STAT 528 - Advanced Regression Analysis II

Count response regression (part I)

Daniel J. Eck
Department of Statistics
University of Illinois

Last time

- basic diagnostics for binary response models
- probit regression and threshold modeling
- basic causal inference

Learning Objectives Today

- Poisson regression
- Residual diagnostics
- Data analysis

Background (again)

We suppose that we have a sample of data (y_i, x_i), $i = 1, \ldots, n$ where

- y_i is a scalar response variable
- x_i is a vector of predictors.

Recall from the exponential family notes that the log likelihood of the exponential family is of the form

$$l(\theta) = \langle y, \theta \rangle - c(\theta), \tag{1}$$

where

- $y \in \mathbb{R}^n$ is a vector statistic having components
- $\theta \in \mathbb{R}^n$ is the canonical parameter vector.

In those notes θ is unconstrained and the likelihood (1) corresponds to a saturated regression model, one parameter for every observation.

A canonical linear submodel of an exponential family is a submodel having parameterization

$$\theta = M\beta,$$

and log likelihood

$$l(\beta) = \langle M'y, \beta \rangle - c(M\beta). \tag{2}$$

In an exponential family GLM, the saturated model canonical parameter vector θ is "linked' ' to the saturated model mean value parameter vector through the change-of-parameter mappings $g(\theta)$.

We can write

$$\mu = \mathrm{E}_\theta(Y) = g(M\beta)$$

which implies that we can write

$$g^{-1}\left(\mathrm{E}_\theta(Y)\right) = M\beta.$$

Poisson regression model

The Poisson regression model [and its variants] is one of the more widely used and studied exponential family GLMs in practice.

The Poisson regression model is used for analyzing a count response variable, $y_i \in \{0, 1, 2, \ldots\}$.

The Poisson regression model allows for users to model expected counts as a function of covariates.

preliminaries

Recall the Poisson distributions with mass function

$$P(Y = y) = \frac{\mu^y e^{-\mu}}{y!}, \qquad (y = 0, 1, 2, \ldots),$$

where $E(Y) = \mu$ and $\mathrm{Var}(Y) = \mu$.

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

$$\mu(x) = \mathrm{E}(Y|X = x) = \exp\left(x'\beta\right). \tag{3}$$

Equivalently,

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

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.

Let's consider the log likelihood of a sample of independent Poisson random variables

$$\sum_{i=1}^{n} y_i \log(\mu_i) - \mu_i = \sum_{i=1}^{n} y_i \theta_i - \exp(\theta_i)$$

where

$$\theta_i = \log(\mu_i) = g^{-1}(\mu_i) \qquad \text{and} \qquad \mu_i = \exp(\theta_i) = g(\theta_i).$$

We see that the Poisson regression model with log link is a canonical linear submodel of an exponential family when we write

$$\theta_i = x_i' \beta.$$

Example: Gala data

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
- 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.

Galapagos variables

- **Species**: the number of plant species found on the island
- **Area**: the area of the island (km^2)
- **Elevation**: the highest elevation of the island (m)
- **Nearest**: the distance from the nearest island (km)
- **Scruz**: the distance from Santa Cruz island (km)
- **Adjacent**: the area of the adjacent island (square km)

M. P. Johnson and P. H. Raven (1973) "Species number and endemism: The Galapagos Archipelago revisited" Science, 179, 893-895

We first load in necessary software.

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

We create a discrete size variable based on the Area variable for demonstration purposes, and then fit the Poisson regression model

```
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)
```

As in logistic regression, we are going to unpack the glm call.

The specific log likelihood for the Poisson regression model is

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

where

- x_i's are the rows of the design matrix M
- y_is are the components of the response vector y (the Species variable corresponding to the number of species on each of the islands)
- β is the submodel canonical parameter vector

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)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -10.3723   -3.5214   -0.9947    1.7193   10.6627
##
## 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
```

Keep in mind that interpretations of β are on the log scale.

A particular component of $\hat{\beta}$ gives the estimated change in logs of expected counts when one makes a unit change to a particular component of the covariate vector with all other components held fixed.

Similar story for logistic regression. A particular component of $\hat{\beta}$ gives the estimated change in log odds when one makes a unit change to a particular component of the covariate vector with all other components held fixed.

This is good to keep in mind and it *may* be instructive for how you answer homework questions. But keep in mind that:

- β is given by M and not span(M)
- modeling of expectations is the motivation for these models

Demonstration of this point

```
m2 <-  glm(Species ~ -1 + Elevation + Nearest + Scruz + Adjacent + Size,
          family = "poisson", data = gala, x = TRUE)
summary(m2)
```

```
##
## Call:
## glm(formula = Species ~ -1 + Elevation + Nearest + Scruz + Adjacent +
##     Size, family = "poisson", data = gala, x = TRUE)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -10.3723   -3.5214   -0.9947    1.7193   10.6627
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## 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 ***
## Size1      2.790e+00  8.108e-02  34.410  < 2e-16 ***
## Size2      3.917e+00  5.761e-02  68.001  < 2e-16 ***
## Size3      4.848e+00  5.987e-02  80.990  < 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: 21190.47  on 30  degrees of freedom
## Residual deviance:   594.18  on 23  degrees of freedom
## AIC: 769.01
##
## Number of Fisher Scoring iterations: 5
```

We see that

```
logLik(m1) - logLik(m2)
```

```
## 'log Lik.' 5.684342e-14 (df=7)
```

and

```
theta1 <- m1$x %*% coef(m1)
theta2 <- m2$x %*% coef(m2)
all.equal(theta1, theta2)
```

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

However, $\hat{\beta}_1 \neq \hat{\beta}_2$

```
coef(m1)
```

```
##   (Intercept)     Elevation       Nearest         Scruz      Adjacent
##  2.7897964708  0.0009360990  0.0064693041 -0.0062664946 -0.0002857805
##         Size2         Size3
##  1.1276155398  2.0586771298
```

```
coef(m2)
```

```
##     Elevation       Nearest         Scruz      Adjacent         Size1
##  0.0009360990  0.0064693041 -0.0062664946 -0.0002857805  2.7897964708
##         Size2         Size3
##  3.9174120106  4.8484736007
```

Other parameterizations

Recall:

$$\begin{tikzcd}
\theta \arrow[r, shift left] \arrow[d, shift left, "\mu = \nabla c(\theta)"] & \beta \arrow[l, shift left, "\theta = M\beta"] \arrow[d, shift left, "\tau = \nabla_\beta c(M\beta)"] \\
\mu \arrow[u, shift left] \arrow[r, shift left, "\tau = M^T\mu"] & \tau \arrow[u, shift left] \arrow[l, shift left]
\end{tikzcd}$$

We start with

$$\langle M'y, \beta \rangle - c_\beta(\beta),$$

and then obtain $\hat{\beta}$ by maximizing the above. From here:

- $\hat{\theta} = M\hat{\beta}$
- $\hat{\mu} = \nabla_\theta c(\theta^*)|_{\theta*=\hat{\theta}}$
- $\hat{\tau} = M'\hat{u}$

For example we can compute $\hat{\theta}$ using the predict function or by hand.

```
theta <- m1$x %*% coef(m1)
head(cbind(predict(m1, type = "link"), theta), 3)
```

```
##                [,1]     [,2]
## Baltra    5.171960 5.171960
## Bartolome 3.694959 3.694959
## Caldwell  2.546560 2.546560
```

Inference

The standard error column of the summary table are estimates of the square root of the variances of the estimated submodel canonical statistic vector $\hat{\beta}$.

Recall from the asymptotic theory of MLE 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
```

We can make inferences about β_j using the Wald statistic corresponding to the hypothesis test

$$H_o : \beta_j = 0, \qquad H_a : \beta_j \neq 0,$$

which is given by

$$\frac{\hat{\beta}_j}{\text{se}(\hat{\beta}_j)} \sim N(0, 1),$$

where this distributional relationship holds under the null hypothesis $\beta_j = 0$. Similarly, we can form a confidence interval

$$\hat{\beta}_j \pm z_{\alpha/2}\text{se}(\hat{\beta}_j)$$

where $0 < \alpha < 1$ is some error threshold.

Deviance and likelihood ratio testing

Recall our $l(\mu; y)$ notation.

The deviance is defined by

$$-2\left[l(\hat{\mu}; y) - l(y; y)\right].$$

This is the likelihood-ratio for testing the null hypothesis that the model against the general alternative (ie, the saturated model). The deviance has reference distribution

$$-2\left[l(\hat{\mu}; y) - l(y; y)\right] \approx \chi^2_{\mathrm{df}}$$

where $\mathrm{df} = n - p$, n is the sample size, and p is the number of model parameters.

We can also test nested models using deviance based testing

$$-2\left[l(\hat{\mu}_1; y) - l(\hat{\mu}_2; y)\right] \approx \chi^2_{\mathrm{df}}$$

where

- $\hat{\mu_1}$ corresponds to a smaller model than that which led to estimation of $\hat{\mu}_2$
- $\mathrm{df} = p_1 - p_2$

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.

```
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
```

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

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

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 hypothesis on the previous slide using the Delta method

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

where

$$h(\beta) = \exp(\beta_l) - \exp(\beta_m) = \mu_l - \mu_m.$$

We obtain the estimate $\hat{\mu}_l - \hat{\mu}_m$ below

```r
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

```r
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
```

Note

Keep in mind that there are three possible tests that we could have made. We can adjust for this using a Bonferroni correction

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

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

Residual 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:

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

where - $\mathrm{Var} = \nabla^2 c(\theta)$ 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. Pearson residuals can be skewed for nonnormal responses.

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

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

and

$$r_{D,i} = \text{sign}(y_i - \hat{\mu}_i)\sqrt{d_i}.$$

In Poisson regression the deviance residuals are

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

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 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).

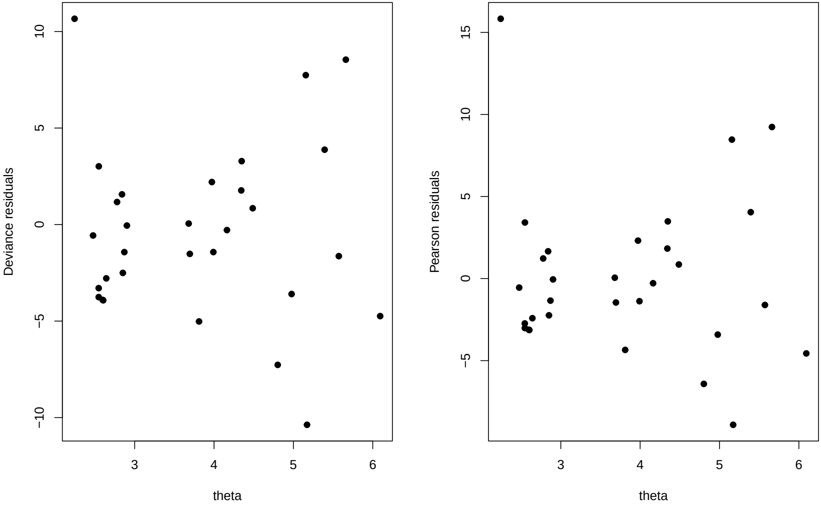
We are looking for features in these residuals vs. fitted values plots:

- 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.

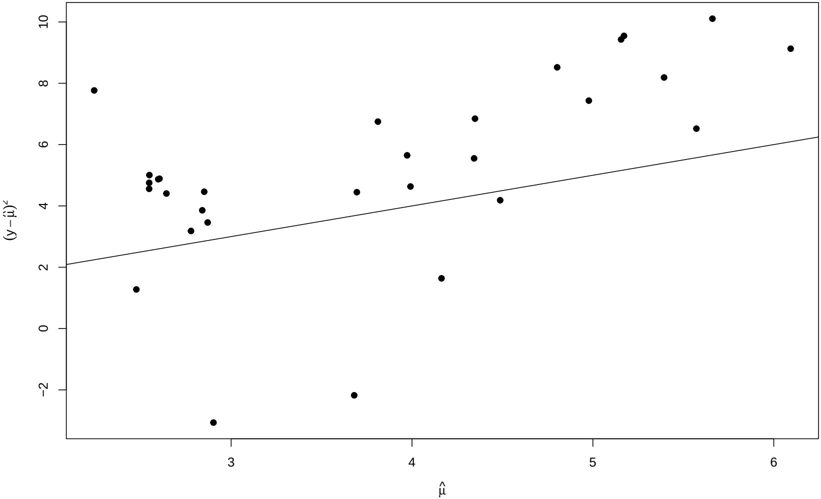
The plots below show the residuals as a function of $\hat{\theta}_i = x_i^T \hat{\beta}$.

```
theta <- as.numeric(m1$x %*% coef(m1))
par(mfrow = c(1,2))
plot(theta, residuals(m1), xlab = "theta", ylab = "Deviance residuals", pch = 19)
plot(theta, residuals(m1, "pearson"), pch = 19,
     xlab = "theta", ylab = "Pearson residuals")
```



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.

```
plot(log(fitted(m1)),log((gala$Species-fitted(m1))^2),
     xlab= expression(hat(mu)),ylab=expression((y-hat(mu))^2), pch = 19)
abline(0,1)
```



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.

This problem is called overdispersion.

(further reading)

We note that a deviance based test tells us that our submodel does not fit the data better than the full saturated model

```
## compare with saturated model
m1$deviance
```

```
## [1] 594.1753
```

```
m1$df.residual
```

```
## [1] 23
```

```
pchisq(m1$deviance, df = m1$df.residual, lower = FALSE)
```

```
## [1] 7.617409e-111
```

The model misses bad in some spots

```
foo <- data.frame(names = rownames(gala),
                  obs = gala$Species,
                  pred = as.numeric(predict(m1, type = "response")),
                  resid = residuals(m1))
head(cbind(foo %>% arrange(desc(obs)) %>% pull(names),
           foo %>% arrange(desc(pred)) %>% pull(names)), 10)
```

```
##       [,1]          [,2]
##  [1,] "SantaCruz"    "Isabela"
##  [2,] "Isabela"      "SantaCruz"
##  [3,] "SantaMaria"   "SanSalvador"
##  [4,] "SanCristobal" "SanCristobal"
##  [5,] "SanSalvador"  "Baltra"
##  [6,] "Pinzon"       "SantaMaria"
##  [7,] "Pinta"        "Pinta"
##  [8,] "Espanola"     "Marchena"
##  [9,] "Fernandina"   "Espanola"
## [10,] "Rabida"       "Pinzon"
```

```
foo %>% arrange(desc(abs(resid))) %>% head(10)
```

```
##                     names obs       pred      resid
## Gardner1         Gardner1  58   9.418902  10.662732
## Baltra             Baltra  58 176.259919 -10.372266
## SantaCruz       SantaCruz 444 287.434827   8.543845
## SantaMaria     SantaMaria 285 173.479088   7.741091
## Marchena         Marchena  51 121.814494  -7.267755
## Tortuga           Tortuga  16  45.214750  -5.018644
## Isabela           Isabela 347 442.973448  -4.741568
## Coamano           Coamano   2  13.517259  -3.923167
## Enderby           Enderby   2  13.421436  -3.902313
## SanCristobal SanCristobal 280 219.998697   3.879621
```

Can play around with the model to improve performance

```
m3 <- glm(Species ~ Elevation + I(Elevation^2) + Nearest + Scruz +
            I(Scruz^2) + Adjacent +
            Area + I(Area^2), family = "poisson", data = gala, x = TRUE)

foo <- data.frame(names = rownames(gala),
                  Species = gala$Species,
                  pred = as.numeric(predict(m3, type = "response")),
                  resid = residuals(m3))
head(cbind(foo %>% arrange(desc(Species)) %>% pull(names),
           foo %>% arrange(desc(pred)) %>% pull(names)), 10)
```

```
##       [,1]          [,2]
##  [1,] "SantaCruz"    "SantaCruz"
##  [2,] "Isabela"      "Isabela"
##  [3,] "SantaMaria"   "SanSalvador"
##  [4,] "SanCristobal" "SanCristobal"
##  [5,] "SanSalvador"  "SantaMaria"
##  [6,] "Pinzon"       "Pinta"
##  [7,] "Pinta"        "Rabida"
##  [8,] "Espanola"     "Marchena"
##  [9,] "Fernandina"   "Fernandina"
## [10,] "Rabida"       "Pinzon"
```

This model offers improvements in residual magnitude and we see that problems are occurring for small species counts.

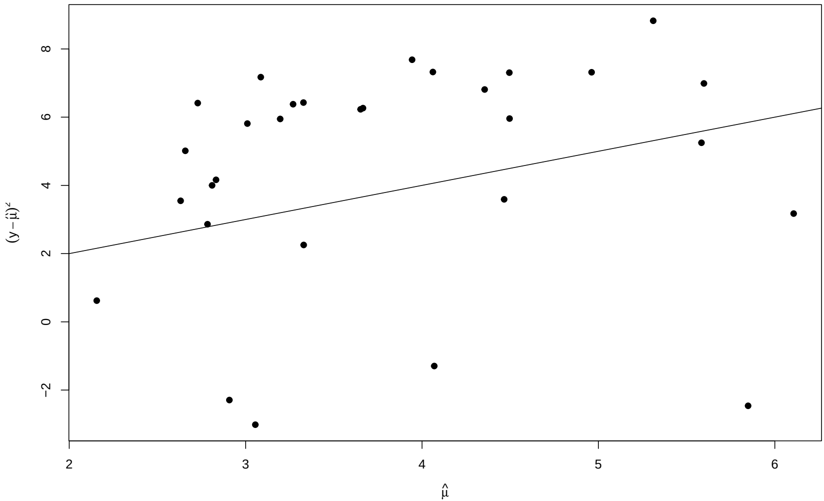
```
foo %>% arrange(desc(abs(resid))) %>% head(11)
```

```
##                 names Species      pred     resid
## Gardner2     Gardner2       5  51.60949 -8.359204
## Gardner1     Gardner1      58  21.88658  6.389305
## Enderby       Enderby       2  26.28609 -6.186163
## Caldwell     Caldwell       3  27.87680 -6.031459
## SantaMaria SantaMaria     285 202.49381  5.459134
## Genovesa     Genovesa      40  15.30862  5.239687
## Coamano       Coamano       2  20.28438 -5.225127
## Espanola     Espanola      97  58.06115  4.656808
## Marchena     Marchena      51  89.57405 -4.438200
## Tortuga       Tortuga      16  38.53710 -4.116449
## Onslow         Onslow       2  14.26763 -4.083611
```

```
cbind(foo$pred, gala[, -8])[which(abs(foo$resid) > 4), ]
```

```
##             foo$pred Species Endemics   Area Elevation Nearest Scruz Adjacent
## Caldwell    27.87680       3        3   0.21       114     2.8  58.7     0.78
## Coamano     20.28438       2        1   0.05        77     1.9   1.9   903.82
## Enderby     26.28609       2        2   0.18       112     2.6  50.2     0.10
## Espanola    58.06115      97       26  58.27       198     1.1  88.3     0.57
## Gardner1    21.88658      58       17   0.57        49     1.1  93.1    58.27
## Gardner2    51.60949       5        4   0.78       227     4.6  62.2     0.21
## Genovesa    15.30862      40       19  17.35        76    47.4  92.2   129.49
## Marchena    89.57405      51       23 129.49       343    29.1  85.9    59.56
## Onslow      14.26763       2        2   0.01        25     3.3  45.9     0.10
## SantaMaria 202.49381     285       73 170.92       640     2.6  49.2     0.10
## Tortuga     38.53710      16        8   1.24       186     6.8  50.9    17.95
```

```r
plot(log(fitted(m3)),log((gala$Species-fitted(m3))^2),
     xlab= expression(hat(mu)),ylab=expression((y-hat(mu))^2), pch = 19)
abline(0,1)
```



```r
theta <- as.numeric(m3$x %*% coef(m3))
par(mfrow = c(1,2))
plot(theta, residuals(m3), xlab = "theta", ylab = "Deviance residuals", pch = 19)
plot(theta, residuals(m3, "pearson"), pch = 19,
     xlab = "theta", ylab = "Pearson residuals")
```