Checking the “Poisson” assumption in the Poisson generalized linear model

The Poisson regression model is a generalized linear model (glm) satisfying the following assumptions:

- The responses y_i are independent of one another, and each y_i is a non-negative integer, $y_i \in \{0, 1, 2, \ldots\}$.
- Each y_i follows the Poisson distribution with mean λ_i, $P(y_i = k|\lambda_i) = \lambda_i^k e^{-\lambda_i}/k!$.
- $\theta_i = \log \lambda_i = X_i\beta$, where $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$ and $X_i = (1, X_{i1}, X_{i2}, \ldots, X_{ip}$ are the predictors.

A reasonable question to ask is, if confronted with data, how do we know to use the Poisson regression model?

There are two basic ideas that I want to discuss here. The first idea is easy to implement; the second idea is much less easy:

- Are the responses all non-negative integers that have no natural upper bound?
- Is the distribution of y consistent with the Poisson distribution?

The first idea is easy: If the data are anything but non-negative integers that are (in principle, at least) unbounded, Poisson regression is the wrong model to use.

The second idea sounds easy but is a little tricky. I will illustrate with an example from Gelman & Hill, which we also looked at in class.

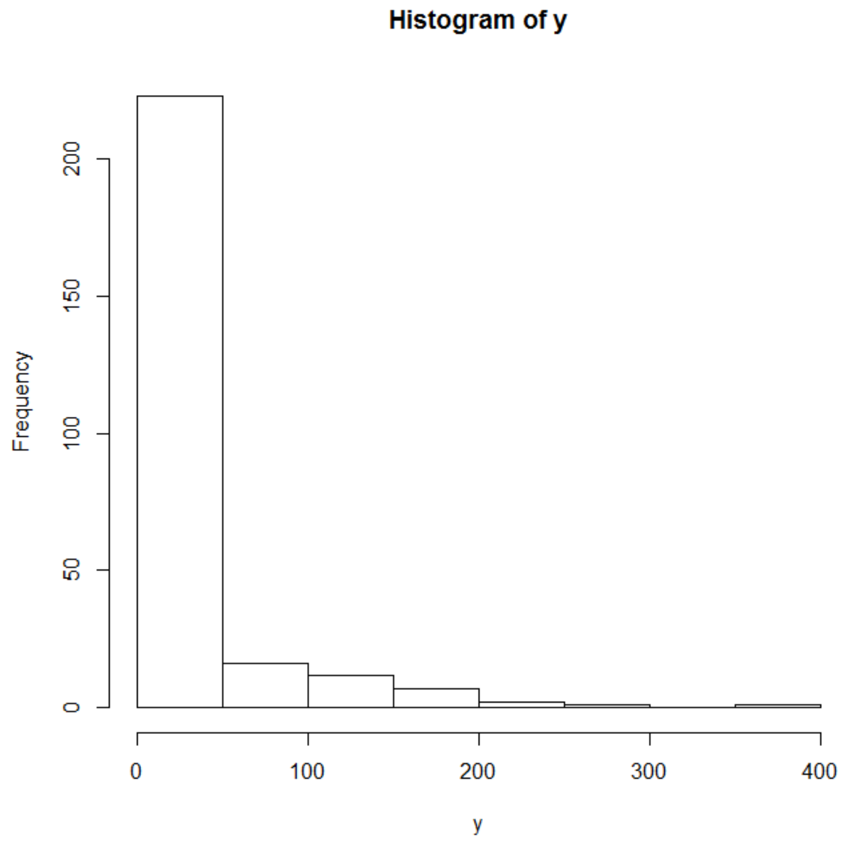
```
> roachdata <- read.csv ("roachdata.csv")
#
# this is data from an experiment on the effectiveness of
# an "integrated pest management system" in apartment buildings
# in a particular city.

> str(roachdata)

'data.frame':   262 obs. of  6 variables:
 $ X        : int  1 2 3 4 5 6 7 8        [observation number]
 $ y        : int  153 127 7 7 0 0        [of roaches trapped after expmt]
 $ roach1   : num  308 331.25 1.67        [of roaches before experiment]
 $ treatment: int  1 1 1 1 1 1 1 1        [pest mgmt tx in this apt bldg?]
 $ senior   : int  0 0 0 0 0 0 0 0        [apts restricted to sr citzns?]
 $ exposure2: num  0.8 0.6 1 1 1.14       [avg of trap-days per apt for y]
```

```
> attach(roachdata)

> glm.1 <- glm (y ˜ roach1 + treatment + senior, family=poisson,
+   offset=log(exposure2))
> summary(glm.1)

Call:
glm(formula = y ˜ roach1 + treatment + senior, family = poisson,
    offset = log(exposure2))

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-17.9430   -5.1529   -3.8059    0.1452   26.7771

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.089e+00  2.123e-02  145.49   <2e-16 ***
roach1       6.983e-03  8.874e-05   78.69   <2e-16 ***
treatment   -5.167e-01  2.474e-02  -20.89   <2e-16 ***
senior      -3.799e-01  3.342e-02  -11.37   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 16954  on 261  degrees of freedom
Residual deviance: 11429  on 258  degrees of freedom
AIC: 12192

Number of Fisher Scoring iterations: 6
```

The model obviously fits much better than the intercept-only model, but is the Poisson assumption really met?

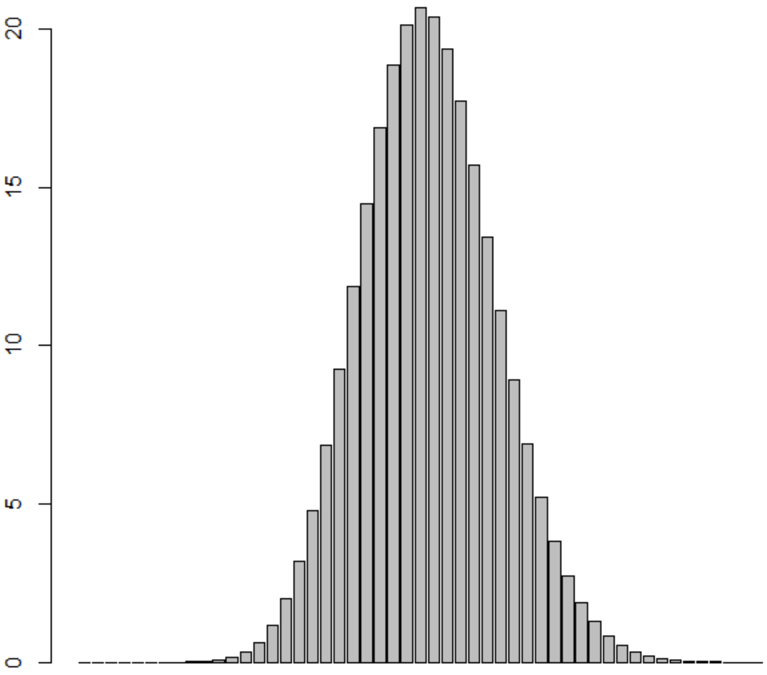
The first possibility that comes to mind for assessing this is to make a histogram of the y's and see if it looks like the right Poisson distribution.

```
hist(y)
```



However, is it the right Poisson distribution? We could compare it with the Poisson distribution whose mean is estimated from the y's alone:

```
> length(y)
[1] 262
> (lambda <- mean(y))
[1] 25.65
> tbl <- NULL
> for (k in 0:50) {
+  tbl <- rbind(tbl,c(obs=sum(y==k),
+     exp=262*exp(-25.65) * 25.65^k / factorial(k)))
+ }
> barplot(tbl[,2])
```



The two distributions don't look very similar, a fact confirmed when we compare the actual counts for $y = 0, 1, 2$, etc., with the counts that come from the Poisson(25.65) distribution, for the first few integers:

```
> dimnames(tbl)[[1]] <- paste(0:50)
> tbl[1:5,]
  obs          exp
0  94 1.899537e-09
1  20 4.872313e-08
2  11 6.248742e-07
3  10 5.342674e-06
4   7 3.425990e-05
```

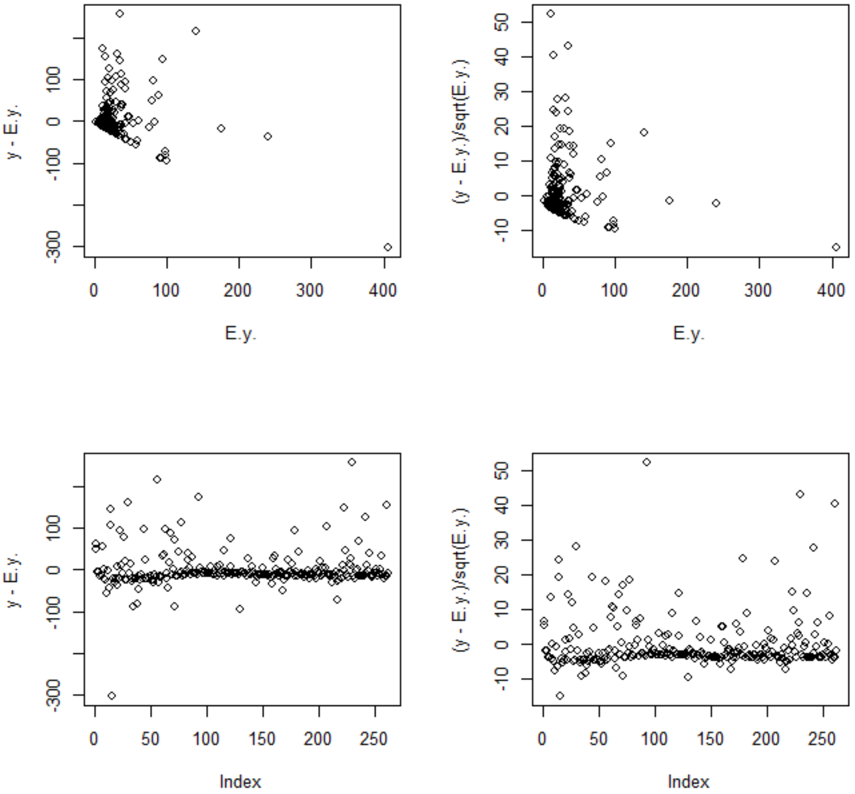
The problem is that, while each y_i is Poisson with mean $\lambda_i = X_i\beta$, these means may be very different from one anther, and different from the overall mean 25.65, so the raw distribution of the y's may not look very much like a Poisson(25.65) distribution (and it does not, in this case!).

One thing we can do is to compare the values predicted from the model with the actual y's. The predicted values are easy to compute:

```
> E.y. <- predict(glm.1,type="response")
```

and then we have to think of a way to compare them with the actual y's. One possibility is just to plot raw residuals $y_i - E[y_i]$ or standardized residuals $(y_i - E[y_i])/E[y_i]$. Here are four possible plots:

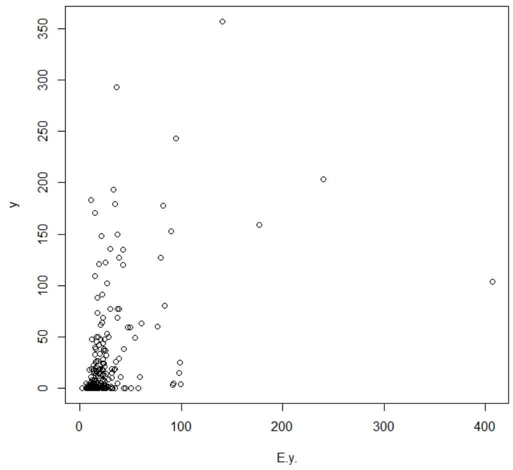
```
> par(mfrow=c(2,2))
> plot(E.y., y - E.y.)
> plot(E.y., (y - E.y.)/sqrt(E.y.))
> plot(y - E.y.)
> plot((y - E.y.)/sqrt(E.y.))
```

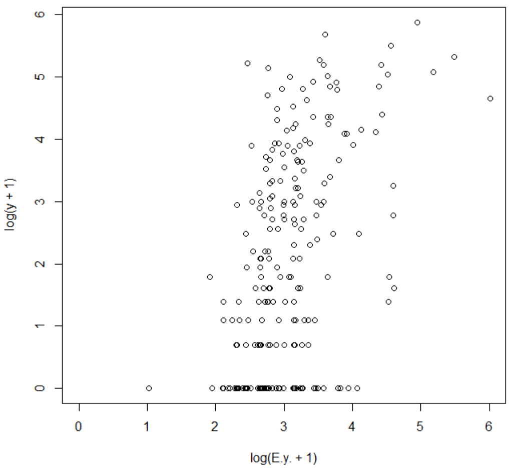


The top two plots are just standard residual plots (with raw and standardized residuals) and they don't help diagnose the problem too much. The bottom two plots begin to suggest something that we could use to diagnose what's wrong, but the presence of some pretty big outliers makes it hard to see what's actually happening.

Let's try a scatter plot with $E[y_i]$ on the x-axis, and y_i on the y-axis. After I made the plot on the left, I tried taking the log of both $E[y_i]$ and y_i to get a better sense of what is going on (note that I added 1 before taking the log, to avoid problems with taking the log of zero. This won't change the pattern in the data). This produces the plot on the right.

```
> plot(E.y.,y)
> plot(log(E.y.+1),log(y+1),xlim=c(0,6))
```





There are many interesting things to see in these plots, but the most striking is that there is a long string of zeros in the observed counts y (if y is zero, then $log(y + 1)$ will still be zero!), corresponding to values of $log(E[y] + 1)$ that range from 1 to 4 or so. There seem to be too many zero's in the observed data, for the Poisson regression model!

This leads us to consider modeling the zeros, and the other counts, separately, as suggested in the class notes.