Generalized Linear Moels Application in R

```
library(MASS)
library(effects)
library(car)
library(ggplot2)
```

Illustration

Poisson Regression

We will return to the HELP data set and the model for predicting number of drinks that we examined previously.

```
helpdata <- read.csv("help.csv")
```

```
mp <- glm(i1 ~ substance + age, data=helpdata, family=poisson())
summary(mp)
```

```
##
## Call:
## glm(formula = i1 ~ substance + age, family = poisson(), data = helpdata)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -8.111  -3.767  -1.248   1.176  16.225
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.885287   0.058283  49.504   <2e-16 ***
## substancecocaine -0.830130   0.027700 -29.968   <2e-16 ***
## substanceheroin  -1.130389   0.033886 -33.359   <2e-16 ***
## age               0.012668   0.001453   8.717   <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: 8898.9  on 452  degrees of freedom
## Residual deviance: 6754.8  on 449  degrees of freedom
## AIC: 8464.2
##
## Number of Fisher Scoring iterations: 6
```

```
confint(mp)
```

```
## Waiting for profiling to be done...
```

```
##                         2.5 %     97.5 %
## (Intercept)       2.770992180  2.9994597
## substancecocaine -0.884638154 -0.7760479
## substanceheroin  -1.197251468 -1.0644093
## age               0.009814905  0.0155114
```

```
exp(coef(mp))
```

```
##      (Intercept) substancecocaine  substanceheroin              age
##       17.9087013        0.4359928        0.3229077        1.0127485
```

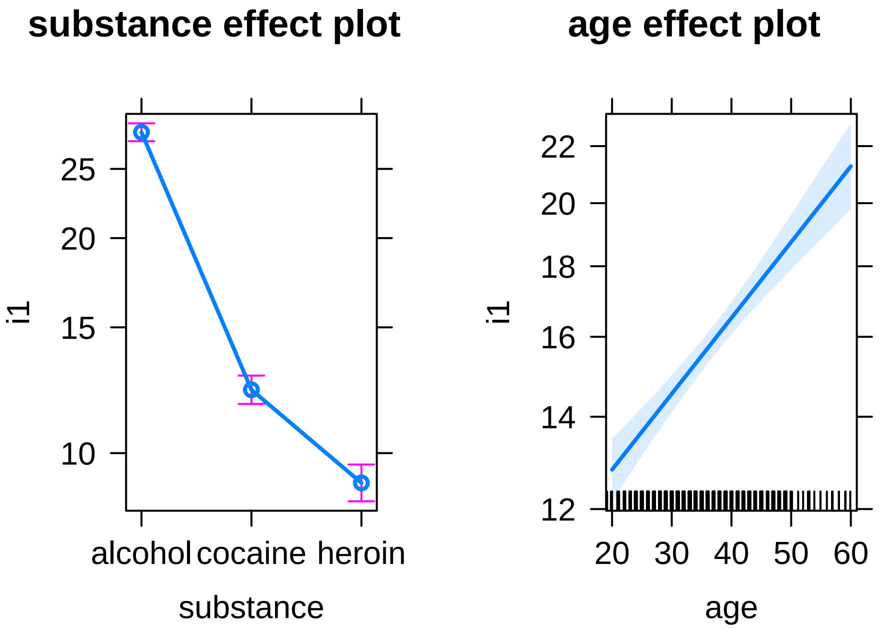
```
exp(confint(mp))
```

```
## Waiting for profiling to be done...
```

```
##                       2.5 %     97.5 %
## (Intercept)      15.9744757 20.0746875
## substancecocaine  0.4128635  0.4602213
## substanceheroin   0.3020232  0.3449315
## age               1.0098632  1.0156323
```

For a one year increase in age (controlling the others), the log of the mean number of drinks will increase by 0.013 and the mean itself will increase by a factor of 1.013 (1.3% increase). If cocaine is the preferred substance, then the log of the mean number of drinks decreases by -.83 and the mean itself would decrease by a factor of .436 (56.4% decrease). Likewise, if heroin is the preferred substance, then the log of mean number of drinks decreases by -1.13 and the mean itself would decrease by a factor of .323 (67.7% decrease).

```
plot(allEffects(mp))
```



The model does not fit the data.

```
1-pchisq(6754.8, 449)
```

```
## [1] 0
```

```
1-pchisq(deviance(mp), df.residual(mp)) # does the same thing
```

```
## [1] 0
```

We can also compute Pearson's χ^2

```
pr <- residuals(mp, "pearson")
sum(pr^2)
```

```
## [1] 8005.507
```

```
1-pchisq(sum(pr^2), df.residual(mp))
```

```
## [1] 0
```

Do we have overdispersion?

```
phi <- sum(pr^2)/df.residual(mp)
phi
```

```
## [1] 17.82964
```

If we estimate ϕ by dividing Pearson's χ^2 by its degrees of freedom, we find that it is much larger than 1.

```
sqrt(phi)
```

```
## [1] 4.222515
```

We could adjust the standard errors by multiplying them by 4.2225 but R will do this for us if we use `family=quasipoisson`. Thus, we should expect that the standard errors from the quasi-poisson model will be inflated by a factor of 4.2225 compared to the standard errors from the Poisson model.

```
mqp <- glm(i1 ~ substance + age, data=helpdata, family=quasipoisson())
summary(mqp)
```

```
##
## Call:
## glm(formula = i1 ~ substance + age, family = quasipoisson(),
##     data = helpdata)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -8.111  -3.767  -1.248   1.176  16.225
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.885287   0.246103  11.724  < 2e-16 ***
## substancecocaine -0.830130   0.116966  -7.097 5.00e-12 ***
## substanceheroin  -1.130389   0.143083  -7.900 2.17e-14 ***
## age               0.012668   0.006136   2.064   0.0395 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 17.82964)
##
##     Null deviance: 8898.9  on 452  degrees of freedom
## Residual deviance: 6754.8  on 449  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

The estimates did not change but the std. errors are larger and therefore the test statistics and p values changed. However, all the coefficients are still statistically significant.

Negative Binomial

We will use the `glm.nb()` function from the `MASS` package to fit the Negative Binomial Model.

```r
mnb <- glm.nb(i1 ~ substance + age, data=helpdata)
summary(mnb)
```

```
## 
## Call:
## glm.nb(formula = i1 ~ substance + age, data = helpdata, init.theta = 0.801907659, 
##     link = log)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.4390  -1.0490  -0.2781   0.2169   2.9102  
## 
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)    
## (Intercept)       2.985736   0.290454  10.280  < 2e-16 ***
## substancecocaine -0.827921   0.129252  -6.405  1.5e-10 ***
## substanceheroin  -1.139235   0.139240  -8.182  2.8e-16 ***
## age               0.009951   0.007267   1.369    0.171    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Negative Binomial(0.8019) family taken to be 1)
## 
##     Null deviance: 632.69  on 452  degrees of freedom
## Residual deviance: 539.65  on 449  degrees of freedom
## AIC: 3430.6
## 
## Number of Fisher Scoring iterations: 1
## 
## 
##               Theta:  0.8019 
##           Std. Err.:  0.0581 
## 
##  2 x log-likelihood:  -3420.5960 
```

Now the age coefficient is not statistically significant. Theta corresponds to $1/\sigma^2$, an estimated variance of:

```r
1/mnb$theta
```

```
## [1] 1.247026
```

It is greater than 0, thus, the variance is larger than the mean.

This model does not fit well. The p value for the residual deviance is less than .05.

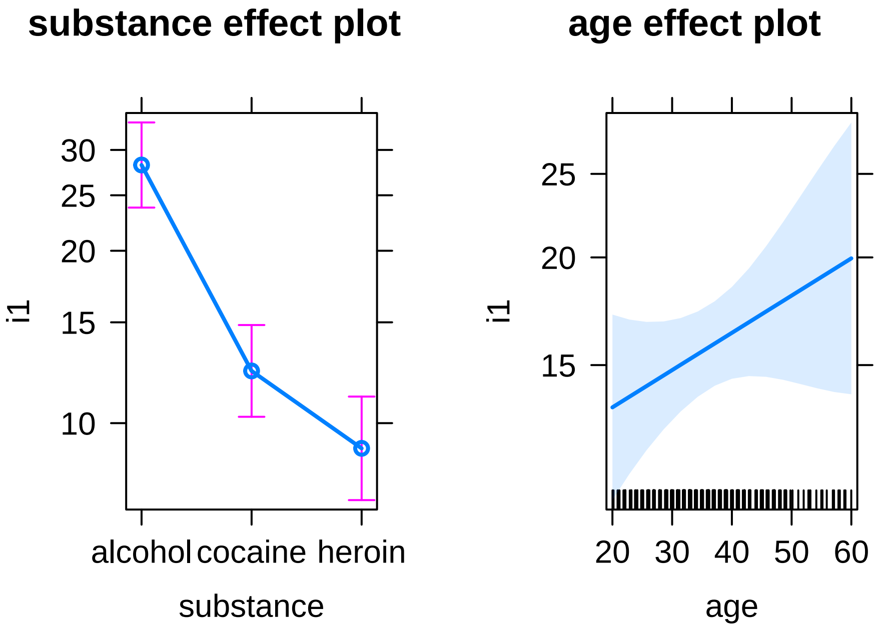
```r
1-pchisq(deviance(mnb), df.residual(mnb))
```

```
## [1] 0.002089888
```

```r
allEffects(mnb)
```

```
##  model: i1 ~ substance + age
## 
##  substance effect
## substance
##  alcohol  cocaine   heroin
## 28.23438 12.33720  9.03680
##
##  age effect
## age
##       20       30       40       50       60
## 13.39820 14.80012 16.34873 18.05937 19.94901
```

```
plot(allEffects(mnb))
```



Let's compare parameter estimates and standard errors under the Poisson, over-dispersed Poisson, and negative binomial models.

```
# you do not need to use this code for anything
se <- function(model) sqrt(diag(vcov(model)))
round(data.frame(
   p=coef(mp),q=coef(mqp),nb=coef(mnb),
   se.p=se(mp),se.q=se(mqp),se.nb=se(mnb)),4)
```

```
##                          p       q      nb   se.p   se.q  se.nb
## (Intercept)         2.8853  2.8853  2.9857 0.0583 0.2461 0.2905
## substancecocaine -0.8301 -0.8301 -0.8279 0.0277 0.1170 0.1293
## substanceheroin  -1.1304 -1.1304 -1.1392 0.0339 0.1431 0.1392
## age               0.0127  0.0127  0.0100 0.0015 0.0061 0.0073
```

The negative binomial estimates are not very different from those based on the Poisson model, and both sets would led to the same conclusions.

Poisson regression underestimates the standard errors. We see that the standard errors for the over-dispersed Poisson and negative binomial models are larger than those of the ordinary Poisson model.

Zero-inflation

The observed proportion of of individuals with 0 drinks in the sample is:

```
zobs <- helpdata$i1 == 0
mean(zobs)
```

```
## [1] 0.1501104
```

The predicted probability of 0 drinks from the Poisson model is:

```
zpoi <- exp(-exp(predict(mp)))
mean(zpoi)
```

```
## [1] 5.642243e-05
```

This is because the inverse link is e^{η_i} and the probability for observing a 0 for the Poisson model is:

$$Pr(Y = 0) = \mu^y \times \frac{e^{-\mu}}{y!} = e^{-\mu} = e^{-e^{\eta}}$$

And the predicted probability of 0 drinks from the negative binomial model is:

```
ab <- mnb$theta
munb <- exp(predict(mnb))
znb <- (ab/(munb + ab))^ab
mean(znb)
```

```
## [1] 0.09487808
```

The inverse link for the negative binomial is the same as that for the Poisson model but the probability for observing a 0 is:

$$Pr(Y = 0) = \frac{\Gamma(\alpha + y)}{y!\Gamma(\alpha)} \frac{\beta^{\alpha}\mu^y}{(\mu + \beta)^{\alpha+y}} = \frac{\beta^{\alpha}}{(\mu + \beta)^{\alpha}} = (\beta/(e^{\eta} + \beta))^{\alpha}$$

Also, recall that $\alpha = \beta = 1/\sigma^2 = \theta$.

Thus, neither the Poisson or the negative binomial models are predicting the proportion of zeros in our actual data very well.

We can fit a zero-inflated Poisson model using the `zeroinfl()` function from the `pscl` package.

```
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
mzip <- zeroinfl(i1 ~ substance + age, data=helpdata)
summary(mzip)
```

```
##
## Call:
## zeroinfl(formula = i1 ~ substance + age, data = helpdata)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -4.4449 -1.5248 -0.7000  0.7568 15.4627
##
## Count model coefficients (poisson with log link):
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        3.053250   0.058613  52.092  < 2e-16 ***
## substancecocaine -0.729014   0.027790 -26.233  < 2e-16 ***
## substanceheroin  -0.760263   0.033601 -22.627  < 2e-16 ***
## age               0.009052   0.001465   6.178  6.5e-10 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.15294    0.85347  -2.523  0.01165 *
## substancecocaine  1.52840    0.51716   2.955  0.00312 **
## substanceheroin   2.75092    0.49754   5.529 3.22e-08 ***
## age              -0.03723    0.01993  -1.868  0.06182 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 1
## Log-likelihood: -3530 on 8 Df
```

Age is not a statistically significant predictor of being in the "always zero" class; however, substance preference is. Age is, however, a statistically significant predictor of mean number of drinks for those not in the "always zero" class, as is substance preference.

The `effects` package functions do not work with zero-inflated Poisson models, so we can not examine effects plots for this model.

One way to determine how well the model is doing is to compute the predicted probability of 0 drinks based on the model.

We can compute predicted probability of being in the "always zero" class and μ for the count model and then calculate the combined probability of zero drinks from both models.

```
p <- predict(mzip, type="zero")
mu <- predict(mzip, type="count")

zip <- p + (1-p)*exp(-mu)
mean(zip)
```

```
## [1] 0.150111
```

The model predicts that 15% of individuals in the study will have no drinks, which is the same as the percentage in the observed data.

Model Selection

The Poisson, negative binomial, and zip models are not nested and we cannot compare them using the log-likelihood difference test or incremental F tests that we've used before. We can instead use Akaike Information Criterion (AIC), which can be obtained using the function `AIC()`. Lower values are better.

```
AIC(mp)
```

```
## [1] 8464.184
```

```
AIC(mnb)
```

```
## [1] 3430.596
```

```
AIC(mzip)
```

```
## [1] 7075.484
```

In this case, we see that the AIC for the Poisson model is 8464.18 and the AIC for the zero-inflated Poisson, which has many more parameters, is 7075.48. The negative binomial model, which has only one more parameter than the Poisson model, has the lowest AIC, 3430.60. Based on the AIC, we would choose the negative binomial model; however, the zero-inflated model did a much better job in that it was much closer to predicting the proportion of zeros actually observed in the data. It is a bit of a tough call here, but what is clear is that Poisson model is not adequate.

You can also check the Bayesian Information Criterion (BIC), which can be obtained using the function `BIC()`. Lower values are better.

```
BIC(mp)
```

```
## [1] 8480.648
```

```
BIC(mnb)
```

```
## [1] 3451.176
```

```
BIC(mzip)
```

```
## [1] 7108.411
```

Note that we cannot obtain an AIC for the overdispersed Poisson model because it is fit with quasi-likelihood.

Diagnostics

Let's examine some diagnostics for the various models. Note that the `car` package functions do not work for zero-inflated Poisson models.

The are quite a few Bonferroni corrected statistically significant outliers for the Poisson model.

```
outlierTest(mp)
```

```
##      rstudent unadjusted p-value Bonferroni p
## 74  16.371102         3.0759e-60   1.3934e-57
## 244 13.768684         3.9331e-43   1.7817e-40
## 286 13.588572         4.6812e-42   2.1206e-39
## 288 11.703368         1.2249e-31   5.5490e-29
## 180 11.079005         1.5863e-28   7.1857e-26
## 431 10.886636         1.3348e-27   6.0466e-25
## 443  9.970666         2.0485e-23   9.2797e-21
## 94   9.893022         4.4635e-23   2.0220e-20
## 177  9.881284         5.0186e-23   2.2734e-20
## 320  9.521422         1.7083e-21   7.7384e-19
```

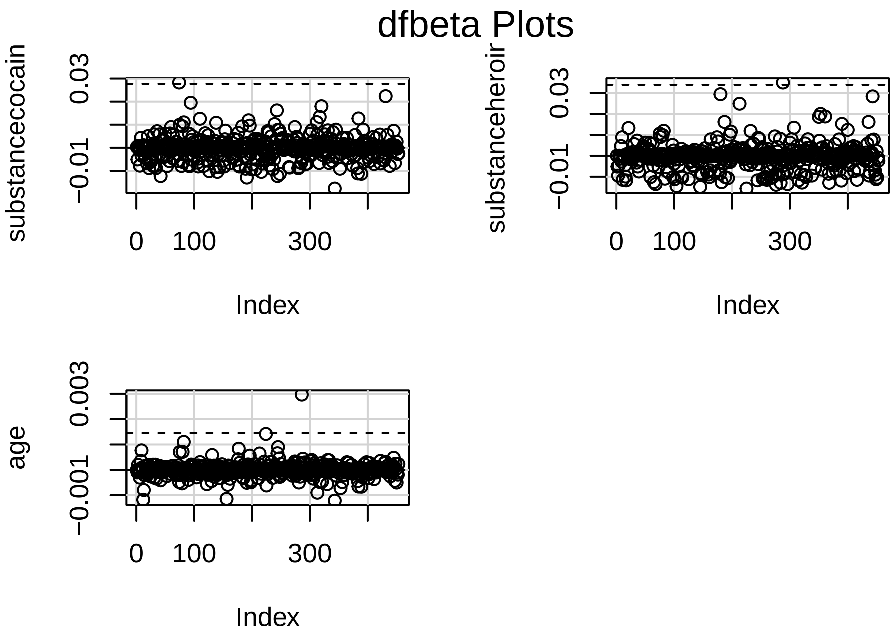
However, there are none for the negative binomial model.

```
outlierTest(mnb)
```

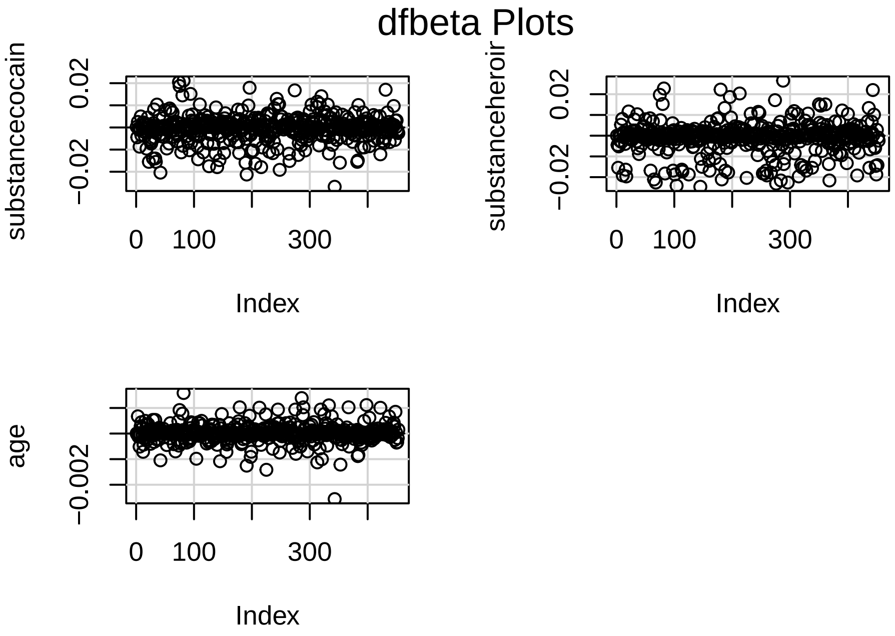
```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##    rstudent unadjusted p-value Bonferroni p
## 74 2.721775          0.0067463           NA
```

There is perhaps one individual who influences the regression coefficient for age in the Poisson model, although this individual only changes the coefficient by .003. The dfbetas look fine for the negative binomial model.

```
dfbetaPlots(mp)
```

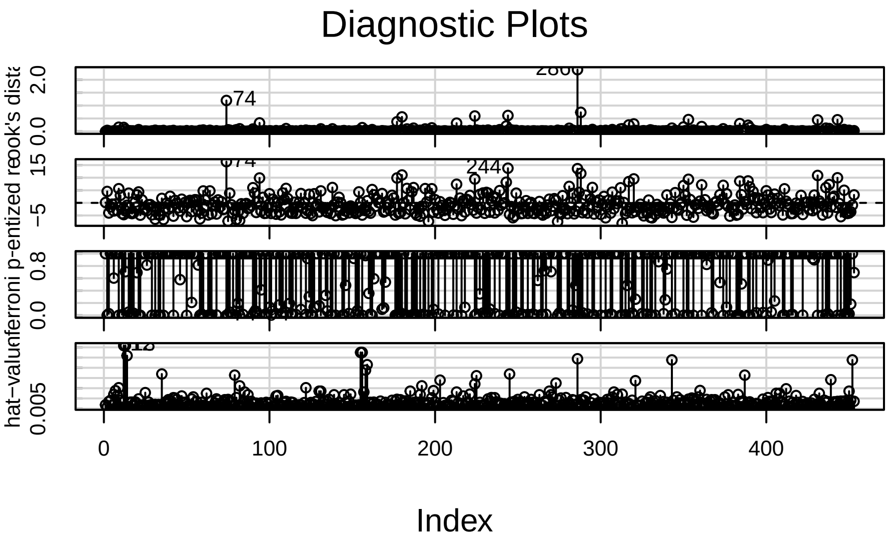


```
dfbetaPlots(mnb)
```

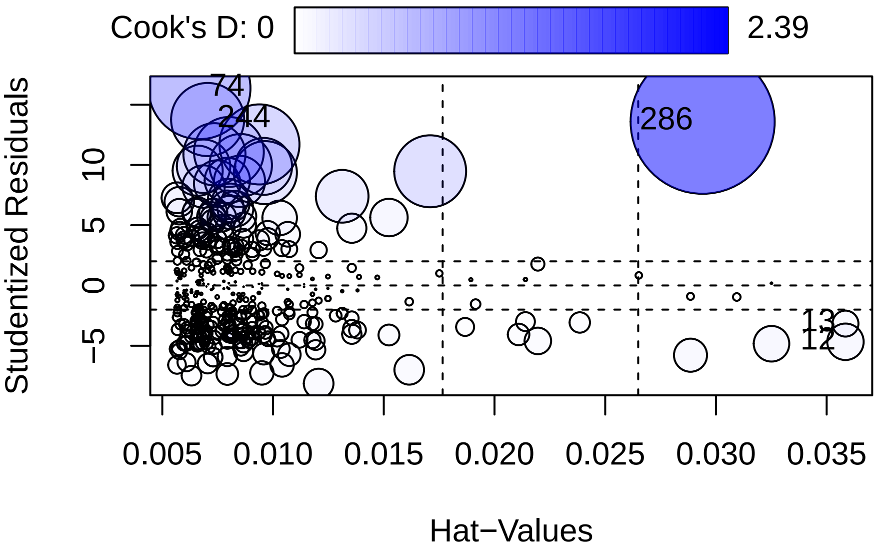


In examining Cook's Distance for the Poisson model, we again have quite a few potentially influential observations.

```
influenceIndexPlot(mp)
```



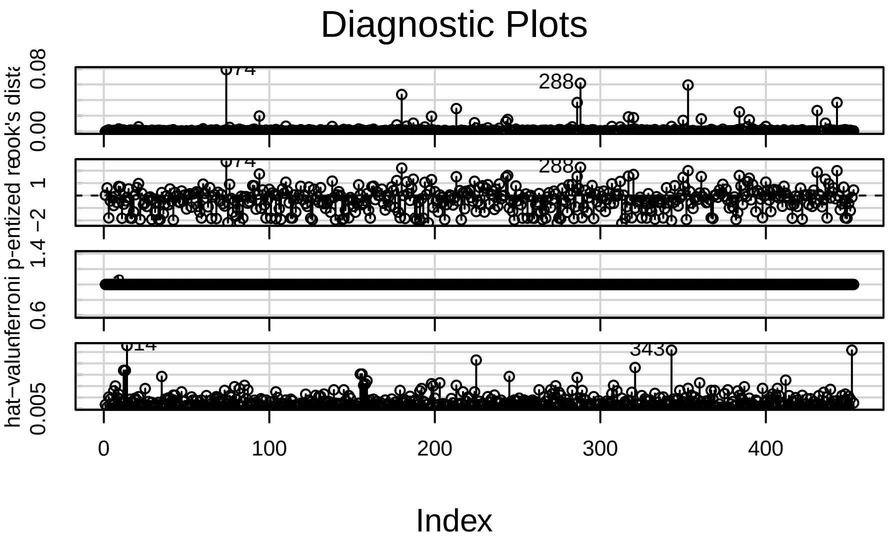
```
influencePlot(mp)
```



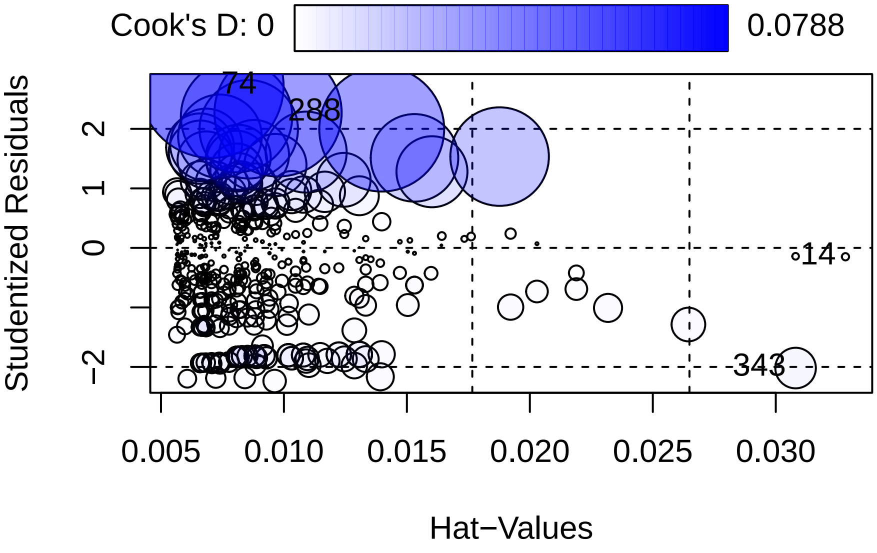
```
##        StudRes         Hat      CookD
## 12  -4.672804 0.035849350 0.1529556
## 13  -3.163034 0.035849350 0.0776252
## 74  16.371102 0.006678772 1.1968776
## 244 13.768684 0.007046633 0.6153776
## 286 13.588572 0.029401037 2.3897839
```

There might be a few influential observations for the negative binomial model too but they are not nearly as large (note the y-axis scale).

```
influenceIndexPlot(mnb)
```



```r
influencePlot(mnb)
```



```
##        StudRes         Hat        CookD
## 14  -0.1484884 0.032834248 0.0002003954
## 74   2.7217746 0.007056672 0.0788165764
## 288  2.2684789 0.009760443 0.0614949183
## 343 -2.0182613 0.030806048 0.0062525895
```

Summary

GLMs consist of three components:

(a) A random component specifying the conditional distribution of the response variable Y given the explanatory variables, traditionally a member of an exponential family — the normal (Gaussian), binomial, Poisson, gamma, or inverse-Gaussian families of distributions.

For distributions in exponential families, the conditional variance of Y is a function of μ, the mean of Y, and of a dispersion parameter ϕ; in the binomial and Poisson families, ϕ is fixed to 1.

(b) A linear predictor, $\eta_i = \beta_0 + \beta_1 X_{i1} + \ldots + \beta_k X_{ik}$

(c) A link function, $g(\mu_i) = \eta_i$, which transforms the expectation of the response to the linear predictor; the inverse of the link is the mean function, $g^{-1}(\eta_i) = \mu_i$.

Traditional GLMs are fit to data by maximum likelihood. The log-likelihood for the model, maximized over the regression coefficients, is

$$\log_e L_0 = \sum_{i=1}^{n} \log_e p(\widehat{\mu}_i, \phi; y_i)$$

where $p(\cdot)$ is the probability density function corresponding to the family employed, y_i are the observed values of the response variable, and $\widehat{\mu}_i$ are the fitted values of the response.

The deviance under a fitted model is $G_0^2 = 2(\log_e L_1 - \log_e L_0)$, where L_1 is the maximized likelihood for a saturated model that dedicates one parameter to each observation, and L_0 is the maximized likelihood under the model in question.

The scaled deviance is $G_0^2/\widehat{\phi}$, where $\widehat{\phi}$ is an estimate of the dispersion.

In analogy to incremental F-tests in an analysis of variance for linear models, differences in deviance may be used for likelihood-ratio tests in GLMs; for models with a dispersion parameter, F-tests are also available.

Wald tests for individual coefficients are produced by dividing the estimated coefficients by their standard errors.

The binomial family is used for dichotomous response variables. Pairing the binomial family with the logit link produces the logistic regression model.

The Poisson family is often used to analyze count data. The canonical link for the Poisson family is the log link.

Over-dispersed binomial and Poisson models introduce a dispersion parameter ϕ that is not fixed to 1; these models are fit by quasi-likelihood.

Many standard linear model diagnostics may be generalized to GLMs. These include hat-values, studentized residuals, and Cook's distance (among others).