Day 4: GLMs for Count Data

Recap

We split regression models into three main parts:

- Random component: outcome variable Y takes some probability distribution, with $E[Y] = \mu$
- Systematic component: Explanatory variables $X_1, \ldots, X_p$ affect the response through some function $\eta = \phi(X_1, \ldots, X_p)$
- Link Function: The random component and systematic component are related through a link function, $g(\mu) = \eta$

Recap

GLMs loosen restrictions on the random component and link function. The distribution used in the random component must be a member of the exponential family:

$$f(y_i|\theta_i, \phi) = \mathsf{exp}\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right\}$$

where

- θ_i is the natural parameter
- ϕ is a dispersion parameter

Recap

GLMs loosen restrictions on the random component and link function. The distribution used in the random component must be a member of the exponential family:

$$f(y_i|\theta_i, \phi) = \text{exp}\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right\}$$

- One common choice of link function sets $g(\mu) = \theta$
- This is the canonical link function

Count Data

- Assume response variable Y_i takes positive integer values with no upper limit.
- One example of a case like this is the number of reported car accidents in a given location and time period.
- We can use the Poisson distribution to model this outcome.
- Random component: $Y_i \sim \text{Poisson}(\mu_i)$, so $E[Y_i] = \mu_i$.

Finding a Link Function

We transform the Poisson distribution to exponential family form:

$$f(y_i|\theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right\}$$

$$\begin{aligned} f(y_i|\mu_i) &= \frac{e^{-\mu}\mu^{y_i}}{y_i!} \\ &\propto e^{-\mu}\mu^{y_i} \\ &= \exp\left(y_i \log\mu - \mu\right) \end{aligned}$$

So we have $\theta = \log\mu$ and $b(\theta) = e^\theta = \mu$.

The canonical link function is $\eta = g(\mu) = \log\mu$.

Log-Linear Model

The Poisson regression model has the random component $Y_i \sim \text{Poisson}(\mu_i)$, where $E[Y_i] = \mu_i$. We assume the link function and systematic component:

$$\log \mu_i = \eta_i = x_i \beta.$$

Coefficient Interpretation

If the model is defined as $\log \mu = \alpha + \beta X$, we compare situations where $X = 0$ and $X = 1$ to see how we can interpret β.

We have

$$\begin{aligned} \beta &= (\alpha + \beta) - (\alpha) \\ &= \log \mu_1 - \log \mu_0 \\ &= \log \frac{\mu_1}{\mu_0} \end{aligned}$$

Exponentiating the coefficient yields a multiplicative interpretation:

$$e^{\beta} = \frac{\mu_1}{\mu_0}.$$

Dispersion Assumption

One property of the Poisson distribution is that $E[Y] = Var[Y]$. We can express this as

$$Var[Y_i] = \sigma^2 E[Y_i],$$

where σ^2 is a dispersion parameter.

- $\sigma^2 < 1$ represents under-dispersion.
- $\sigma^2 > 1$ represents over-dispersion.

Over-dispersion is the more common problem in practice.

Variance-Stabilizing Transformations

Variance-stabilizing transformations are one way of handling over-dispersion.

Goal: Look for some transformation $f(Y)$ that makes $Var[f(Y)]$ constant in terms of μ.

We can take the Taylor expansion:

$$f(Y) \approx f(\mu) + f'(\mu)(Y - \mu),$$

which gives

$$\begin{aligned} Var(f(Y)) &\approx Var[f(\mu) + f'(\mu)(Y - \mu)] \\ &= Var[f'(\mu)(Y - \mu)] \\ &= f'(\mu)^2 Var[Y] \\ &= f'(\mu)^2 \mu \end{aligned}$$

Variance-Stabilizing Transformations

For Poisson-distributed data, $f(Y) = \sqrt{Y}$ is a common variance-stabilizing transformation.

$$\begin{aligned} Var(\sqrt{\mu}) &\approx \left(\frac{d}{d\mu}\sqrt{\mu}\right)^2 \mu \\ &= \left(\frac{1}{2}\mu^{-\frac{1}{2}}\right)^2 \mu \\ &= \frac{1}{4}\mu^{-1}\mu \\ &= \frac{1}{4} \end{aligned}$$

Residual Plots

The link function makes it difficult to use the traditional residual plots. Instead, we choose from two alternatives:

- Pearson Residuals: $p_i = \frac{r_i}{\sqrt{\hat{\phi}\exp(X_i\beta)}}$, with

$$\hat{\phi} = \frac{1}{n-p}\sum_{i=1}^{n}\frac{(y_i - \exp\left(X_i\hat{\beta}\right))^2}{\exp\left(X_i\hat{\beta}\right)}$$

- Deviance Residauls:

$$d_i = \mathsf{sign}(y_i - \exp\left(X_i\hat{\beta}\right))\sqrt{2\left[y_i\log\left(\frac{y_i}{\exp\left(X_i\hat{\beta}\right)}\right) - (y_i - \exp\left(X_i\hat{\beta}\right)\right]}$$

Use the `type` argument in the `resid()` function to specify these residuals.

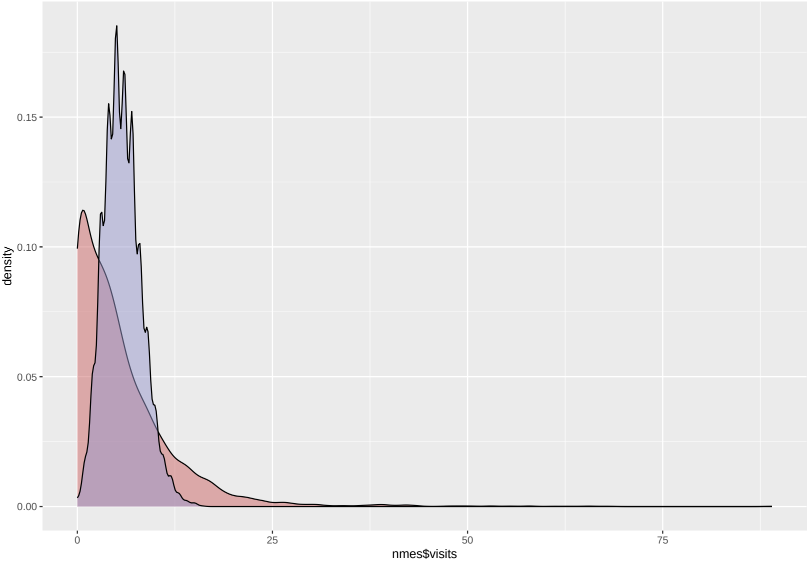
Hurdle Models

Consider a case where the outcome variable is made of count data with many more observations equal to 0.

Hurdle models can combine GLMs for binary data and for count data.

- Step 1: Use logistic regression or probit regression to predict whether $Y_i = 0$.
- Step 2: Use Poisson regression to predict non-zero outcomes.

Hurdle Models



Hurdle Models

```r
hurdle_mod <- pscl::hurdle(visits ~ ., data = nmes)

summary(hurdle_mod)
```

```
Call:
pscl::hurdle(formula = visits ~ ., data = nmes)

Pearson residuals:
    Min      1Q  Median      3Q     Max 
-5.4144 -1.1565 -0.4770  0.5432 25.0228 

Count model coefficients (truncated poisson with log link):
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)      1.406459   0.024180  58.167  < 2e-16 ***
hospital         0.158967   0.006061  26.228  < 2e-16 ***
healthpoor       0.253521   0.017708  14.317  < 2e-16 ***
healthexcellent -0.303677   0.031150  -9.749  < 2e-16 ***
chronic          0.101720   0.004719  21.557  < 2e-16 ***
gendermale      -0.062247   0.013055  -4.768 1.86e-06 ***
school           0.019078   0.001872  10.194  < 2e-16 ***
insuranceyes     0.080879   0.017139   4.719 2.37e-06 ***
Zero hurdle model coefficients (binomial with logit link):
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)      0.043147   0.139852   0.309 0.757688    
hospital         0.312449   0.091437   3.417 0.000633 ***
healthpoor      -0.008716   0.161024  -0.054 0.956833    
healthexcellent -0.289570   0.142682  -2.029 0.042409 *  
chronic          0.535213   0.045378  11.794  < 2e-16 ***
gendermale      -0.415658   0.087608  -4.745 2.09e-06 ***
school           0.058541   0.011989   4.883 1.05e-06 ***
insuranceyes     0.747120   0.100880   7.406 1.30e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Number of iterations in BFGS optimization: 14
```