Chapter 14: Generalised Linear Models

Answers to Exercises

Brian Fogarty

Contents

Exercise 1

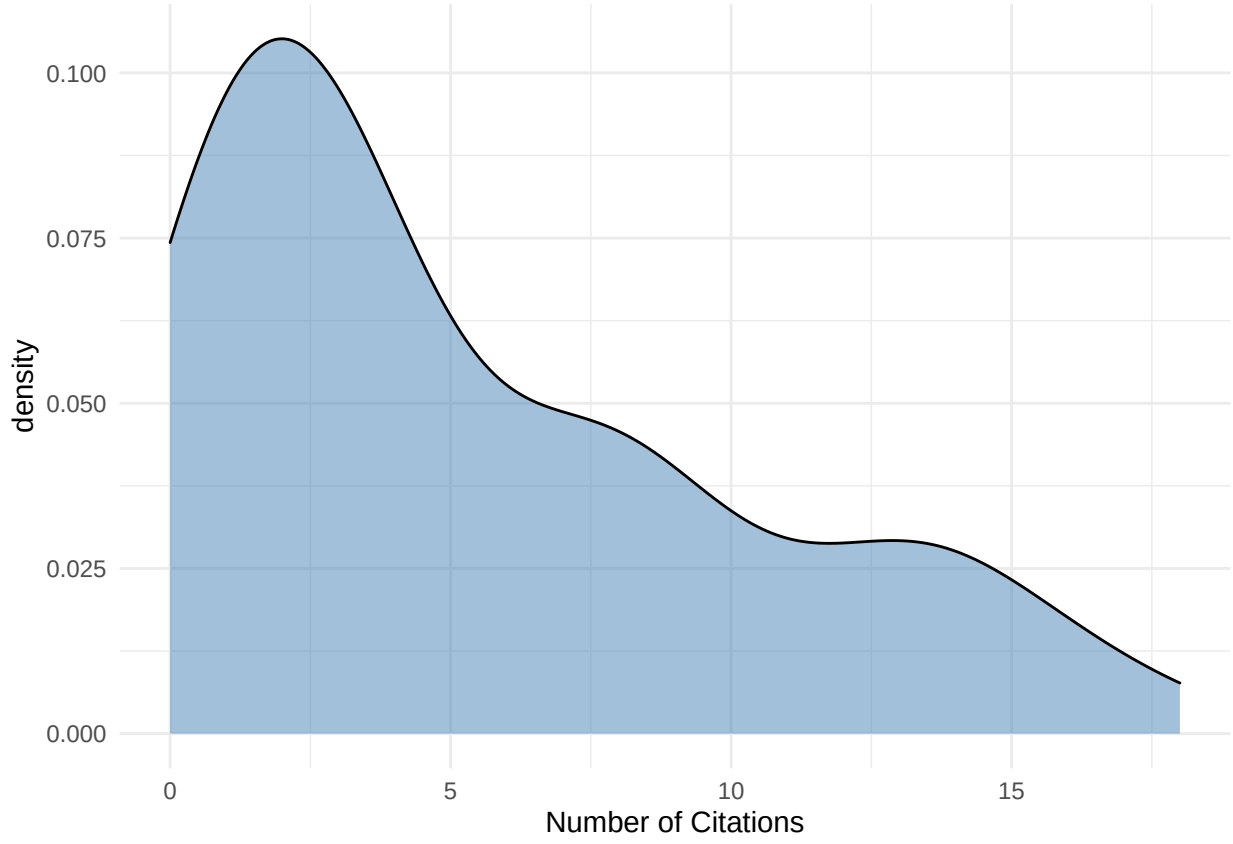
Exercise 1.a

```
library(tidyverse)

citations <- read_csv("citations_twitter.csv")
glimpse(citations)
```

```
Rows: 308
Columns: 7
$ articlecites    <dbl> 8, 0, 13, 1, 2, 14, 1, 10, 6, 3, 13, 12, 3, 3, 14, 7, ~
$ womanleadauthor <dbl> 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, ~
$ womanauthor     <dbl> 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, ~
$ tweet_dum       <dbl> 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, ~
$ fullprof        <dbl> 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1, ~
$ retweets        <dbl> 2, 0, 2, NA, NA, 0, NA, 35, 0, NA, 3, 5, NA, NA, 0, 1,~
$ totfav          <dbl> 6, 0, 7, NA, NA, 1, NA, 64, 1, NA, 8, 5, NA, NA, 1, 2,~
```

We can use the `filter()` function to create a smoothed density plot of citations for only tweeted articles.

```
citations %>%
  filter(tweet_dum==1 & !is.na(articlecites)) %>%
ggplot() +
  geom_density(mapping = aes(articlecites), fill = "steelblue", alpha = .5) +
  labs(x = "Number of Citations") +
  theme_minimal()
```



This distribution takes on a similar shape as the one in Chapter 14, but is more compact and suggests no specific problems (e.g., excess zeros).

Exercise 1.b

```
summary(model.prm <- glm(articlecites ~ womanauthor + fullprof +
                           retweets, family = "poisson", data = citations))
```

```
Call:
glm(formula = articlecites ~ womanauthor + fullprof + retweets,
    family = "poisson", data = citations)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.3686  -1.8364  -0.7725   1.0267   4.6280
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.567006   0.084048  18.644  < 2e-16 ***
womanauthor -0.007300   0.100667  -0.073  0.94219
fullprof     0.168852   0.101394   1.665  0.09585 .
retweets     0.006679   0.002070   3.227  0.00125 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 330.72  on 74  degrees of freedom
Residual deviance: 319.61  on 71  degrees of freedom
  (233 observations deleted due to missingness)
AIC: 550.06

Number of Fisher Scoring iterations: 5
```

We see that `retweets` has a positive statistically significant effect on `citations`. Notice that 233 observations (out of 308 observations) are excluded from our model; thus, our N is only 75 observations. This large drop in observations is due to the number of missing values (`NA`s) in `retweet`. Given the model's relatively small N, we could also say that `fullprof` has a positive statistically significant (at the .10-level) on `citations`. Whether or not to consider `fullprof` "statistically significant" is a researcher decision.

Exercise 1.c

We now test for overdispersion with the `dispersiontest()` function from the `AER` package.

```
library(AER)
dispersiontest(model.prm)
```

```
    Overdispersion test

data:  model.prm
z = 4.9427, p-value = 3.852e-07
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
  4.285182
```

Yep, overdispersion is present and we should use a negative binomial model (or, a different count model).

Exercise 1.d

```
library(MASS)

summary(model.nbrm <- glm.nb(articlecites ~ womanauthor + fullprof +
                          retweets, data=citations))
```

```
Call:
glm.nb(formula = articlecites ~ womanauthor + fullprof + retweets,
    data = citations, init.theta = 1.39840135, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1235  -0.9206  -0.3560   0.4323   1.7991

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.554637   0.181103   8.584   <2e-16 ***
womanauthor 0.028061   0.222338   0.126    0.900
fullprof    0.142494   0.226845   0.628    0.530
retweets    0.007210   0.005887   1.225    0.221
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.3984) family taken to be 1)

    Null deviance: 87.519  on 74  degrees of freedom
Residual deviance: 85.447  on 71  degrees of freedom
  (233 observations deleted due to missingness)
AIC: 421.77

Number of Fisher Scoring iterations: 1

              Theta:  1.398
          Std. Err.:  0.306

 2 x log-likelihood:  -411.774
```

The `model.nbrm` results show that none of our predictors have a statistically significant effect on `citations`.

Exercise 1.e

Although we already know that problematic overdispersion is present, we'll quickly check that `model.nbrm` is preferred over `model.prm`.

```
logLik(model.nbrm)
```

```
'log Lik.' -205.8871 (df=5)
```

```
logLik(model.prm)
```

```
'log Lik.' -271.0324 (df=4)
```

We see that our NBRM has a much larger log-likelihood (closer to 0) indicating a better fit to the data.

```
lrtest(model.prm, model.nbrm)
```

```
Likelihood ratio test
```

```
Model 1: articlecites ~ womanauthor + fullprof + retweets
Model 2: articlecites ~ womanauthor + fullprof + retweets
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   4 -271.03
2   5 -205.89  1 130.29  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that $p \leq 0.05$ and thus we conclude that `model.nbrm` is preferred over `model.prm`, according to the likelihood-ratio test.

```
c(model.prm$aic, model.nbrm$aic)
```

```
[1] 550.0649 421.7743
```

`model.nbrm`'s AIC value is much smaller than `model.prm`'s AIC value and thus is the preferred model.

Exercises 1.f-1.g

N/A because there are no statistically significant predictors.

Exercise 2

Exercise 2.a

```
ferg <- read_csv("Ferguson International News.csv")
glimpse(ferg)
```

```
Rows: 57
Columns: 5
$ country  <chr> "Algeria", "Argentina", "Australia", "Austria", "Bangladesh",~
$ amount   <dbl> 11, 8, 24, 7, 8, 57, 19, 31, 2, 16, 14, 3, 6, 31, 83, 1, 2, 2~
$ gini     <dbl> 29, 44, 34, 30, 32, 33, 33, 53, 32, 34, 51, NA, 26, 27, 46, 4~
$ efrac    <dbl> 0.4353, 0.3073, 0.3159, 0.1263, 0.0050, 0.5512, 0.6972, 0.070~
$ soc_safe <dbl> 2.426, 2.221, 1.368, 1.176, 2.559, 1.441, 2.147, 2.897, 2.632~
```

```
summary(ferg$amount)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    3.00    9.00   19.19   25.00   89.00
```

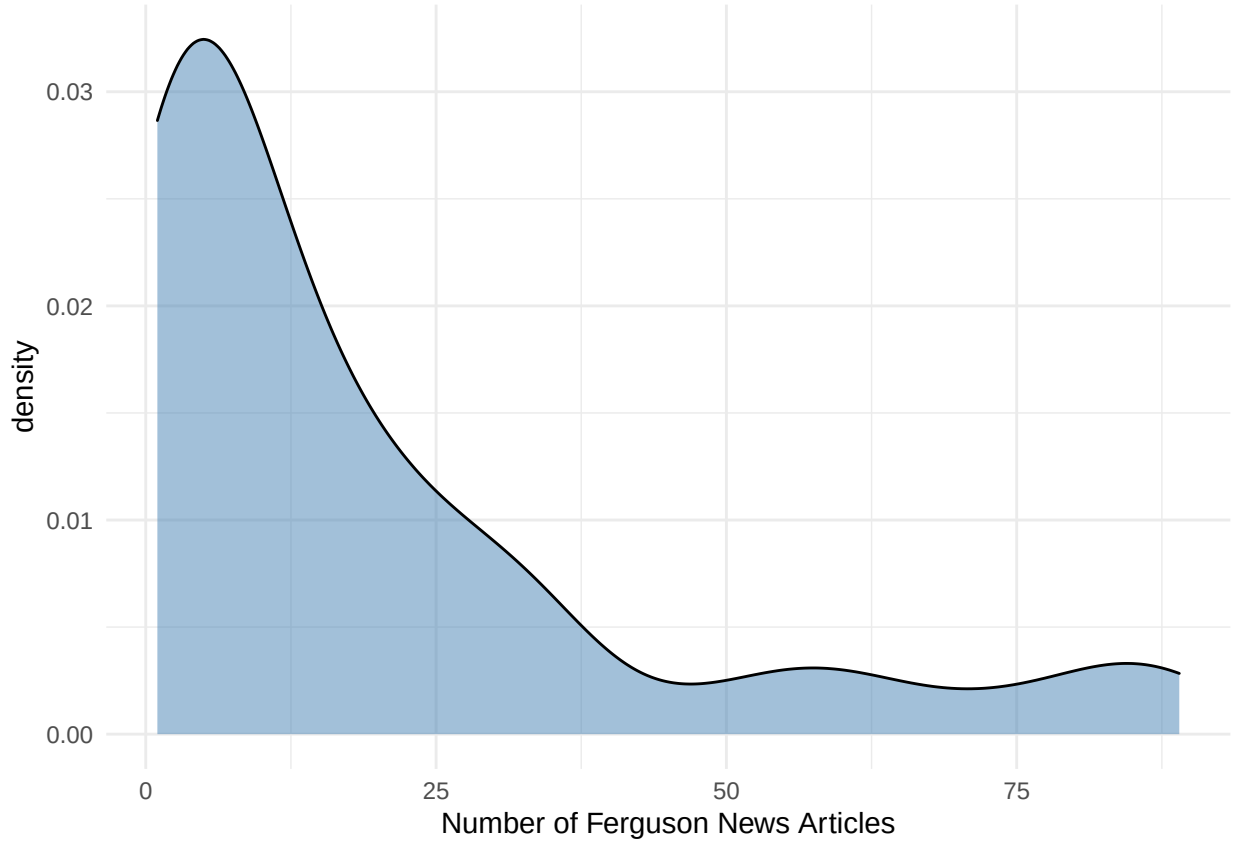
```
var(ferg$amount,na.rm=TRUE)
```

```
[1] 549.9085
```

We see the variance is much larger than the mean. This suggests we may have a problem with overdispersion.

Exercise 2.b

```
ferg %>%
ggplot() +
  geom_density(mapping = aes(amount), fill = "steelblue", alpha = .5) +
  labs(x = "Number of Ferguson News Articles") +
  theme_minimal()
```



We see that most newspapers have a small number of articles on Ferguson and a few newspapers have a relatively large number of articles on Ferguson. This is a very typical count distribution. There is nothing in distribution that suggests obvious potential problems.

Exercise 2.c

```
summary(model.prm <- glm(amount ~ gini + efrac + soc_safe,
          family = "poisson", data = ferg))
```

```
Call:
glm(formula = amount ~ gini + efrac + soc_safe, family = "poisson",
    data = ferg)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-7.810  -3.330  -2.006   1.619  11.921

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.568914   0.150158  30.427  < 2e-16 ***
gini        -0.002758   0.006204  -0.445    0.657
efrac       -0.997452   0.142135  -7.018 2.26e-12 ***
soc_safe    -0.601312   0.075294  -7.986 1.39e-15 ***
---
```

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

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1166.68  on 47  degrees of freedom
Residual deviance:  861.13  on 44  degrees of freedom
  (9 observations deleted due to missingness)
AIC: 1066.7

Number of Fisher Scoring iterations: 5
```

We see that `efrac` and `soc_safe` have a statistically significant effect on `amount`.

```
dispersiontest(model.prm)
```

```
    Overdispersion test

data:  model.prm
z = 3.0838, p-value = 0.001022
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
  20.44675
```

Overdispersion is present and we should use a negative binomial model (or, a different count model).

Exercise 2.d

```
summary(model.nbrm <- glm.nb(amount ~ gini + efrac + soc_safe, data = ferg))
```

```
Call:
glm.nb(formula = amount ~ gini + efrac + soc_safe, data = ferg,
    init.theta = 1.10632454, link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9458  -1.1913  -0.5332   0.4914   2.2854

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.716737   0.631891   7.464 8.36e-14 ***
gini         0.003425   0.022928   0.149   0.8813
efrac       -1.341639   0.537840  -2.494   0.0126 *
soc_safe    -0.739692   0.289954  -2.551   0.0107 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.1063) family taken to be 1)

    Null deviance: 73.035  on 47  degrees of freedom
```

```
Residual deviance: 52.102  on 44  degrees of freedom
  (9 observations deleted due to missingness)
AIC: 378.01

Number of Fisher Scoring iterations: 1

              Theta:  1.106
          Std. Err.:  0.224

 2 x log-likelihood:  -368.005
```

With NBRM, we see that `efrac` and `soc_safe` are still statistically significant.

Exercise 2.e

```
logLik(model.nbrm)
```

```
'log Lik.' -184.0026 (df=5)
```

```
logLik(model.prm)
```

```
'log Lik.' -529.3545 (df=4)
```

We see that our NBRM has a much larger log-likelihood (closer to 0) indicating a better fit to the data.

```
lrtest(model.prm, model.nbrm)
```

```
Likelihood ratio test

Model 1: amount ~ gini + efrac + soc_safe
Model 2: amount ~ gini + efrac + soc_safe
  #Df  LogLik Df Chisq Pr(>Chisq)
1   4 -529.35
2   5 -184.00  1 690.7  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that $p \leq 0.05$ and thus we conclude that `model.nbrm` is preferred over `model.prm`, according to the likelihood-ratio test.

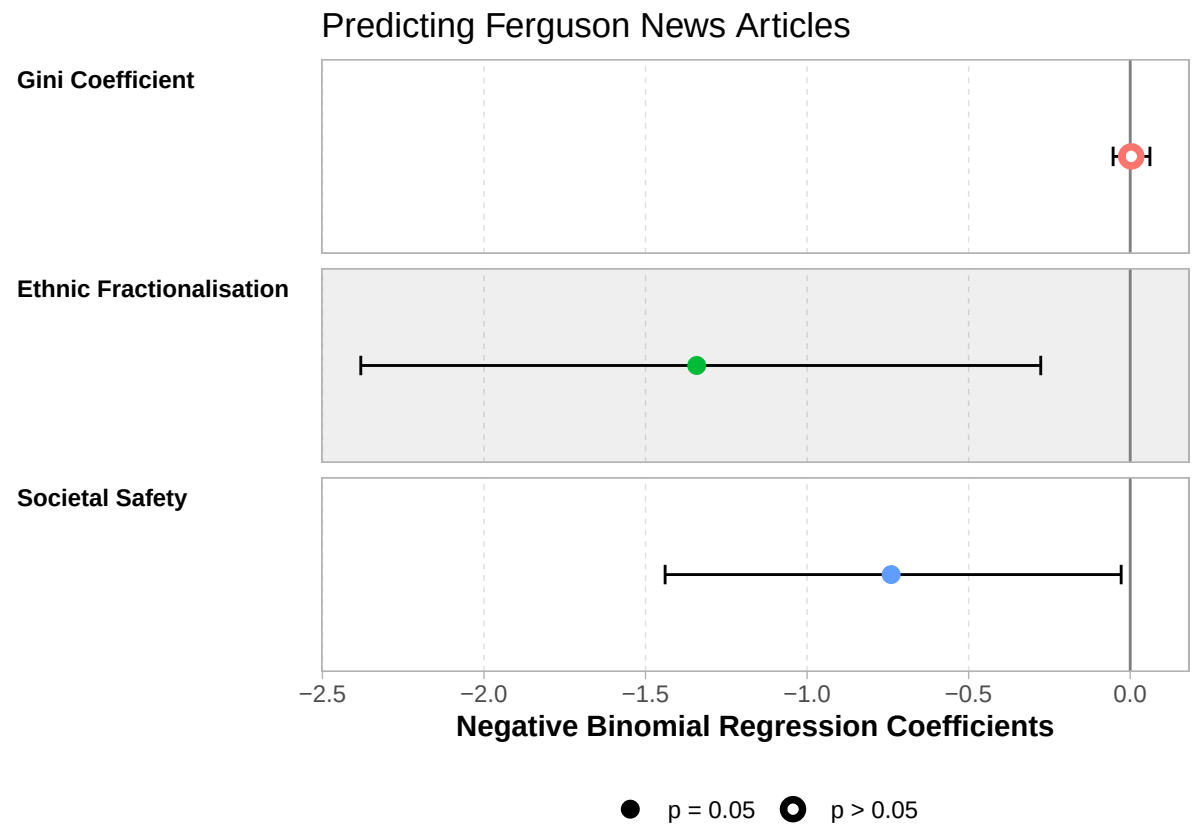
```
c(model.prm$aic, model.nbrm$aic)
```

```
[1] 1066.7090  378.0052
```

`model.nbrm`'s AIC value is much smaller than `model.prm`'s AIC value and thus is the preferred model.

Exercise 2.f

```
library(GGally)
ggcoef_model(model.nbrm,
               variable_labels = c(
                 gini = "Gini Coefficient",
                 efrac = "Ethnic Fractionalisation",
                 soc_safe = "Societal Safety"),
               show_p_values = FALSE,
               signif_stars = FALSE) +
  labs(title = "Predicting Ferguson News Articles",
       x = "Negative Binomial Regression Coefficients")
```



Exercise 2.g

The interpretations in Chapter 14 were for dummy variables, but here we have ratio-level variables. So, we'll interpret the predicted counts for the lowest and highest values of `efrac` and `soc_safe`.

```
library(ggeffects)

summary(ferg$efrac)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 0.0041  0.0616  0.2677  0.3301  0.5469  0.9250       3
```

```
ggpredict(model.nbrm, terms = "efrac[.0041,.925]")

# Predicted counts of amount
```

```
efrac | Predicted |         95% CI
--------------------------------
 0.00 |     26.31 | [16.65, 41.57]
 0.92 |      7.65 | [ 3.91, 14.95]

Adjusted for:
*     gini = 36.88
* soc_safe =  2.12
```

The predicted number of articles for newspapers in countries with the lowest ethnic fractionalisation is 26.31. The predicted number of articles for newspapers in countries with the highest ethnic fractionalisation is 7.65. Thus, newspapers in countries with the highest ethnic fractionalisation are expected to have 18.66 fewer articles on Ferguson than newspapers in countries with the lowest ethnic fractionalisation.

```
summary(ferg$soc_safe)
```

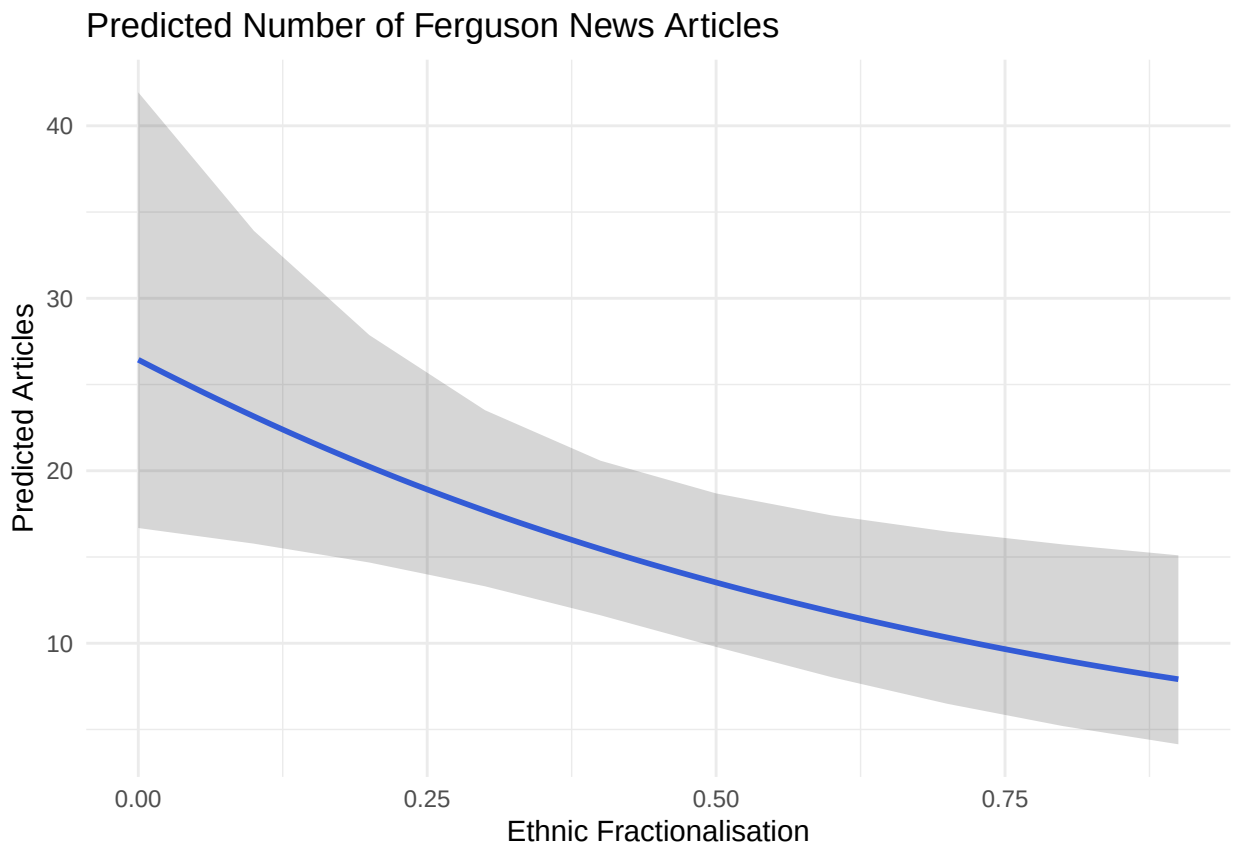
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  1.088   1.433   2.059   2.082   2.632   3.441       2
```

```
ggpredict(model.nbrm, terms = "soc_safe[1.088,3.441]")

# Predicted counts of amount

soc_safe | Predicted |         95% CI
-------------------------------------
    1.09 |     35.20 | [18.50, 66.97]
    3.44 |      6.18 | [ 2.75, 13.86]

Adjusted for:
*  gini = 36.88
* efrac =  0.36
```

The predicted number of articles for newspapers in countries with the highest level of societal safety is 35.2. The predicted number of articles for newspapers in countries with the lowest level of societal safety is 6.18. Thus, newspapers in countries with the highest level of societal safety are expected to have 29.02 fewer articles on Ferguson than newspapers in countries with the lowest level of societal safety.

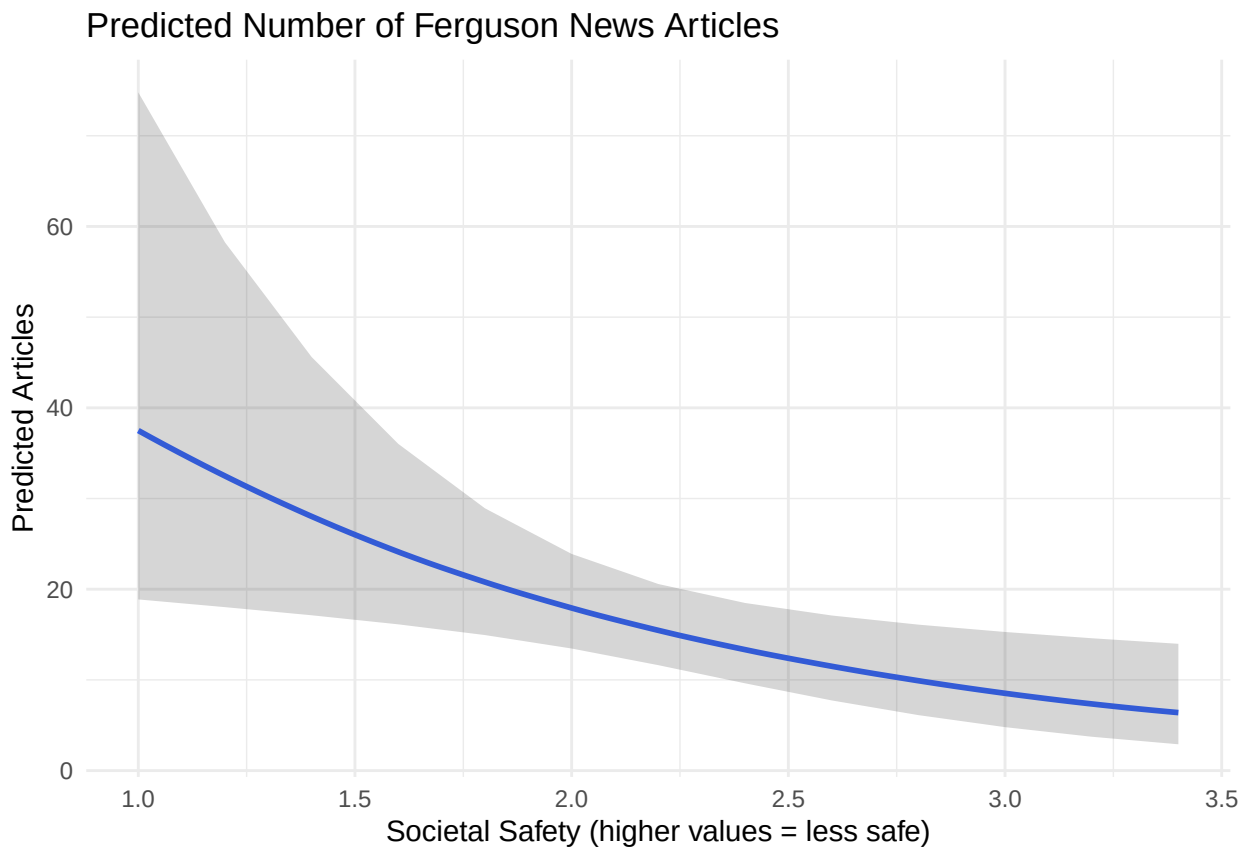
Exercise 2.h

```
ggpredict(model.nbrm, terms = "efrac") %>%
ggplot(mapping = aes(x = x, y = predicted)) +
  geom_smooth(se = FALSE) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .2) +
  labs(title = "Predicted Number of Ferguson News Articles",
       x = "Ethnic Fractionalisation", y = "Predicted Articles") +
  theme_minimal()
```



The plot shows that as ethnic fractionalisation increases, the predicted number of articles on Ferguson decrease.

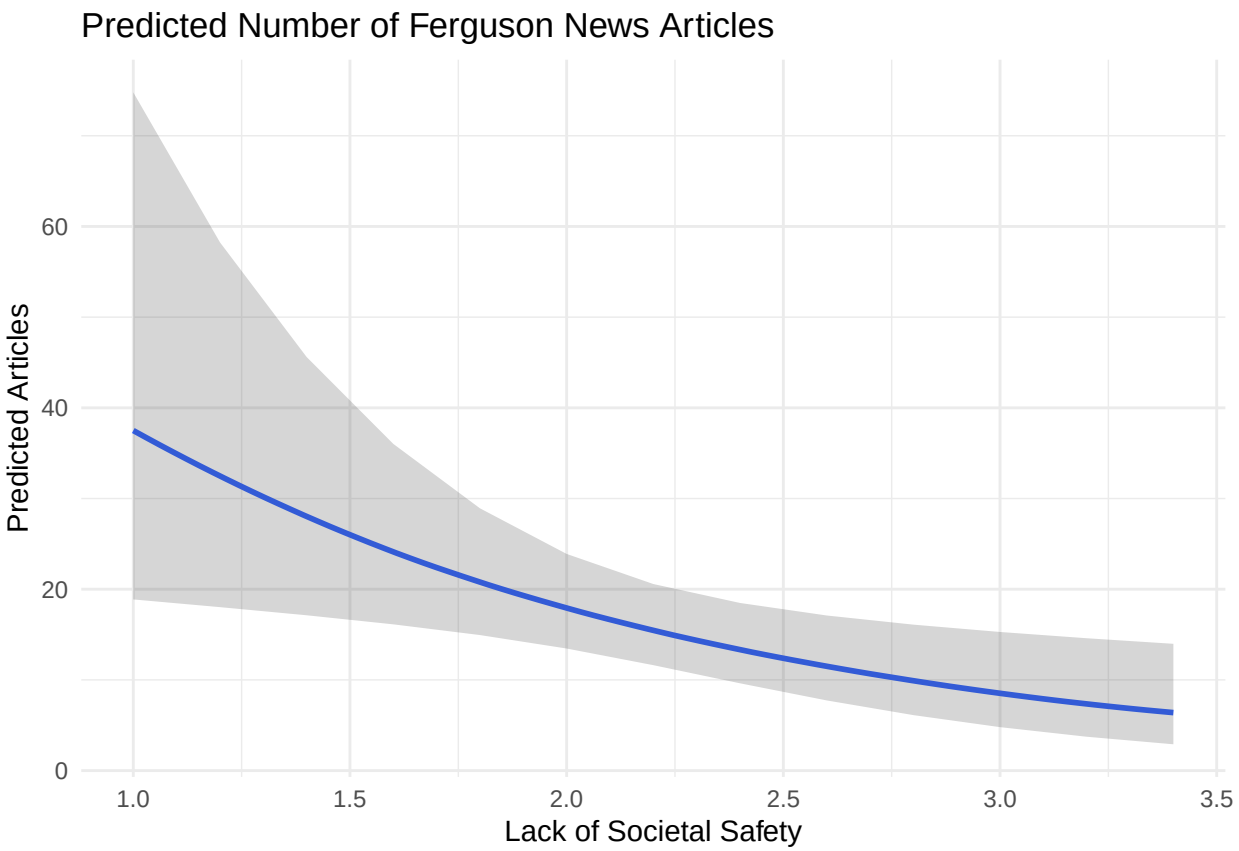
```
ggpredict(model.nbrm, terms = "soc_safe") %>%
ggplot(mapping = aes(x = x, y = predicted)) +
  geom_smooth(se = FALSE) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .2) +
  labs(title = "Predicted Number of Ferguson News Articles",
       x = "Societal Safety (higher values = less safe)", y = "Predicted Articles") +
  theme_minimal()
```



The plot shows that as societal safety decreases (means increasing on the societal safety measure), the predicted number of articles on Ferguson decreases.

This phrasing is a little awkward, so we could alter the label on the x-axis to make the discussion more intuitive.

```
ggpredict(model.nbrm, terms = "soc_safe") %>%
ggplot(mapping = aes(x = x, y = predicted)) +
  geom_smooth(se = FALSE) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = .2) +
  labs(title = "Predicted Number of Ferguson News Articles",
       x = "Lack of Societal Safety", y = "Predicted Articles") +
  theme_minimal()
```



The plot shows that as the lack of societal safety increases, the predicted number of articles on Ferguson decreases.