trt==1])))
```

```
## [1] 5.005832 9.854248
```

The mean and variance of `d` is *conditional* on `trt`.

Basic count models - negative binomial

A negative binomial count model assumes our count data came from a negative binomial distribution and attempts to recover the mean *conditional on predictors* as well as theta.

The `glm.nb` function from the `MASS` package fits a count model assuming a negative binomial distribution.

```r
library(MASS)
m <- glm.nb(d ~ trt)
coef(m)
```

```
## (Intercept)         trt
##    2.995722    2.010110
```

```r
m$theta
```

```
## [1] 1.196483
```