Search Count Model 2.0

Gloria Grace

October 20, 2022

Data

```
search <- read_csv('https://sldr.netlify.app/data/election_searches.csv
```

Fit

```r
search_nb1 <- glmmTMB(Searches ~ Age + Education + Sex, data = search,
                      family = nbinom1(link = 'log'))
```

```r
summary(search_nb1)
```

```
##  Family: nbinom1  ( log )
## Formula:          Searches ~ Age + Education + Sex
## Data: search
##
##      AIC      BIC   logLik deviance df.resid
##  17705.6  17763.4  -8842.8  17685.6     2396
##
##
## Dispersion parameter for nbinom1 family (): 12.7
##
## Conditional model:
```

```
##                                          Estimate Std. Error z val
## (Intercept)                              2.596764   0.125007  20.7
## Age                                     -0.009080   0.001303  -6.9
## EducationAdvanced                        0.418309   0.121829   3.4
## EducationBachelors                       0.356933   0.116310   3.0
## EducationHigh school graduate            0.131943   0.123158   1.0
## EducationLess than 9th grade            -0.211429   0.646757  -0.3
## EducationSome college or associate degree 0.315713   0.115774   2.7
## SexMale                                  0.114891   0.033337   3.4
## SexOther                                -0.091803   0.357553  -0.2
##
## (Intercept)                             ***
## Age                                     ***
## EducationAdvanced                       ***
## EducationBachelors                      **
## EducationHigh school graduate
## EducationLess than 9th grade
## EducationSome college or associate degree **
## SexMale                                 ***
## SexOther
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Prediction Plot

Hypothetical data

```
hyp_search <- expand.grid(Age = 26,
                          Education = c('Bachelors'),
                          Sex = c('Male', 'Female'))
```

Make predictions on link scale

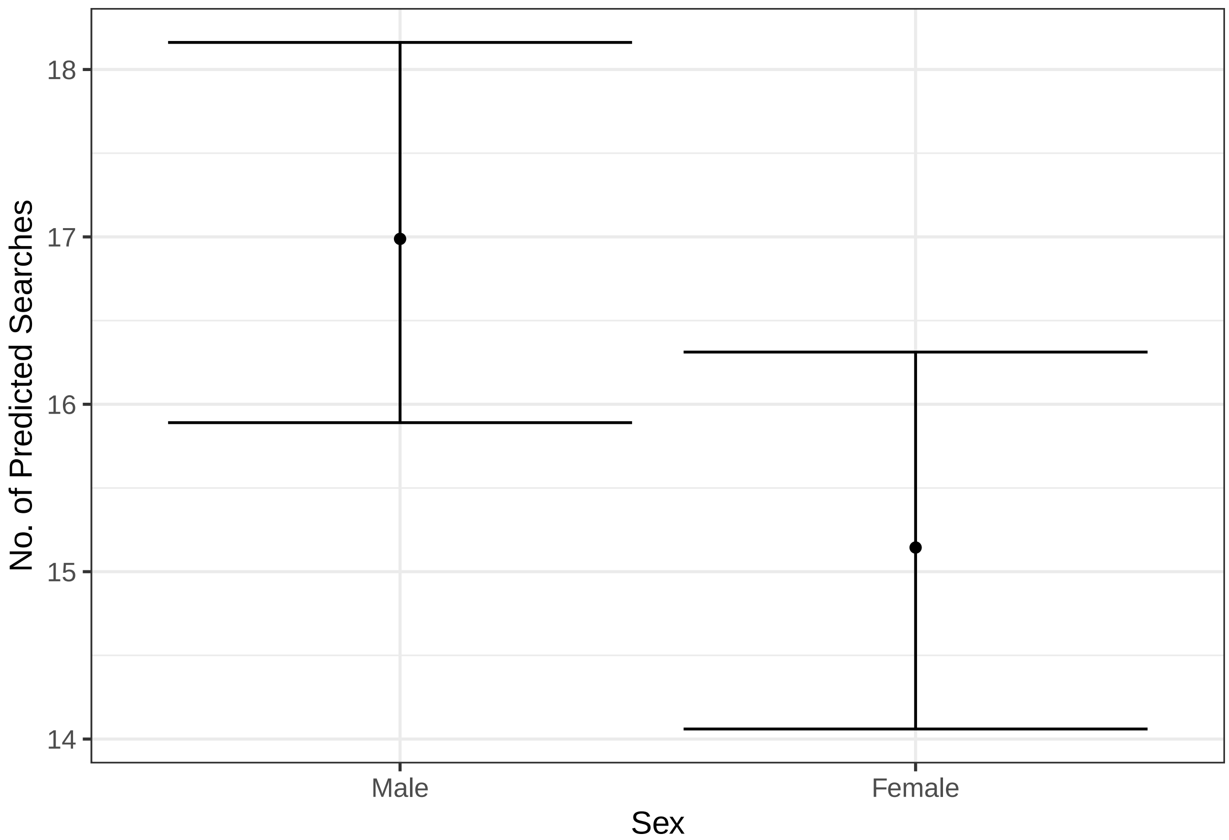
```
search_preds <- predict(search_nb1,
                        newdata = hyp_search,
                        type = 'link',
                        se.fit = TRUE)
```

Compute on link scale

```
hyp_search <- hyp_search |>
  mutate(pred = exp(search_preds$fit),
         ci_lower = exp(search_preds$fit - 1.96*search_preds$se.fit),
         ci_upper = exp(search_preds$fit + 1.96*search_preds$se.fit))
```

Draw the plot

```
gf_point(pred ~ Sex,
         data = hyp_search) |>
  gf_errorbar(ci_lower + ci_upper ~ Sex) |>
  gf_labs(y = 'No. of Predicted Searches')
```



Model Selection

```r
search_nb2 <- glmmTMB(Searches ~ Age + Party + Sex, data = search,
                      family = nbinom1(link = 'log'))
```

```r
search_nb3 <- glmmTMB(Searches ~ Age + Vote_Sway + Sex, data = search,
                      family = nbinom1(link = 'log'))
```

Check which is better:

```r
AIC(search_nb1, search_nb2, search_nb3)
```

```
##            df      AIC
## search_nb1 10 17705.57
## search_nb2  9 17724.72
## search_nb3  6 17728.85
```

Based on the AIC test above on three different model which differs on one particular predictor, the model of negative binomial one with the Sex, Age, and Education predictor was the best fit.