poisson glm in R

Gavin Fay

2025-02-11

setup

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.
```

```r
library(broom)
```

setup

```r
#first read in the data & create richness variable
RIKZ <- read_table(file = "../data/RIKZ.txt")
```

```
##
## -- Column specification -------------------------------
## cols(
##   .default = col_double()
## )
## i Use `spec()` for the full column specifications.
```

```r
RIKZ <- RIKZ |>
  mutate(Richness = rowSums(RIKZ[,2:76]>0)) |>
  select(Richness, 77:89)
```

Generalized linear modeling

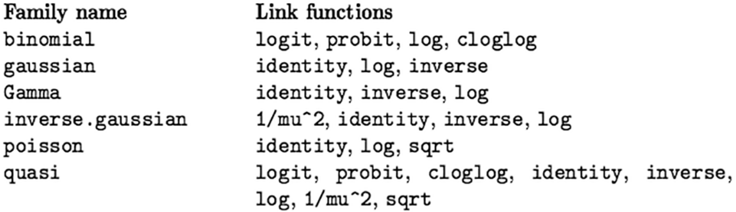
Recall from lecture:

$$\eta = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$$

$$f_Y(y; \mu, \varphi) = \exp\left[\frac{A}{\varphi} y \lambda(\mu) - \gamma(\lambda(\mu)) + \tau(y, \varphi)\right]$$

$$\mu = m(\eta),\ \eta = m^{-1}(\mu) = l(\mu)$$

The combination of a response distribution, a link function and other information needed to carry out the modeling exercise is called the *family* of the generalized linear model.



Family name	**Link functions**
`binomial`	`logit`, `probit`, `log`, `cloglog`
`gaussian`	`identity`, `log`, `inverse`
`Gamma`	`identity`, `inverse`, `log`
`inverse.gaussian`	`1/mu^2`, `identity`, `inverse`, `log`
`poisson`	`identity`, `log`, `sqrt`
`quasi`	`logit`, `probit`, `cloglog`, `identity`, `inverse`, `log`, `1/mu^2`, `sqrt`

The `glm()` function

The `R` function to fit a generalized linear model is `glm()` which uses the form:
`fitted.model <- glm(formula, family=family.generator, data=data.frame)`

Only new piece is the call to 'family.generator'
Although complex, its use is fairly simple.
Where there is a choice of link, link can be supplied with the family name as a parameter.

Simple (inefficient) use: The following are equivalent.

```
> RIKZ_lm1 <- lm(Richness ~ NAP, data = RIKZ)
> RIKZ_glm1 <- glm(Richness ~ NAP, family = gaussian,
+                  data = RIKZ)
```

Most of the extraction functions that can be applied to `lm()` can also be used with `glm()`.

The `glm()` function

```
> tidy(RIKZ_lm1)
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)     6.69     0.658     10.2  5.25e-13
2 NAP            -2.87     0.631     -4.55 4.42e- 5
```

```
> tidy(RIKZ_glm1)
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)     6.69     0.658     10.2  5.25e-13
2 NAP            -2.87     0.631     -4.55 4.42e- 5
```

Poisson regression

$$P(X = x) = \frac{e^{-\mu}\mu^{x}}{x!}, \; \mu_i = e^{\alpha+\beta_1 x_{1,i}+\cdots+\beta_j x_{j,i}}$$

RIKZ example

```
> #fit the poisson regression model
> RIKZ_poisson <- glm(Richness ~ NAP, data = RIKZ,
+                     family = poisson)
```

Note that the default link for the poisson is `log` so we don't have to specify here (see `?family`).

summary(RIKZ_poisson)

```
Call:
glm(formula = Richness ~ NAP, family = poisson, data = RIKZ)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.79100    0.06329  28.297  < 2e-16 ***
NAP         -0.55597    0.07163  -7.762 8.39e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 179.75  on 44  degrees of freedom
Residual deviance: 113.18  on 43  degrees of freedom
AIC: 259.18

Number of Fisher Scoring iterations: 5
```

`broom::augment()`

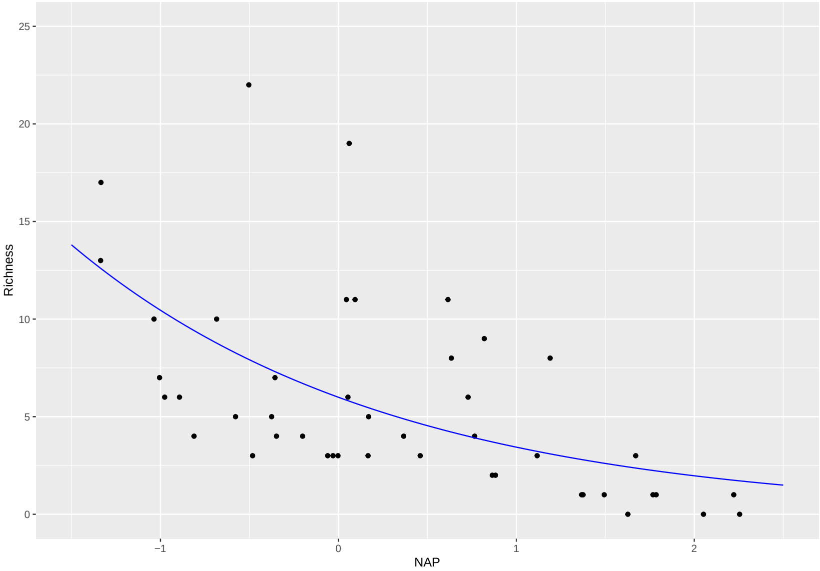
As with `lm()`, the `augment()` function can be used to obtain the predictions from a fitted model object, and for a new data frame.

```
> newobject <- augment(oldobject, newdata,
+                      type.predict = c("link","response","
+                      se_fit = TRUE)
```

Note: default for `type.predict` is on the scale of the linear predictors. Set to `"response"` to obtain predictions on the scale of the response variable.

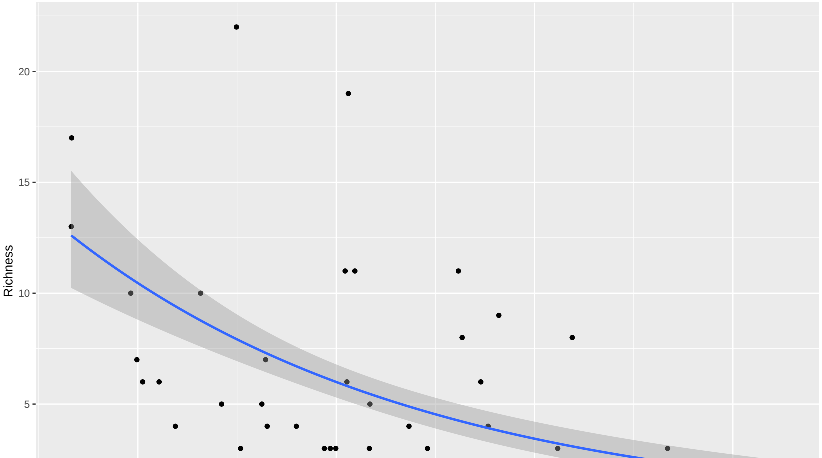
Observed and fitted values for Poisson RIKZ

```
> #create new values for beach height
> new_data <- tibble(NAP = seq(-1.5,2.5,length.out=100))
> #predict values for response based on new salinity
> RIKZ_pois_pred <- augment(RIKZ_poisson,
+                      newdata = new_data,
+                      type.predict = "response",
+                      se_fit = "TRUE")
> #plot
> ggplot(RIKZ, aes(x=NAP, y= Richness)) +
+          geom_point() +
+          geom_line(data = RIKZ_pois_pred,
+                    aes(x=NAP, y = .fitted),
+                    col = "blue") +
+          labs(y = "Richness", x = "NAP") +
+          ylim(0,25) + xlim(-1.5,2.5)
```


25
20
15
10
5
0
Richness
−1
0
1
2
NAP

fitting the GLM in the ggplot call

```
> ggplot(RIKZ,aes(NAP,Richness)) +
+   geom_point() +
+   geom_smooth(method = "glm",
+     method.args = list(family = poisson), se = TRUE)
`geom_smooth()` using formula = 'y ~ x'
```



Quasi-Poisson

```
> RIKZ_quasip_1 <- glm(Richness ~ NAP,
+                      family = quasipoisson,
+                      data = RIKZ)
> summary(RIKZ_quasip_1)

Call:
glm(formula = Richness ~ NAP, family = quasipoisson, data

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.7910     0.1104  16.218  < 2e-16 ***
NAP          -0.5560     0.1250  -4.448 6.02e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '

(Dispersion parameter for quasipoisson family taken to be

    Null deviance: 179.75  on 44  degrees of freedom
```

Full Poisson Model

```
> RIKZ_pois_full <- glm(Richness ~ NAP + factor(week) +
+                          factor(exposure),
+                          data = RIKZ,
+                          family = poisson)
> summary(RIKZ_pois_full)

Call:
glm(formula = Richness ~ NAP + factor(week) + factor(exposu
    family = poisson, data = RIKZ)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         2.53136    0.12866  19.675   <2e-16 **
NAP                -0.48950    0.07449  -6.571    5e-11 **
factor(week)2      -0.75723    0.35132  -2.155   0.0311 *
factor(week)3      -0.50717    0.21148  -2.398   0.0165 *
factor(week)4       0.12361    0.22617   0.547   0.5847
factor(exposure)10 -0.42602    0.19022  -2.240   0.0251 *
```

Drop Exposure?

```
> drop1(RIKZ_pois_full, test = "Chisq")
Single term deletions

Model:
Richness ~ NAP + factor(week) + factor(exposure)
                 Df Deviance    AIC    LRT  Pr(>Chi)
<none>                47.800 203.80
NAP               1   93.460 247.46 45.660 1.407e-11 ***
factor(week)      3   58.372 208.37 10.572   0.01428 *
factor(exposure)  2   53.466 205.46  5.666   0.05885 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```