Count Regression Notes

Daniel J. Eck

Contents

We will suppose that we have a sample of data (y_i, x_i), $i = 1, \ldots, n$ where y_i is a non-negative integer response variable and x_i is a fixed vector of predictors taking values in $\mathbb{R}^p$. We will start with Poisson regression, a count regression model arising from exponential family. Other modeling approaches will also be discussed. Namely, overdispersion, negative binomial regression, and zero-inflated count models.

Poisson regression model with log link

The Poisson regression model is another of one of the most widely used and studied GLMs in practice. A Poisson regression model is sometimes known as a log-linear model, especially when used to model contingency tables. The Poisson regression model is used for analyzing a discrete count response variable, $y_i \in \{0, 1, 2, \ldots\}$ or a contingency table. The Poisson regression model allows for users to model the rate as a function of covariates. The log link arises as the canonical link.

For a count response variable Y and a vector of predictors X, let $\mu(x) = \mathrm{E}(Y|X = x)$. The Poisson regression model is then

$$\mu(x) = \mathrm{E}(Y|X = x) = \exp\left(x^T \beta\right). \tag{1}$$

Equivalently,

$$\log\left(\mu(x)\right) = x^T \beta.$$

In vector notation, we can express the above as

$$\boldsymbol{\mu} = \exp(M\beta) \quad \text{and} \quad \log(\boldsymbol{\mu}) = M\beta$$

where the above $\exp(\cdot)$ and $\log(\cdot)$ operations are understood as componentwise operations. To see where the log link comes from in the specification of $\mu(x)$ comes from consider the log likelihood of the Poisson distribution with rate parameter μ_i for each subject

$$l\left(\beta\right) \propto \sum_{i=1}^{n} y_i \log(\mu_i) - \sum_{i=1}^{n} \mu_i$$
$$= \sum_{i=1}^{n} y_i \theta_i - \sum_{i=1}^{n} g(\theta_i),$$

where
$$\theta_i = \log\left(\mu_i\right) \qquad \text{and} \qquad \mu_i = \exp(\theta_i) = g(\theta_i).$$

We see that the Poisson regression model with log link is the same as the canonical linear submodel of an exponential family with $\theta_i = x_i^T \beta$ which in vector notation is $\theta = M\beta$. Putting this together, we have that

$$\mathrm{E}(Y|X = x) = g\left(x^T\beta\right) \qquad \text{and} \qquad g^{-1}\left(\mathrm{E}(Y|X = x)\right) = x^T\beta$$

where the link function g^{-1} is the logarithmic function, the inverse-link function (change of parameters map) is the exponential function. Hence the name log-linear models.

Therefore, a linear function of the canonical submodel parameter vector is linked to the mean of the Poisson distribution through a change-of-parameter mapping $g(\theta)$. As stated before, this is the basis of exponential family generalized linear models with link function g^{-1}.

Example: Gala data

```
rm(list = ls())
library(tidyverse)
library(faraway)
```

We will demonstrate Poisson regression modeling on the Galapagos data frame in the `faraway` package. This data frame consists of $n = 30$ observations and 7 variables in total.

For 30 Galapagos Islands, we have a count of the number of plant species found on each island and the number that are endemic to the island. We also have five geographic variables for each island. A few missing values have been filled in for simplicity. We will model the number of species using Poisson regression using the glm function in R. The Endemic variable is thrown out since it won't be used in this analysis. We create a discrete size variable based on the Area variable for demonstration purposes.

```
gala = gala %>%
  mutate(Size = as.factor(1 + ifelse(Area > 1,1,0) + ifelse(Area > 25,1,0)))
m1 = glm(Species ~ Elevation + Nearest + Scruz + Adjacent + Size,
         family = "poisson", data = gala, x = TRUE)
```

Now let's unpack the glm function call above. We decided that we wanted to fit an exponential family regression model with log likelihood taking the general form

$$l(\beta) = \langle M^T y, \beta \rangle - c(M\beta),$$

where y is the vector of responses β is the submodel canonical statistic vector corresponding to the model matrix M specified by the formula in the glm function call above. The first few rows of M are displayed below

```
M = m1$x
head(M)
```

```
##              (Intercept) Elevation Nearest Scruz Adjacent Size2 Size3
## Baltra                1       346     0.6   0.6     1.84     0     1
## Bartolome             1       109     0.6  26.3   572.33     1     0
## Caldwell              1       114     2.8  58.7     0.78     0     0
## Champion              1        46     1.9  47.4     0.18     0     0
## Coamano               1        77     1.9   1.9   903.82     0     0
## Daphne.Major          1       119     8.0   8.0     1.84     0     0
```

The specific log likelihood for the Poisson regression model can then be written as

$$l(\beta) \propto \sum_{i=1}^{n} y_i x_i^T \beta - \exp\left(x_i^T \beta\right)$$

where the x_is are the rows of the design matrix M and the y_is are the components of the response vector y (the Species variable corresponding to the number of species on each of the islands). The glm function then performs a Fisher scoring based optimization routine to maximize the above likelihood. We can view summary information for $\hat{\beta}$ and the fitting process using the summary function

```
summary(m1)
```

```
##
## Call:
## glm(formula = Species ~ Elevation + Nearest + Scruz + Adjacent +
##      Size, family = "poisson", data = gala, x = TRUE)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.790e+00  8.108e-02  34.410  < 2e-16 ***
## Elevation    9.361e-04  5.402e-05  17.329  < 2e-16 ***
## Nearest      6.469e-03  1.748e-03   3.702 0.000214 ***
## Scruz       -6.266e-03  6.268e-04  -9.997  < 2e-16 ***
## Adjacent    -2.858e-04  2.961e-05  -9.652  < 2e-16 ***
## Size2        1.128e+00  9.535e-02  11.826  < 2e-16 ***
## Size3        2.059e+00  9.419e-02  21.856  < 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: 3510.73  on 29  degrees of freedom
## Residual deviance:  594.18  on 23  degrees of freedom
## AIC: 769.01
##
## Number of Fisher Scoring iterations: 5
```

The Estimate column in the above summary table is $\hat{\beta}$. The standard error column as estimates of the square root of the variances of the estimated submodel canonical statistic vector $\hat{\beta}$. Recall from the asymptotic theory of maximum likelihood estimation that

$$\sqrt{n}(\hat{\beta} - \beta) \xrightarrow{d} N(0, \Sigma^{-1}),$$

where Σ^{-1} is the inverse of the Fisher information matrix. We can extract these same standard errors using the vcov function

```
sqrt(diag(vcov(m1)))
```

```
##  (Intercept)     Elevation       Nearest        Scruz       Adjacent         Size2
## 8.107577e-02 5.402068e-05 1.747556e-03 6.268325e-04 2.960794e-05 9.535081e-02
##        Size3
## 9.419199e-02
```

These values are the same as those in the Std. Error column in the above summary table

```
all.equal(summary(m1)$coef[, 2], sqrt(diag(vcov(m1))))
```

```
## [1] TRUE
```

Inference for β in Poisson regression

The Galapagos example connects the exponential family that we have developed to R-based data analysis. We motivated this connection in the logistic regression notes. In Poisson regression the link function is log, and thus $\log(\mu(x)) = x_i^T\beta$. Therefore, a unit increase in one predictor variables x_j corresponds to an increase of β_j (estimated by $\hat{\beta}_j$) in the log mean with everything else being held fixed. See the summary table below for estimates of β in our Galapagos example

```
summary(m1)
```

```
##
## Call:
## glm(formula = Species ~ Elevation + Nearest + Scruz + Adjacent +
##     Size, family = "poisson", data = gala, x = TRUE)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.790e+00  8.108e-02  34.410  < 2e-16 ***
## Elevation    9.361e-04  5.402e-05  17.329  < 2e-16 ***
## Nearest      6.469e-03  1.748e-03   3.702 0.000214 ***
## Scruz       -6.266e-03  6.268e-04  -9.997  < 2e-16 ***
## Adjacent    -2.858e-04  2.961e-05  -9.652  < 2e-16 ***
## Size2        1.128e+00  9.535e-02  11.826  < 2e-16 ***
## Size3        2.059e+00  9.419e-02  21.856  < 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: 3510.73  on 29  degrees of freedom
## Residual deviance:  594.18  on 23  degrees of freedom
## AIC: 769.01
##
## Number of Fisher Scoring iterations: 5
```

Deviance testing and AIC/BIC

We can use the deviance and degrees of freedom objects to perform the likelihood ratio test to determine whether or not the main effects model fits the data well.

```r
## test against intercept only model
pchisq(m1$null.deviance - m1$deviance, df = m1$df.null - m1$df.residual,
       lower = FALSE)
```

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

```r
m_null = glm(Species ~ 1, family = "poisson", data = gala, x = TRUE)
anova(m_null, m1, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: Species ~ 1
## Model 2: Species ~ Elevation + Nearest + Scruz + Adjacent + Size
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        29     3510.7
## 2        23      594.2  6   2916.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let's consider the smaller model that ignores the Elevation variable. A likelihood ratio test shows that the larger model is preferable at any reasonably chosen significance level α.

```r
m_small = glm(Species ~ Nearest + Scruz + Adjacent + Size,
          family = "poisson", data = gala, x = TRUE)

## built in likelihood ratio test using anova.glm
anova(m_small, m1, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: Species ~ Nearest + Scruz + Adjacent + Size
## Model 2: Species ~ Elevation + Nearest + Scruz + Adjacent + Size
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        24     878.14
## 2        23     594.18  1   283.96 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## perform the above directly (different machine tolerances)
pchisq(m_small$deviance - m1$deviance, df = 1, lower = FALSE)
```

```
## [1] 1.029422e-63
```

```r
## AIC/BIC
AIC(m_small, m1)
```

```
##         df       AIC
## m_small  6 1050.9666
## m1       7  769.0063
```

```r
BIC(m_small, m1)
```

```
##         df       BIC
## m_small  6 1059.3738
## m1       7  778.8146
```

Further testing

In this example we see that the levels of the Size variable are statistically significant at any reasonable error threshold α. Suppose instead that we wanted to test if large islands are expected to have a different number of species than medium sized islands. Informally, the summary table above suggests that large islands have mores species than medium sized islands, but this is not a formal comparison. Formally, we want to test

$$H_o : \mu_l - \mu_m = 0, \qquad H_a : \mu_l - \mu_m \neq 0,$$

where μ_l and μ_m, respectively, correspond to the mean-value parameters for large and medium-sized islands. We know that $\mu_l = \exp(\beta_l)$ and $\mu_m = \exp(\beta_m)$ where β_l and β_m, respectively, correspond to the canonical parameters for large and medium-sized islands. We can test the above hypothesis using the Delta method

$$\sqrt{n}\left[(\hat{\mu}_l - \hat{\mu}_m) - (\mu_l - \mu_m)\right] \xrightarrow{d} N\left(0, \nabla h(\beta)^T \Sigma^{-1} \nabla h(\beta)\right)$$

where $h(\beta) = \exp(\beta_l) - \exp(\beta_m) = \mu_l - \mu_m$. We obtain the estimate $\hat{\mu}_l - \hat{\mu}_m$ below

```
comp = c(0,0,0,0,0,-1,1)
betahat = m1$coefficients
grad = exp(betahat) * comp
est = sum(grad)
est
```

```
## [1] 4.747314
```

and the estimate of $\nabla h(\beta)$ is the column vector below

```
grad
```

```
## (Intercept)    Elevation      Nearest        Scruz     Adjacent        Size2
##    0.000000     0.000000     0.000000     0.000000     0.000000    -3.088284
##        Size3
##     7.835597
```

The asymptotic variance $\nabla h(\beta)^T \Sigma^{-1} \nabla h(\beta)$ and corresponding standard error are estimated below

```
InvFish = vcov(m1)
asympVar = as.numeric(t(grad) %*% InvFish %*% grad)
asympVar
```

```
## [1] 0.2965522
```

```
SE = sqrt(asympVar)
SE
```

```
## [1] 0.5445661
```

The ratio $(\hat{\mu}_l - \hat{\mu}_m)/\text{se}(\hat{\mu}_l - \hat{\mu}_m)$ is given by

```
est/SE
```

```
## [1] 8.717608
```

and a corresponding 95% confidence interval is

```
est + qnorm(c(0.025,0.975)) * SE
```

```
## [1] 3.679984 5.814644
```

Keep in mind that there are three possible tests that we could have made. If multiple hypotheses are tested, the probability of observing a rare event increases, and therefore, the likelihood of incorrectly rejecting a null hypothesis (Type I error) increases. We can adjust for this using a Bonferroni correction which scales the desired error tolerance by the number of tests in order to align the family-wise error rate of all tests (probability of at least one false rejection) is at the original testing threshold.

```
est + qnorm(c(0.025/3, 1-0.025/3)) * SE
```

```
## [1] 3.443633 6.050994
```

Diagnostics (will revisit)

The diagnostic methods discussed here are similar to those used for linear models with normal errors. However, some adaptions are necessary, and not all diagnostic methods will be applicable.

Residual-based diagnostics

Residuals represent the discrepancy between the model and the observed data, and are essential for exploring the adequacy of the model. In the Gaussian case, the residuals are $\hat{\epsilon} = y - \hat{\mu}$. In Faraway, these are referred to as response residuals for GLMs and they can be used directly to check the constant variance assumption in linear models with Gaussian errors. However, since the variance of the GLM is often not constant and is often a function of the canonical parameter, some modifications to the residuals are necessary.

Pearson residuals: the Pearson residual is comparable to the standardized residual used for linear models and is defined as:

$$r_P = \frac{y - \hat{\mu}}{\sqrt{\widehat{\text{Var}}(\hat{\mu})}}$$

where $\widehat{\text{Var}}$ is the estimated variance under the original exponential family. Notice that $\sum_{i=1}^{n} r_{P,i}^2$ is the Pearson χ^2 statistic, hence the name. And note that Pearson residuals can be skewed for nonnormal responses. A useful result for Pearson residuals is that $\sum_{i=1}^{n} r_{P,i}^2 \approx n - p$. This is because $E(r_{P,i}^2) \approx 1$ and $\sum_{i=1}^{n} r_{P,i}^2 \approx \chi_{n-p}^2$.

Deviance residuals: The deviance residuals are defined by analogy to the Pearson residuals. In other words, we set the deviance residual r_D so that

$$\sum_{i=1}^{n} r_{D,i}^2 = \text{Deviance}.$$

In Poisson regression the deviance residuals are

$$r_{D,i} = \text{sign}(y_i - \hat{\mu}_i) \left[2 \left(y_i \log \left(\frac{y_i}{\hat{\mu}_i} \right) - (y_i - \hat{\mu}_i) \right) \right]^{1/2}.$$

Notice that both types of residuals possess scaling by the estimated mean values or standard deviations in their constructions. Thus, a Poisson regression model has to well-estimate both small and large expected counts. We now revisit the Galapagos data to explore these residuals.

```
## Deviance residuals
head(residuals(m1))
```

```
##        Baltra    Bartolome     Caldwell     Champion      Coamano Daphne.Major
## -10.37226589  -1.51907573  -3.29219582   3.01748621  -3.92316686  -0.05066076
```

```
## Pearson residuals
head(residuals(m1, "pearson"))
```

```
##       Baltra    Bartolome      Caldwell     Champion       Coamano Daphne.Major
##   -8.90760179  -1.45715550  -2.73281460   3.41729331  -3.13259915  -0.05056044
```

For linear models, the residuals vs. fitted values plot is probably the single most valuable graphic. For GLMs, we must decide on the appropriate scale for the fitted values. Usually, it is better to plot on the scale of the linear predictors (θ) than on the fitted responses (μ).

We are looking for features in these residuals vs. fitted values plots. First of all, is there any nonlinear relationship between the residuals and the fitted values? If so, this would be an indication of a lack of fit that might be rectified by a change in the model. For a linear model, we may transform the response variable but this is likely impractical for a GLM since it would change the assumed distribution of the response variable. We might also consider changing the link function, but often this undesirable since the canonical link functions facilitate desirable theoretical properties and yield models which are relatively easy to interpret.

It is best to make a change in the choice of predictors or transformations to these predictors since this involves the least disruption to the GLM theoretical foundations.

The plots below show the residuals as a function of $\hat{\theta}_i = x_i^T \hat{\beta}$.

```
dat = data.frame(theta = as.numeric(M %*% betahat),
                 mu = exp(as.numeric(M %*% betahat)),
                 deviance_residuals = residuals(m1),
                 pearson_residuals = residuals(m1, "pearson")) %>%
  pivot_longer(., cols = deviance_residuals:pearson_residuals,
               names_to = "residuals")
dat$residuals =
  factor(dat$residuals, levels = c("deviance_residuals", "pearson_residuals"),
         labels = c("Deviance", "Pearson"))

ggplot(dat) +
  aes(x = theta, y = value) +
  labs(y = "residuals") +
  geom_point() +
  facet_wrap(~residuals) +
  geom_hline(yintercept = 0, lty = 2, col = "red") +
  theme_minimal()
```
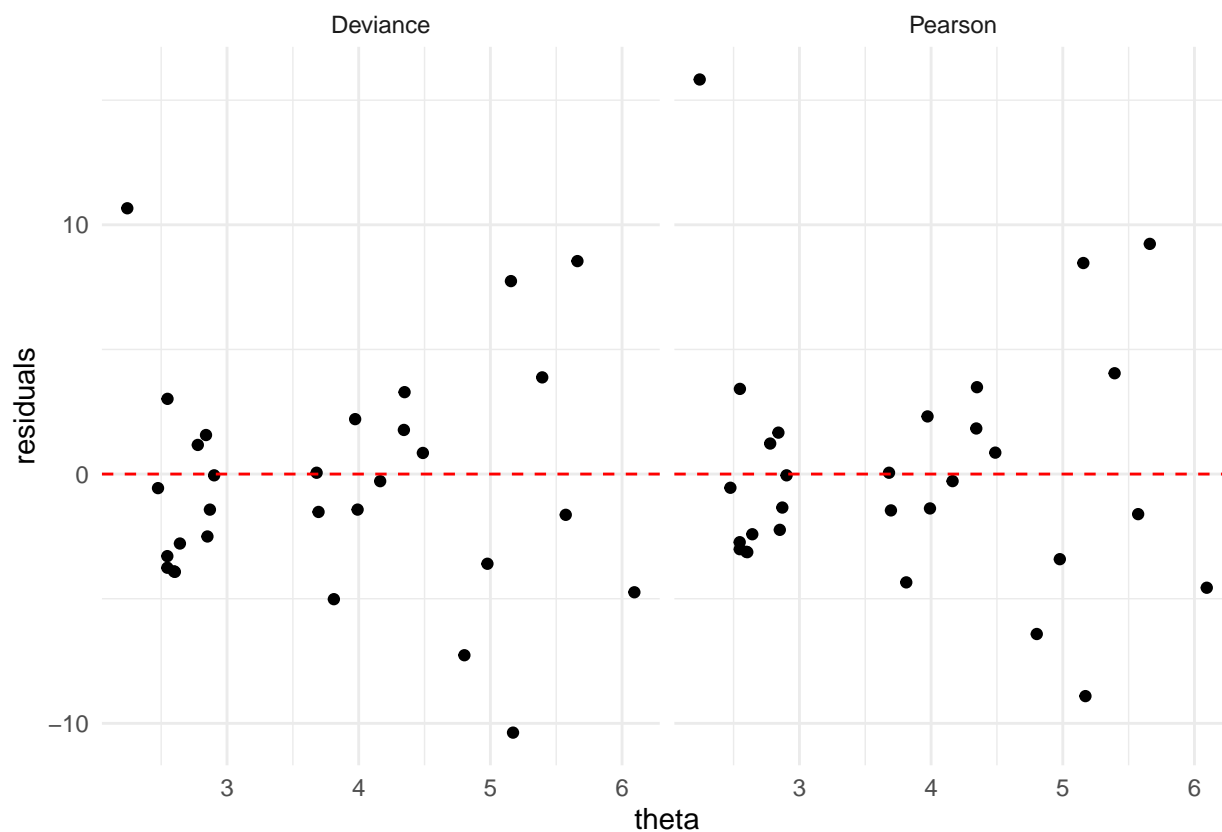
The plots below show the residuals as a function of $\hat{\mu}$.

```r
ggplot(dat) +
  aes(x = mu, y = value) +
  labs(y = "residuals") +
  geom_point() +
  facet_wrap(~residuals) +
  geom_hline(yintercept = 0, lty = 2, col = "red") +
  theme_minimal()
```
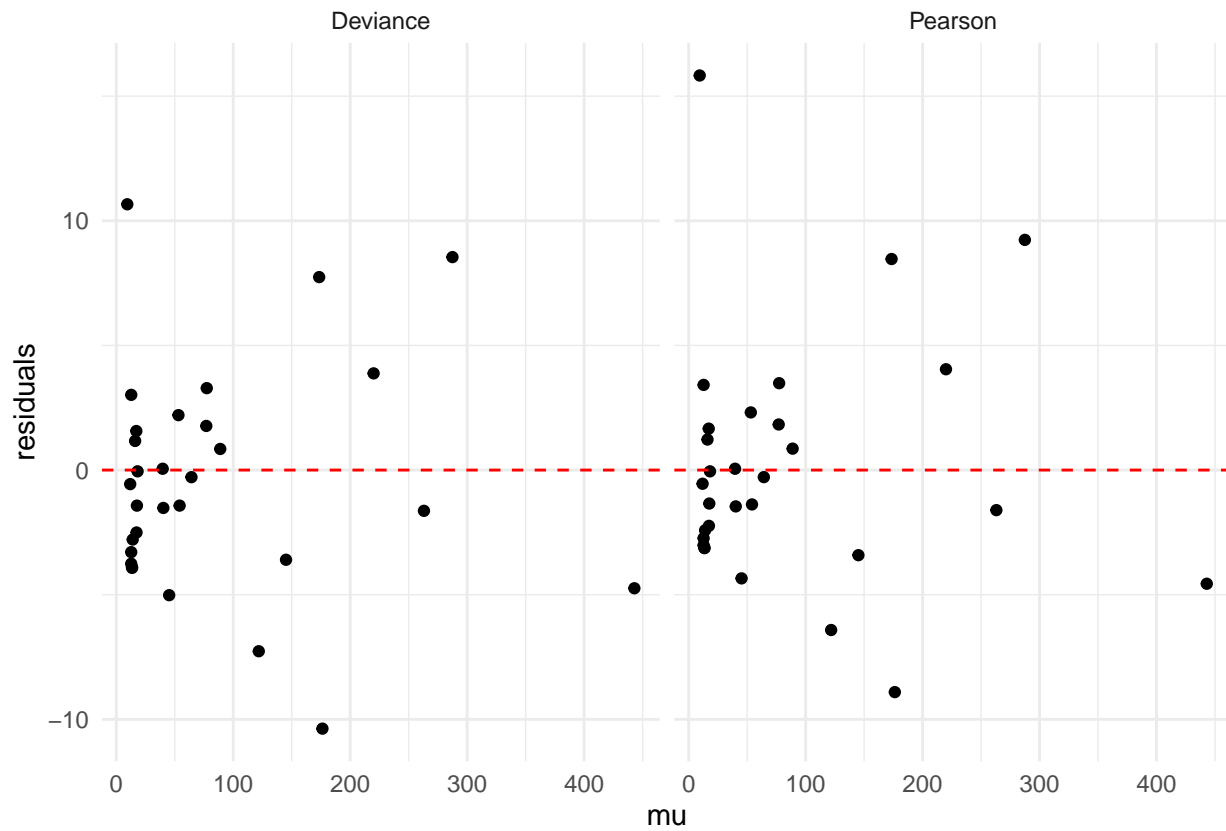
half-normal plot

A half-normal plot of the Deviance residuals of the Poisson model is shown on the below. The half-normal plot depicts points

$$\left(\Phi^{-1} \left(\frac{n+i}{2n+1} \right), r_{D,(i)} \right),$$

where $r_{D,(i)}$ denotes an ordering of deviance residuals so that $r_{D,(1)} \leq \cdots \leq r_{D,(n)}$. The half-normal plot will not necessarily return a straight line relationship because deviance residuals are not necessarily normally distributed. However, this plot is useful for diagnosing outliers.

```
halfnorm(residuals(m1), pch = 19)
```
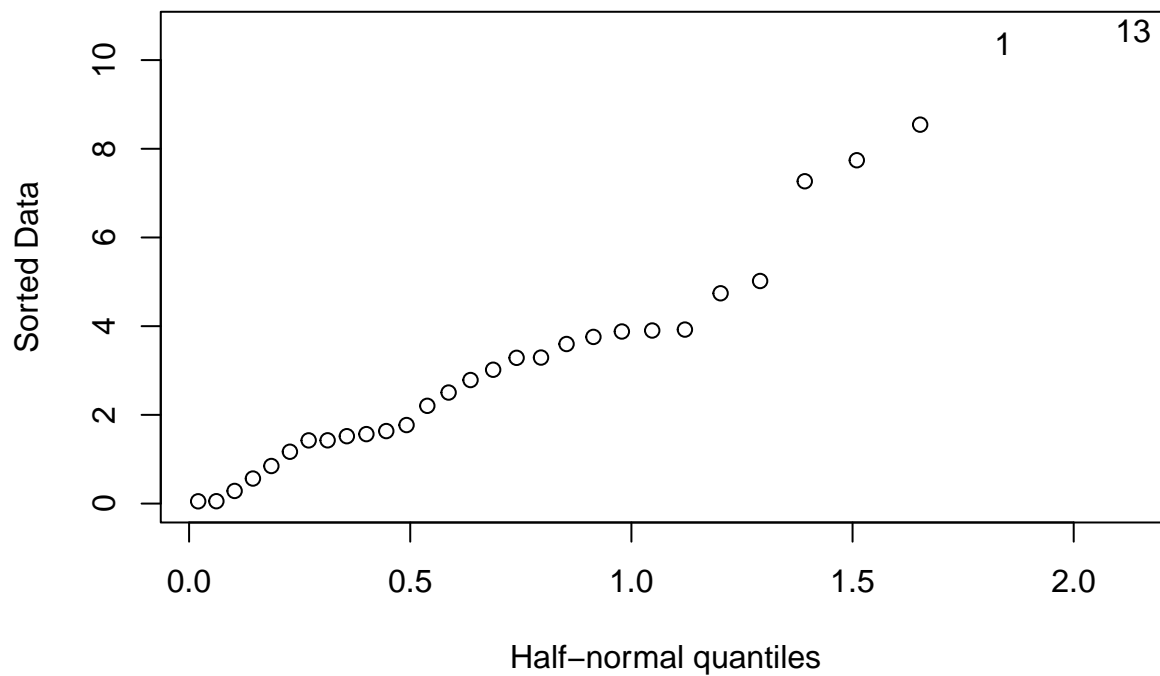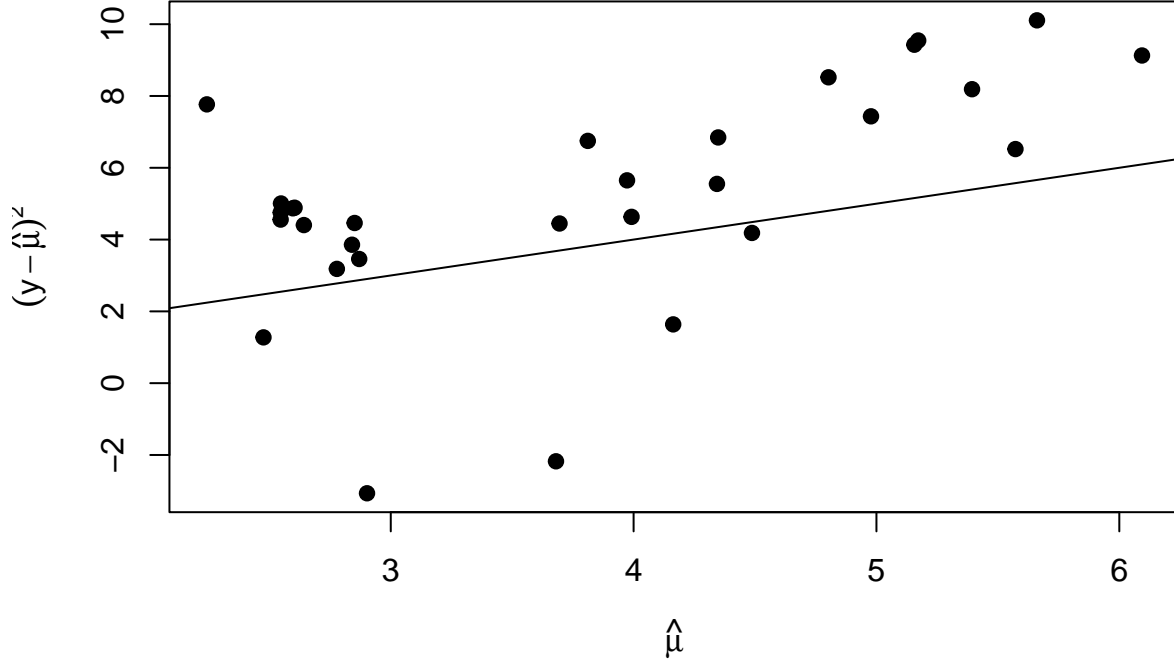
variance vs mean

The relationship between the mean and the variance is shown below. A line showing that the variance increases linearly in the mean (not a perfect slope of 1) is also shown.

```r
plot(log(fitted(m1)),log((gala$Species-fitted(m1))^2),
     main = "variance vs mean (on log scale)",
     xlab= expression(hat(mu)),ylab=expression((y-hat(mu))^2),
     pch = 19)
abline(0,1)
```

variance vs mean (on log scale)

We see that the variance is proportional to, but larger than, the mean. When the variance assumption of the Poisson regression model is broken but the link function and choice of predictors are correct, the estimates of β are consistent, but the standard errors will be wrong.

Overdispersion

The Poisson distribution has only one parameter (mean equals variance) and is therefore not very flexible for empirical fitting purposes. We can generalize by allowing ourselves a dispersion parameter to allow for increased flexibility in various Poisson models. For example, suppose the Poisson response Y has rate λ which is itself a random variable. The tendency to fail for a machine may vary from unit to unit even though they are the same model. We can model this by letting λ be gamma distributed with $E(\lambda) = \mu$ and $\text{Var}(\lambda) = \mu/\phi$. Now Y is negative binomial with mean $E(Y) = \mu$. The mean is the same as the Poisson, but the variance $\text{Var}(Y) = \mu(1 + \phi)/\phi$ which is not equal to μ.

If we know the specific data generating process, as in the above example, we could model the response as a negative binomial or some other more flexible distribution. However, when the mechanism is not known, we can introduce a dispersion parameter ϕ such that $\text{Var}(Y) = \phi E(Y) = \phi\mu$. The case $\phi = 1$ is the regular Poisson regression case, while $\phi > 1$ is overdispersion and $\phi < 1$ is underdispersion.

A common explanation for large deviance (or poor fit) is the presence of a few outliers. When large number of points are identified as outliers, they become unexceptional, and it may be the case that the error distribution is misspecified. In the presence of overdispersion, the exponential family takes on a different functional form

$$f(y|\theta, \phi) = \exp\left(\frac{\langle y, \theta \rangle - c(\theta)}{a(\phi)} - b(y, \phi)\right), \tag{2}$$

where y, θ, and $c(\theta)$ are as before, ϕ is a dispersion parameter, and $b(y, \phi)$ is a function of the data y and the dispersion parameter ϕ. From the perspective of the canonical exponential families that we have motivated throughout, the function $b(y, \phi)$ is similar to the base measure h that was dropped from consideration in log likelihood based arguments that focused on the parameters. Notice that the density (2) is a generalization of the exponential family density which specifies that $a(\phi) = 1$ and $b(y, \phi) = \log(h(y))$.

Note that the dispersion parameter can be estimated using

$$\hat{\phi} = \frac{\sum_{i=1}^{n}(y - \hat{\mu}_i)^2/\hat{\mu}_i}{n - p}.$$

Notice that the estimation of the dispersion and the regression parameters is independent, so choosing a dispersion other than one has no effect on the regression parameter estimates. See pages 149-150 in Agresti [2013] for more details.

We investigate the overdispersed Poisson regression model with respect to the Galapogas data.

```
n = nrow(gala)
p = length(betahat)
y = gala$Species

## estimate dispersion directly
fits = predict(m1, type = "response")
dp = sum((y - fits)^2/fits) / (n - p)
summary(m1, dispersion = dp)
```

```
##
## Call:
## glm(formula = Species ~ Elevation + Nearest + Scruz + Adjacent +
##     Size, family = "poisson", data = gala, x = TRUE)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.7897965  0.4437393   6.287 3.24e-10 ***
## Elevation    0.0009361  0.0002957   3.166  0.00154 **
## Nearest      0.0064693  0.0095646   0.676  0.49880
## Scruz       -0.0062665  0.0034307  -1.827  0.06776 .
## Adjacent    -0.0002858  0.0001620  -1.764  0.07781 .
## Size2        1.1276155  0.5218686   2.161  0.03072 *
## Size3        2.0586771  0.5155262   3.993 6.51e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 29.9553)
##
##     Null deviance: 3510.73  on 29  degrees of freedom
## Residual deviance:  594.18  on 23  degrees of freedom
## AIC: 769.01
##
## Number of Fisher Scoring iterations: 5
```

```
## fit the model with dispersion
m2 = glm(Species ~ Elevation + Nearest + Scruz + Adjacent + Size,
         family = "quasipoisson", data = gala, x = TRUE)
summary(m2)
```

```
##
## Call:
## glm(formula = Species ~ Elevation + Nearest + Scruz + Adjacent +
##     Size, family = "quasipoisson", data = gala, x = TRUE)
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   2.7897965   0.4437452    6.287 2.05e-06 ***
## Elevation      0.0009361   0.0002957    3.166 0.004314 **
## Nearest        0.0064693   0.0095648    0.676 0.505552
## Scruz         -0.0062665   0.0034308   -1.827 0.080777 .
## Adjacent      -0.0002858   0.0001621   -1.764 0.091094 .
## Size2          1.1276155   0.5218755    2.161 0.041376 *
## Size3          2.0586771   0.5155330    3.993 0.000572 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 29.95609)
##
##     Null deviance: 3510.73  on 29  degrees of freedom
## Residual deviance:  594.18  on 23  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

In this case the dispersion is quite large leading to an increase in standard errors of over a factor of 5

```
## dispersion and sqrt(dispersion)
c(dp, sqrt(dp))
```

```
## [1] 29.955296  5.473143
```

```
se = function(model) sqrt(diag(vcov(model)))
round(data.frame(coef.m1=coef(m1),
                 coef.m2=coef(m2),
                 se.m1=se(m1),
                 se.m2=se(m2),
                 ratio=se(m2)/se(m1)), 4)
```

```
##              coef.m1 coef.m2  se.m1  se.m2  ratio
## (Intercept)   2.7898  2.7898 0.0811 0.4437 5.4732
## Elevation     0.0009  0.0009 0.0001 0.0003 5.4732
## Nearest       0.0065  0.0065 0.0017 0.0096 5.4732
## Scruz        -0.0063 -0.0063 0.0006 0.0034 5.4732
## Adjacent     -0.0003 -0.0003 0.0000 0.0002 5.4732
## Size2         1.1276  1.1276 0.0954 0.5219 5.4732
## Size3         2.0587  2.0587 0.0942 0.5155 5.4732
```

The overdispersed Poisson model clearly offers improvements to the residuals vs fitted values plot

```
# dispersion
par(mfrow = c(1,2))
plot(predict(m1), (gala$Species - fitted(m1))^2,
     main = "Overdispersed Poisson model",
     xlab= expression(hat(theta)),
     ylab=expression((y-hat(mu))^2),
     pch = 19)
lines(sort(predict(m1)), dp*sort(fitted(m1)), lty="dashed")

# no dispersion
plot(predict(m1), (gala$Species - fitted(m1))^2,
     main = "Standard Poisson model",
     xlab= expression(hat(theta)),
     ylab=expression((y-hat(mu))^2),
```

```
    pch = 19)
lines(sort(predict(m1)), sort(fitted(m1)), lty="dashed")
```

Overdispersed Poisson model   ## Standard Poisson model
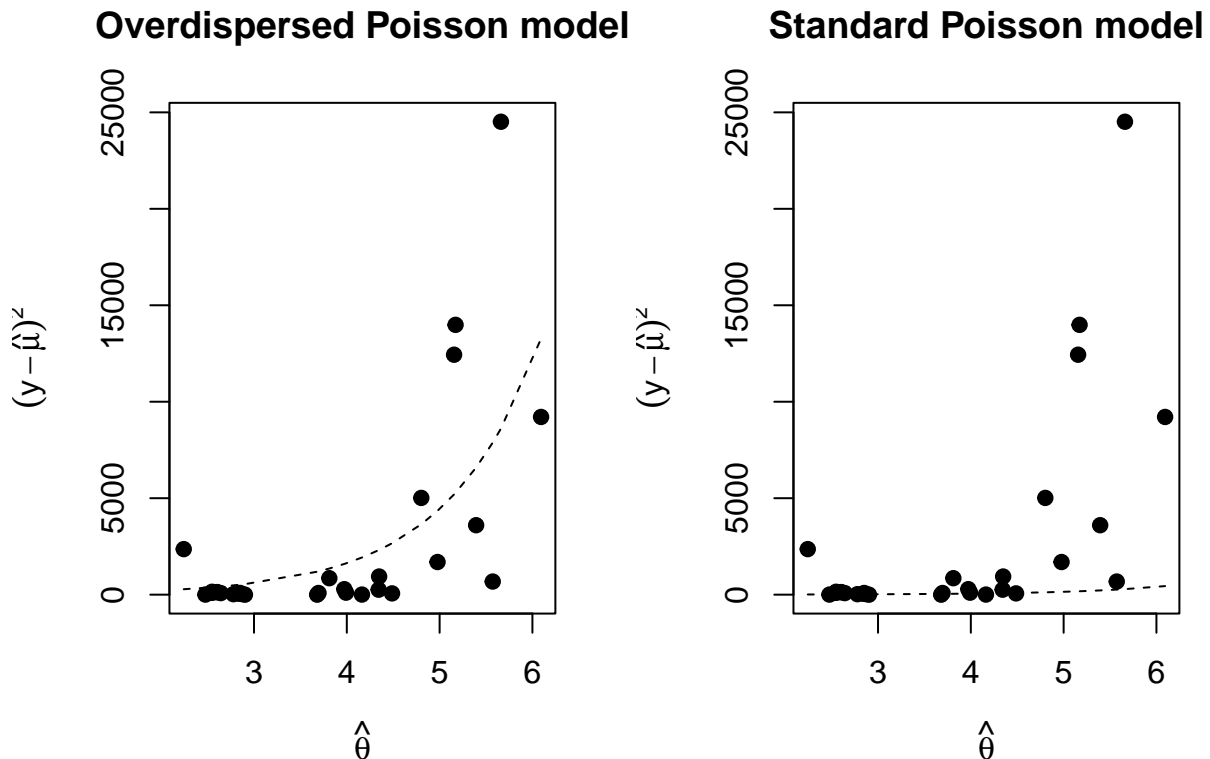


```
# log mean-value scale with dispersion (this distorts the success)
par(mfrow = c(1,2))
plot(log(fitted(m1)), log((gala$Species - fitted(m1))^2),
     main = "Overdispersed Poisson model",
     xlab= expression(hat(theta)),
     ylab=expression((y-hat(mu))^2),
     pch = 19)
lines(log(sort(fitted(m1))), log(dp*sort(fitted(m1))), lty="dashed")

# log mean-value scale with no dispersion
plot(log(fitted(m1)), log((gala$Species - fitted(m1))^2),
      main = "Overdispersed Poisson model",
     xlab= expression(hat(theta)),
     ylab=expression((y-hat(mu))^2),
     pch = 19)
lines(log(sort(fitted(m1))), log(sort(fitted(m1))), lty="dashed")
```
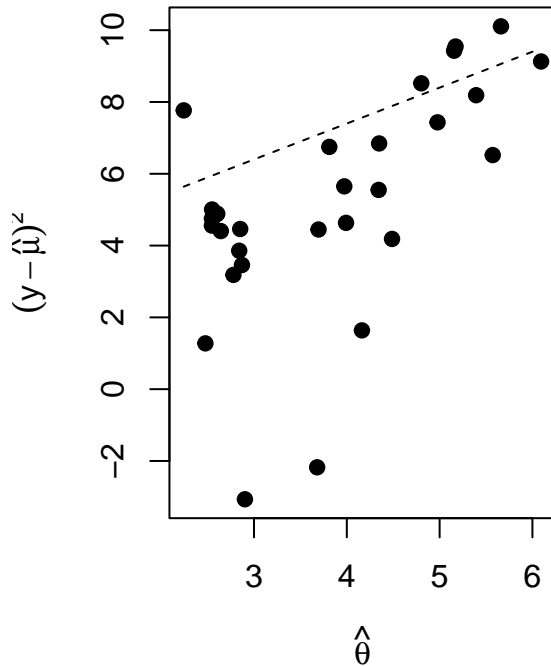
Overdispersed Poisson model **Overdispersed Poisson model**
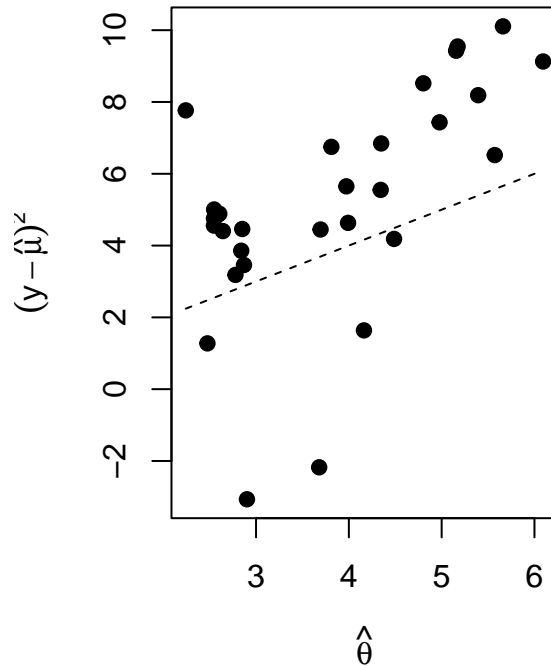
The `AER` package includes a function that allows for one to test whether or not dispersion is present. We can see that dispersion is present at most reasonable significance testing levels.

```
library(AER)
dispersiontest(m1, trafo=1)
```

```
##
##  Overdispersion test
##
## data:  m1
## z = 2.5007, p-value = 0.006198
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##    alpha
## 21.92009
```

Note that the `AER` package defines dispersion differently than the `glm` function.

Negative Binomial regression

Given a series of independent trials, each trial with probability of success p, let Z be the number of trials until the kth success. This is the basis for the negative binomial distribution. The mass function for the negative binomial distribution is:

$$\mathbb{P}(Z = z) = \binom{z-1}{k-1}p^k(1-p)^{z-k}, \qquad z = k, k+1, \dots$$

The negative binomial distribution can arise naturally in several ways. One can envision a system that can withstand k hits before failure. The probability of a hit in a given time period is p. The negative binomial also arises from the generalization of the Poisson where the rate parameter is gamma distributed. The negative binomial also comes up as a limiting distribution for urn schemes that can be used to model contagion.

We get a more convenient parameterization if we let $Y = Z - k$ and $p = (1 + \alpha)^{-1}$ so that:

$$\mathbb{P}(Y = y) = \binom{y + k - 1}{k - 1} \frac{\alpha^y}{(1 + \alpha)^{y+k}}, \qquad y = 0, 1, 2, \dots$$

then $\mathrm{E}(Y) = \mu = k\alpha$ and $\mathrm{Var}(Y) = k\alpha + k\alpha^2 = \mu + \mu^2/k$. The log-likelihood is then

$$\sum_{i=1}^{n} \left(y_i \log \left(\frac{\alpha}{1 + \alpha} \right) - k \log(1 + \alpha) + \sum_{j=0}^{y_i - 1} \log(j + k) - \log(y_i!) \right).$$

The most convenient way to link the mean response μ to a linear combination of the the predictors X in typical GLM fashion is through

$$\log \left(\frac{\mu}{\mu + k} \right) = \log \left(\frac{\alpha}{1 + \alpha} \right) = \theta = x^T \beta.$$

We can specify the change of parameters map $g : \theta \to \mu$ and the link function as $g^{-1} : \mu \to \theta$ as

$$g(\theta) = \frac{ke^\theta}{1 - e^\theta} = \mu, \qquad g^{-1}(\mu) = \log \left(\frac{\mu}{\mu + k} \right) = x^T \beta.$$

We can regard k as fixed and determined by the application or as an additional parameter to be estimated.

Example: Solder data

Consider this example. ATT ran an experiment varying five factors relevant to a wave-soldering procedure for mounting components on printed circuit boards. The response variable, skips, is a count of how many solder skips appeared in a visual inspection. The data comes from Comizzoli et al. (1990) (See the source material on the help page for the solder dataset in the faraway package). We start with a Poisson regression:

```
library(faraway)
modp = glm(skips ~ . , family=poisson, data=solder)

## saturated model fits better
pchisq(deviance(modp), df.residual(modp), lower = FALSE)
```

```
## [1] 5.281852e-65
```

```
## Faraway mentions rule of thumb that saturated model is
## likely better when deviance noticeably exceeds df
c(deviance(modp), df.residual(modp))
```

```
## [1] 1796.973  881.000
```

Perhaps including interaction terms will improve the fit:

```
modp2 = glm(skips ~ (Opening +Solder + Mask + PadType + Panel)^2,
            family=poisson, data=solder)

pchisq(deviance(modp2), df.residual(modp2), lower=FALSE)
```

```
## [1] 2.238499e-08
```

```
c(deviance(modp2), df.residual(modp2))
```

```
## [1] 1007.736  773.000
```

The fit is improved but not enough to conclude that the model fits. We could try adding more interactions but that would make interpretation increasingly difficult. A check for outliers reveals no problem. An

17

alternative model for counts is the negative binomial. The functions for fitting come from the `MASS` package. We can specify the link parameter k. Here we choose $k = 1$ to demonstrate the method, although there is no substantive motivation from this application to use this value. Note that the $k = 1$ case corresponds to an assumption of a geometric distribution for the response.

```
library(MASS)
modn = glm(skips ~ ., negative.binomial(1), solder)
modn
```

```
##
## Call:  glm(formula = skips ~ ., family = negative.binomial(1), data = solder)
##
## Coefficients:
## (Intercept)      OpeningM      OpeningS    SolderThin        MaskA3        MaskA6
##    -1.57251       0.50852       2.00093       1.04754       0.65065       2.52776
##       MaskB3        MaskB6     PadTypeD6     PadTypeD7     PadTypeL4     PadTypeL6
##     1.26631       2.07062      -0.45434       0.01979       0.46751      -0.46812
##    PadTypeL7     PadTypeL8     PadTypeL9      PadTypeW4      PadTypeW9        Panel2
##    -0.28999      -0.08057      -0.51864      -0.13917      -1.48133       0.29536
##       Panel3
##     0.34262
##
## Degrees of Freedom: 899 Total (i.e. Null);  881 Residual
## Null Deviance:        1743
## Residual Deviance: 556.7     AIC: 3884
```

```
## LRT test
pchisq(deviance(modn), df.residual(modn), lower=FALSE)
```

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

We could experiment with different values of k, but there is a more direct way of achieving this by allowing the parameter k to vary and be estimated using maximum likelihood in:

```
modn = glm.nb(skips ~ ., solder)
summary(modn)
```

```
##
## Call:
## glm.nb(formula = skips ~ ., data = solder, init.theta = 4.52811339,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.29859    0.13202  -9.837  < 2e-16 ***
## OpeningM     0.50353    0.07932   6.348 2.18e-10 ***
## OpeningS     1.91104    0.07111  26.876  < 2e-16 ***
## SolderThin   0.93513    0.05323  17.567  < 2e-16 ***
## MaskA3       0.58383    0.09592   6.087 1.15e-09 ***
## MaskA6       2.26096    0.10101  22.384  < 2e-16 ***
## MaskB3       1.20651    0.09572  12.605  < 2e-16 ***
## MaskB6       1.98172    0.09158  21.638  < 2e-16 ***
## PadTypeD6   -0.46189    0.11145  -4.144 3.41e-05 ***
## PadTypeD7   -0.03182    0.10584  -0.301 0.763655
## PadTypeL4    0.38119    0.10177   3.745 0.000180 ***
## PadTypeL6   -0.57860    0.11327  -5.108 3.25e-07 ***
## PadTypeL7   -0.36569    0.11006  -3.323 0.000891 ***
```

```
## PadTypeL8    -0.15882     0.10734  -1.480 0.138953
## PadTypeL9    -0.56554     0.11306  -5.002 5.67e-07 ***
## PadTypeW4    -0.19851     0.10783  -1.841 0.065630 .
## PadTypeW9    -1.56332     0.13538 -11.547  < 2e-16 ***
## Panel2        0.29574     0.06322   4.678 2.90e-06 ***
## Panel3        0.33380     0.06298   5.300 1.16e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(4.5281) family taken to be 1)
##
##     Null deviance: 4097.6  on 899  degrees of freedom
## Residual deviance: 1012.1  on 881  degrees of freedom
## AIC: 3679.5
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  4.528
##           Std. Err.:  0.518
##
##  2 x log-likelihood:  -3639.514
```

We see that $\hat{k} = 4.528$ with a standard error of 0.518. We can compare negative binomial models using the usual inferential techniques. For instance, we see that the overall fit is much improved.

```
## LRT test
pchisq(deviance(modn), df.residual(modn), lower=FALSE)
```

```
## [1] 0.001367546
```

Zero Inflated Count Models

Sometimes we see count response data where the number of zeroes appearing is significantly greater than the Poisson or negative binomial models would predict. This commonly arises in life history analyses of plants and animals where many subjects die before they reproduce, arrest and bookings data where many people are either not arrested or receive zero day sentences, and insurance claims data. Modifying the Poisson by adding a dispersion parameter does not adequately model this divergence from the standard count distributions.

We consider a sample of 915 biochemistry graduate students. The response variable is the number of articles produced during the last three years of the PhD. We are interested in how this is influenced by the gender, marital status, number of children, prestige of the department and productivity of the advisor of the student. The dataset may be found in the `pscl` package. We start by fitting a Poisson regression model:

```
library(pscl)
modp = glm(art ~ ., data=bioChemists, family=poisson)
sumary(modp)
```

```
##               Estimate Std. Error z value  Pr(>|z|)
## (Intercept)  0.3046168  0.1029814  2.9580  0.003097
## femWomen    -0.2245942  0.0546135 -4.1124 3.915e-05
## marMarried   0.1552434  0.0613744  2.5294  0.011424
## kid5        -0.1848827  0.0401269 -4.6075 4.076e-06
## phd          0.0128226  0.0263970  0.4858  0.627139
## ment         0.0255427  0.0020061 12.7327 < 2.2e-16
##
```

```
## n = 915 p = 6
## Deviance = 1634.37098 Null Deviance = 1817.40530 (Difference = 183.03432)
```

We can see that deviance is significantly larger than the degrees of freedom (a rule of thumb indicated poor fit). Some experimentation reveals that this cannot be solved by using a richer linear predictor or by eliminating some outliers [Faraway, 2016]. We might consider a dispersed Poisson model or negative binomial but some thought suggests that there are good reasons why a disproportionate number of students might produce no articles at all.

We count and predict how many students produce between zero and seven articles. Very few students produce more than seven articles so we ignore these. The `predprob` function produces the predicted probabilities for each case. By summing these, we get the expected number for each article count.

```r
ocount = table(bioChemists$art)[1:8]
pcount = colSums(predprob(modp)[,1:8])
plot(pcount, ocount, type="n", xlab="Predicted", ylab="Observed",
     ylim = c(0, 300), axes = FALSE)
axis(side = 1)
axis(side = 2)
text(pcount,ocount, 0:7)
```
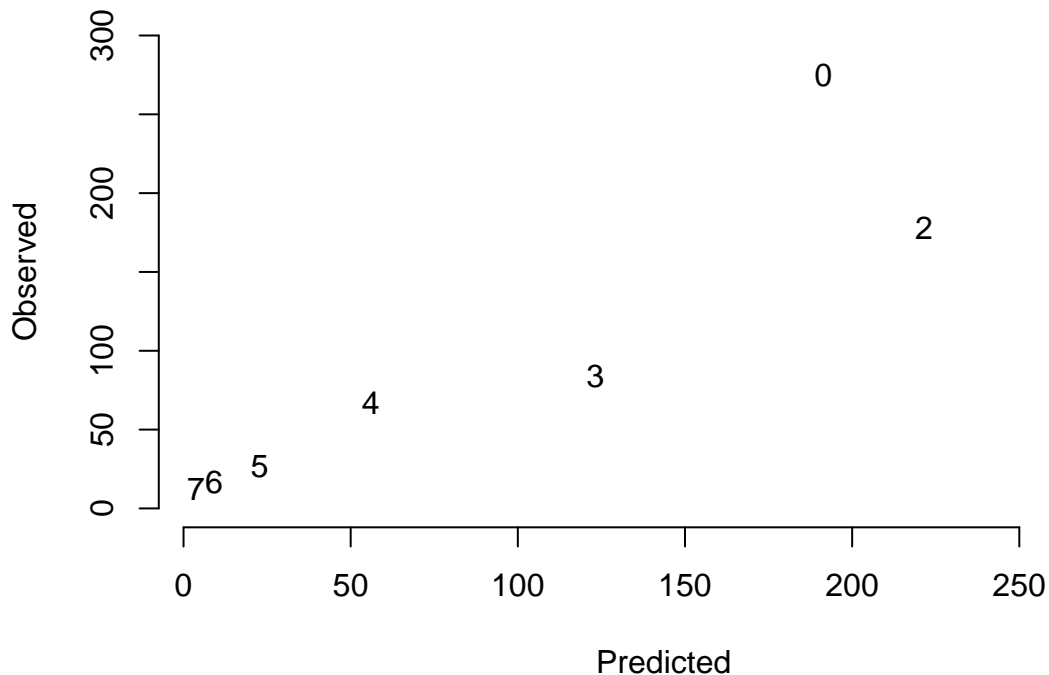

We see that there are many more students with zero articles than would be predicted by the Poisson model. In contrast, the relationship between observed and predicted is linear for the students who produce at least one article.

We now consider a zero-inflated Poisson model. First, some motivation. Suppose we ask the general public how many games of chess they have played in the last month. Some people will say zero because they do not play chess but some zero responses will be from chess players who have not managed a game in the last month. Circumstances such as these require a *mixture* model. A general specification of this model takes the form:

$$\mathbb{P}(Y = 0) = \phi + (1 - \phi)f(0)$$
$$\mathbb{P}(Y = j) = (1 - \phi)f(j), \qquad j > 0.$$

The parameter ϕ represents the proportion of subjects who will always respond zero (the non-chess players in the motivating example). One can model the proportion ϕ using a binary response model. The distribution

f models the counts of those individuals that can have a positive response. Note that it is possible that some of these individuals will have zero response which combine with the always-zero individuals. We can use a Poisson model for f in which case this is called zero-inflated Poisson model.

```
modz = zeroinfl(art ~ ., data=bioChemists)
summary(modz)
```

```
##
## Call:
## zeroinfl(formula = art ~ ., data = bioChemists)
##
## Pearson residuals:
##     Min     1Q Median     3Q    Max
## -2.3253 -0.8652 -0.2826  0.5404  7.2976
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.640838   0.121307   5.283 1.27e-07 ***
## femWomen    -0.209145   0.063405  -3.299 0.000972 ***
## marMarried   0.103751   0.071111   1.459 0.144565
## kid5        -0.143320   0.047429  -3.022 0.002513 **
## phd         -0.006166   0.031008  -0.199 0.842378
## ment         0.018098   0.002294   7.888 3.07e-15 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.577059   0.509386  -1.133  0.25728
## femWomen     0.109746   0.280082   0.392  0.69518
## marMarried  -0.354014   0.317611  -1.115  0.26502
## kid5         0.217097   0.196482   1.105  0.26919
## phd          0.001274   0.145263   0.009  0.99300
## ment        -0.134114   0.045243  -2.964  0.00303 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 21
## Log-likelihood: -1605 on 12 Df
```

We notice that the `ment` variable which counts the number of articles produced by the mentor is the most significant predictor in both the Poisson and binary regressions. Notice that the signs of regression coefficient estimates are reversed across the two models. This is because the zero-inflated approach models the probability of a zero count (not the probability of a successful 1 count). Hence there is no contradiction in inferences.

We can use the standard likelihood testing theory to compare nested models. For example, suppose we consider a simplified version of the zero-inflated Poisson model where we now have different predictors for the two components of the model. The count part of the model is specified before the | and the binary response model after.

```
# smaller model
modz2 = zeroinfl(art ~ fem+kid5+ment | ment, data=bioChemists)

# summary table for smaller model
summary(modz2)
```

```
##
## Call:
```

```
## zeroinfl(formula = art ~ fem + kid5 + ment | ment, data = bioChemists)
##
## Pearson residuals:
##     Min     1Q Median     3Q    Max
## -2.2802 -0.8807 -0.2718  0.5131  7.4788
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.694517   0.053025  13.098  < 2e-16 ***
## femWomen    -0.233857   0.058400  -4.004 6.22e-05 ***
## kid5        -0.126516   0.039668  -3.189  0.00143 **
## ment         0.018004   0.002224   8.096 5.67e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.68488    0.20529  -3.336 0.000849 ***
## ment        -0.12680    0.03981  -3.185 0.001448 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 13
## Log-likelihood: -1608 on 6 Df
# test of nested models
pchisq(2*(modz$loglik-modz2$loglik), 6, lower = FALSE)
```

```
## [1] 0.4041153
```

Given the large p-value of 0.4, we conclude that our simplification of the model is justifiable. For interpretation, the exponentiated coefficients are more useful:

```
exp(coef(modz2))
```

```
## count_(Intercept)     count_femWomen        count_kid5        count_ment
##         2.0027412          0.7914748         0.8811604         1.0181669
##  zero_(Intercept)          zero_ment
##         0.5041522          0.8809082
```

We can also use the model to make predictions. Consider a single male with no children whose mentor produced six articles:

```
newman = data.frame(fem="Men",mar="Single",kid5=0,ment=6)
predict(modz2, newdata=newman, type="prob")[1:6]
```

```
## [1] 0.27758792 0.19394034 0.21635996 0.16091421 0.08975799 0.04005363
```

We see that most likely outcome for this student is that no articles will be produced with a probability of 0.278. We can query the probability of no production from the zero part of the model:

```
predict(modz2, newdata=newman, type="zero")
```

```
##        1
## 0.190666
```

So the additional probability to make this up to 0.278 comes from the Poisson count part of the model. This difference might be attributed to students who had the potential to write an article.

We note that the zero-inflated Poisson distribution also arises as a special case of an aster model used in life history analyses for populations of plants or animals. The aster model can be thought of as a generalized

generalized linear regression model and it models individuals through stages of their lifecycle. One can think of a simple lifecycle:

$$1 \to Y_1 \to Y_2,$$

where Y_1 encodes subjects either reaching or not reaching a reproductive stage (modeled as a binary random variable) and then, conditional on reproduction, Y_2 encodes how many offspring produced by the subjects in the reproductive stage (modeled as a Poisson or 0-truncated Poisson random variable).

Acknowledgments

References

Alan Agresti. *Categorical Data Analysis*. John Wiley & Sons, 2013.

Julian J Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press, 2016.