Chapter 5: Generalized Linear Modeling

Tyson S. Barrett

Summer 2017

Utah State University

Introduction

Logistic Regression

Poisson Regression

Beta Regression

Conclusions

Introduction

Good Quote

"You must stick to your conviction, but be ready to abandon your assumptions."
— Dennis Waitley

GLMs

Generalized Linear Models (GLMs):

1. Are extensions of linear regression to areas where assumptions of normality and homoskedasticity do not hold
2. There are several versions of GLM's, each for different types and distributions of outcomes.

We are going to go through several of the most common GLMs.

Types

We discuss:

1. Logistic Regression
2. Poisson Regression
3. GLM with Gamma distribution
4. Negative binomial
5. Beta Regression

Logistic Regression

Logistic Regression

For binary outcomes (e.g., yes or no, correct or incorrect, sick or healthy)

$$logit(Y) = \beta_0 + \beta_1 X_1 + ... + \epsilon$$

where $logit(Y) = ln\left(\frac{Prob(Y=1)}{1-Prob(Y=1)}\right)$

Prep Data

```r
## First creating binary depression variable
## Use mutate()
df <- df %>%
  mutate(dep = dpq010 + dpq020 + dpq030 + dpq040 + dpq050 +
               dpq060 + dpq070 + dpq080 + dpq090) %>%
  mutate(dep2 = ifelse(dep >= 10, 1,
                ifelse(dep < 10, 0, NA)))
## Fix some placeholders
df <- df %>%
  mutate(asthma = washer(mcq010, 9),
         asthma = washer(asthma, 2, value = 0)) %>%
  mutate(sed = washer(pad680, 9999, 7777))
```

Note:

1. IF depression ≥ 10 then dep2 is 1,
2. IF dpression < 10, then dep2 is 0,
3. ELSE dep2 is NA.

Running Logistic Regression

- βs are in “log-odds”
- e^{β} is an “odds ratio”

In `R`, this is simple.

Running Logistic Regression

```r
l_fit <- glm(dep2 ~ asthma + sed + race + famsize,
             data = df,
             family = "binomial")
summary(l_fit)
```

Running Logistic Regression

```
##
## Call:
## glm(formula = dep2 ~ asthma + sed + race + famsize, family = "binomi
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7831  -0.4479  -0.4078  -0.3645   2.5471
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.6203555  0.2380770 -11.006  < 2e-16 ***
## asthma             0.5688452  0.1276326   4.457 8.32e-06 ***
## sed                0.0005638  0.0002610   2.160   0.0307 *
## raceOtherHispanic  0.7162568  0.2328673   3.076   0.0021 **
## raceWhite          0.1287059  0.2116414   0.608   0.5431
## raceBlack          0.0189205  0.2205461   0.086   0.9316
## raceOther         -0.4901414  0.2570123  -1.907   0.0565 .
## famsize           -0.0318309  0.0373218  -0.853   0.3937
```

Output of Logistic Regression

We used `glm()` (stands for generalized linear model)

- The key to making it logistic, since you can use `glm()` for a linear model using maximum likelihood instead of `lm()` with least squares, is `family = "binomial"`
- Default link in "binomial" is `logit`.
- Can also do `probit` to use probit regression.

Poisson Regression

Poisson Regression

Again, use the `glm()` function.

- The difference here is we will be using an outcome that is a count variable.
- For example, the sedentary variable (`sed`) that we have in `df` is a count of the minutes of sedentary activity.

Running Poisson Regression

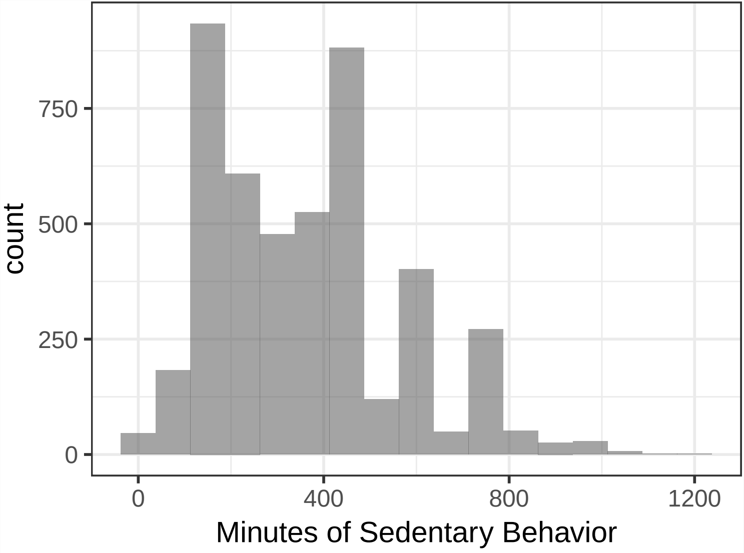
```
p_fit <- glm(sed ~ asthma + race + famsize,
             data = df,
             family = "poisson")
summary(p_fit)
```

Running Poisson Regression

```
##
## Call:
## glm(formula = sed ~ asthma + race + famsize, family = "poisson",
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -27.362   -8.430   -1.477    5.823   34.507
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        5.6499871  0.0035550 1589.31   <2e-16 ***
## asthma             0.0614965  0.0021434   28.69   <2e-16 ***
## raceOtherHispanic  0.1393438  0.0040940   34.04   <2e-16 ***
## raceWhite          0.3484622  0.0033438  104.21   <2e-16 ***
## raceBlack          0.3400346  0.0034430   98.76   <2e-16 ***
## raceOther          0.3557953  0.0036273   98.09   <2e-16 ***
## famsize           -0.0188673  0.0005488  -34.38   <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: 496351  on 4436  degrees of freedom
## Residual deviance: 475428  on 4430  degrees of freedom
##   (195 observations deleted due to missingness)
## AIC: 508999
##
## Number of Fisher Scoring iterations: 5
```

Running Poisson Regression

Sedentary may be over-dispersed:



Running Poisson Regression

and so other methods related to poisson may be necessary.

- See gamma, hurdle models, and negative binomial models next

Gamma

- very similar to poisson but does not require integers and can handle more dispersion.
- the outcome must have values > 0.

Gamma

```r
## Adjust sed
df$sed_gamma <- df$sed + .01
g_fit <- glm(sed_gamma ~ asthma + race + famsize,
             data = df,
             family = "Gamma")
summary(g_fit)
```

Gamma

```
##
## Call:
## glm(formula = sed_gamma ~ asthma + race + famsize, family = "Gamma",
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3589  -0.4613  -0.0845   0.2926   1.6868
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.567e-03  1.132e-04  31.515  < 2e-16 ***
## asthma            -1.604e-04  5.865e-05  -2.735  0.00626 **
## raceOtherHispanic -4.874e-04  1.309e-04  -3.723  0.00020 ***
## raceWhite         -1.090e-03  1.078e-04 -10.115  < 2e-16 ***
## raceBlack         -1.068e-03  1.102e-04  -9.697  < 2e-16 ***
## raceOther         -1.110e-03  1.145e-04  -9.695  < 2e-16 ***
## famsize            5.107e-05  1.552e-05   3.289  0.00101 **
## ---
```

Two-Part or Hurdle Models

- Use the `pscl` package to run a hurdle model.
- These models are built for situations where there is a count variable with many zeros (“zero-inflated”).
- The hurdle model makes slightly different assumptions regarding the zeros than the pure negative binomial that we present next.
- The hurdle consists of two models: one for whether the person had a zero or more (binomial) and if more than zero, how many (poisson).

To run a hurdle model, we are going to make a sedentary variable with many more zeros to illustrate and then we will run a hurdle model.

Two-Part or Hurdle Models

```
## Zero inflated sedentary (don't worry too much about the specifics)
df$sed_zero <- ifelse(sample(1:100,
                             size = length(df$sed),
                             replace=TRUE) %in% c(5,10,11,20:25), 0,
                      df$sed)
## Hurdle model
library(pscl)
h_fit = hurdle(sed_zero ~ asthma + race + famsize,
               data = df)
summary(h_fit)
```

Two-Part or Hurdle Models

```
##
## Call:
## hurdle(formula = sed_zero ~ asthma + race + famsize, data = df)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -3.9248 -1.4783 -0.2191  1.2563 11.0364
##
## Count model coefficients (truncated poisson with log link):
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        5.6727458  0.0036627 1548.80   <2e-16 ***
## asthma             0.0627030  0.0022628   27.71   <2e-16 ***
## raceOtherHispanic  0.1201634  0.0042592   28.21   <2e-16 ***
## raceWhite          0.3248979  0.0034416   94.40   <2e-16 ***
## raceBlack          0.3337217  0.0035384   94.31   <2e-16 ***
## raceOther          0.3359265  0.0037427   89.75   <2e-16 ***
## famsize           -0.0200684  0.0005781  -34.71   <2e-16 ***
## Zero hurdle model coefficients (binomial with logit link):
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        2.84791    0.23592  12.072   <2e-16 ***
## asthma            -0.20907    0.13695  -1.527   0.1269
## raceOtherHispanic -0.57535    0.25379  -2.267   0.0234 *
## raceWhite         -0.48597    0.22052  -2.204   0.0275 *
## raceBlack         -0.31269    0.22953  -1.362   0.1731
## raceOther         -0.37082    0.24153  -1.535   0.1247
## famsize           -0.05421    0.03545  -1.529   0.1262
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 12
## Log-likelihood: -2.307e+05 on 14 Df
```

Hurdle Models

Notice that the output has two parts:

1. “Count model coefficients (truncated poisson with log link):” and
2. “Zero hurdle model coefficients (binomial with logit link):”.

Together they tell us about the relationship between the predictors and a count variable with many zeros.

Negative Binomial

- negative binomial is also for zero-inflated count variables.
- It makes slightly different assumptions than the hurdle and doesn't use a two-part approach.
- Use the `MASS` package and the `glm.nb()` function.

```
library(MASS)
fit_nb <- glm.nb(sed_zero ~ asthma + race + famsize,
                 data = df)
summary(fit_nb)
```

Negative Binomial

Note that this model is not really appropriate because our data is somewhat contrived.

Beta Regression

Beta Regression

- For outcomes that are bound between a lower and upper bound
- For example, if we are looking at test scores that are bound between 0 and 100.
- It is a very flexible method and allows for some extra analysis regarding the variation.

Running Beta Regression

- Use the betareg package.
- But first, we are going to reach a little and create a ficticiously bound variable in the data set.

```
## Variable bound between 0 and 1
df$beta_var <- sample(seq(.05, .99, by = .01),
                      size = length(df$asthma),
                      replace = TRUE)
library(betareg)
fit_beta <- betareg(beta_var ~ asthma + race + famsize,
                    data = df)
summary(fit_beta)
```

Running Beta Regression

```
##
## Call:
## betareg(formula = beta_var ~ asthma + race + famsize, data = df)
##
## Standardized weighted residuals 2:
##     Min      1Q  Median      3Q     Max
## -2.0364 -0.6739 -0.0598  0.6311  2.9235
##
## Coefficients (mean model with logit link):
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        0.195018   0.063399   3.076   0.0021 **
## asthma            -0.057137   0.043789  -1.305   0.1920
## raceOtherHispanic -0.053873   0.072158  -0.747   0.4553
## raceWhite         -0.025079   0.058871  -0.426   0.6701
## raceBlack         -0.059966   0.061116  -0.981   0.3265
## raceOther         -0.077520   0.065502  -1.183   0.2366
## famsize           -0.009472   0.010843  -0.874   0.3824
##
## Phi coefficients (precision model with identity link):
##       Estimate Std. Error z value Pr(>|z|)
## (phi)  2.45394    0.04465   54.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Type of estimator: ML (maximum likelihood)
## Log-likelihood:  83.2 on 8 Df
## Pseudo R-squared: 0.00112
## Number of iterations: 15 (BFGS) + 1 (Fisher scoring)
```

Beta Regression

- The output provides coefficients and the “Phi” coefficients.
- Both are important parts of using beta regression but we are not going to discuss it here.

Conclusions

Conclusions

There are many resources available to learn more about each of these GLM's.

As for now, we are going to move on to more complex modeling where there are clustering or repeated measures in the data.

Conclusions

One of the great things about `R` is that most modeling is very similar to the basic `lm()` function.

- In all of these GLM's the arguments are nearly all the same:
    - a formula,
    - the data, and
    - family of model.
- As you'll see for Multilevel and Other Models chapters, this does not change much.
- Having a good start with basic models and GLM's gets you ready for nearly every other modeling type in `R`.