Homework 4

Sonya Eason

```
if (!require(pacman)) install.packages("pacman")
```

```
Loading required package: pacman
```

```
pacman::p_load(knitr, broom, tidyverse)
```

```
library(tidyverse)
library(knitr)
library(broom)
```

```
crab <- read_csv("data/crab.csv")
```

```
Rows: 173 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (5): Color, Spine, Width, Satellite, Weight

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
ambiguity <- read_csv("data/ambiguity.csv")
```

```
Rows: 870 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (1): name
dbl (10): ambiguity, distID, ideology, totalIssuePages, democrat, mismatch, ...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Exercise 1

a) The response is the number of fishes caught by each visitor visitor in their one week stay. We sare specifically looking at hte mean number.

b) The possible values are 0, 1, 2, 3, up to the maximum number fish there are in the state wildlife park.

c) Lambda represents the mean number of fish caught per week, which is 21.5.

d) A zero-inflated model could be considered here since there are likely a lot of zeroes representing people who caught zero fish during their stay, and there are also two subgroups of zeroes. These zeroes can be separated into two subgroups the people who never fish, i.e. the true zeroes., and those who just didn't catch fish on their trips this time.

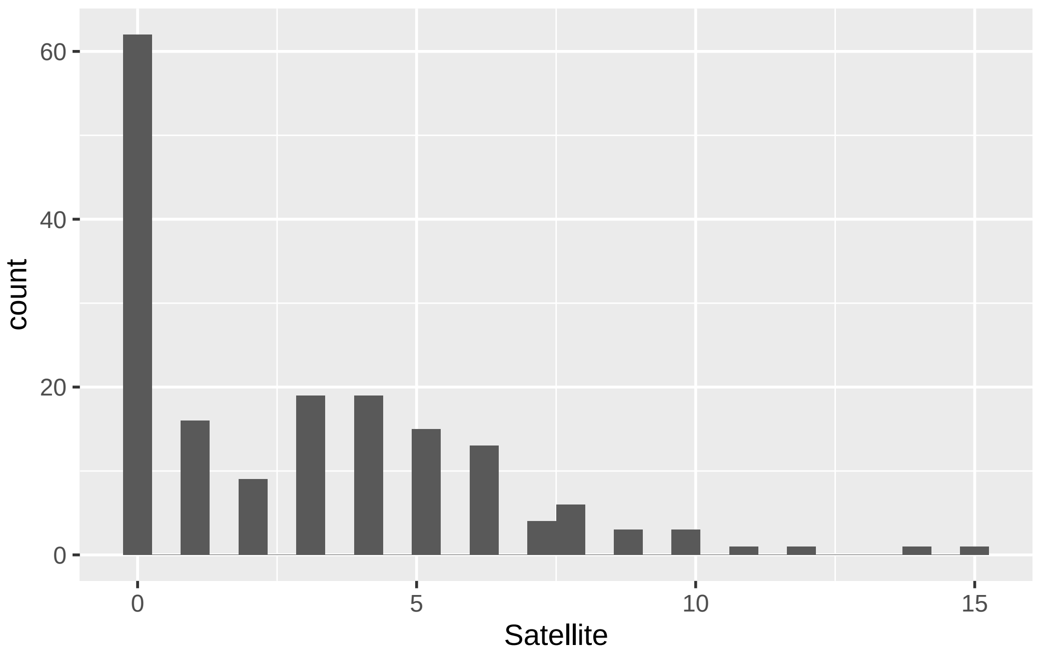
Exercise 2

a)

```
crab <- crab |>
  mutate(
    Color = factor(Color),
    Spine = factor(Spine)
  )
```

```
ggplot(crab, aes(x = Satellite)) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Is there preliminary evidence the number of satellites could be modeled as a Poisson response? Briefly explain

In a Poisson response, we do expe…

right-skewed, bounded on left, unbounded, count, poisson

b)

```
sat_model <- glm(Satellite ~ Width + Weight + Spine, family = poisson, data = crab)


sat_model |>
  tidy(conf.int = T) |>
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-1.062	0.928	-1.144	0.253	-2.875	0.763
Width	0.039	0.048	0.816	0.415	-0.055	0.132
Weight	0.000	0.000	2.771	0.006	0.000	0.001
Spine2	-0.214	0.211	-1.017	0.309	-0.644	0.185
Spine3	-0.049	0.108	-0.458	0.647	-0.257	0.165

c) When a female crab has one worn or broken spine, the number of satellites is expected to change by a multiplicative factor of 0.8073484 compared to when the female crab has two spines that are both in good condition, holding all else constant. When a female crab has two worn or broken spines, the number of satellites is expected to change by a multiplicative factor of 0.9521811 compared to when the female crab has two spines that are both in good condition, holding all else constant.

Exercise 3

```
crab |>
  group_by(Spine) |>
  summarize(mean = mean(Satellite),
            var = var(Satellite))
```

```
# A tibble: 3 x 3
  Spine  mean   var
  <fct> <dbl> <dbl>
1 1      3.65 11.5 
2 2      2     5.57
3 3      2.81  9.82
```

A quasi-Poisson regression is suitable because there is evidence of overdispersion seen by variances that are larger than means at each level. In a typical Poisson model, we'd expect mean = variance, so we'd use quasi-Poison to account for the fact that that does not hold here.

b)

```
sat_model_2 <- glm(Satellite ~ Width + Weight + Spine, family = quasipoisson, data = crab)

sat_model_2 |>
tidy(conf.int = TRUE) |>
  kable(digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-1.062	1.652	-0.643	0.521	-4.281	2.195
Width	0.039	0.085	0.458	0.647	-0.130	0.203
Weight	0.000	0.000	1.556	0.122	0.000	0.001
Spine2	-0.214	0.375	-0.571	0.568	-1.006	0.480
Spine3	-0.049	0.192	-0.257	0.797	-0.416	0.337

c)

```r
se <- tidy(sat_model)$std.error

se_overdis <- tidy(sat_model_2)$std.error

dispersion_param <- (se_overdis/se)^2

dispersion_param
```

```
[1] 3.169448 3.169448 3.169448 3.169448 3.169448
```

The estimated dispersion parameter is 3.169448.

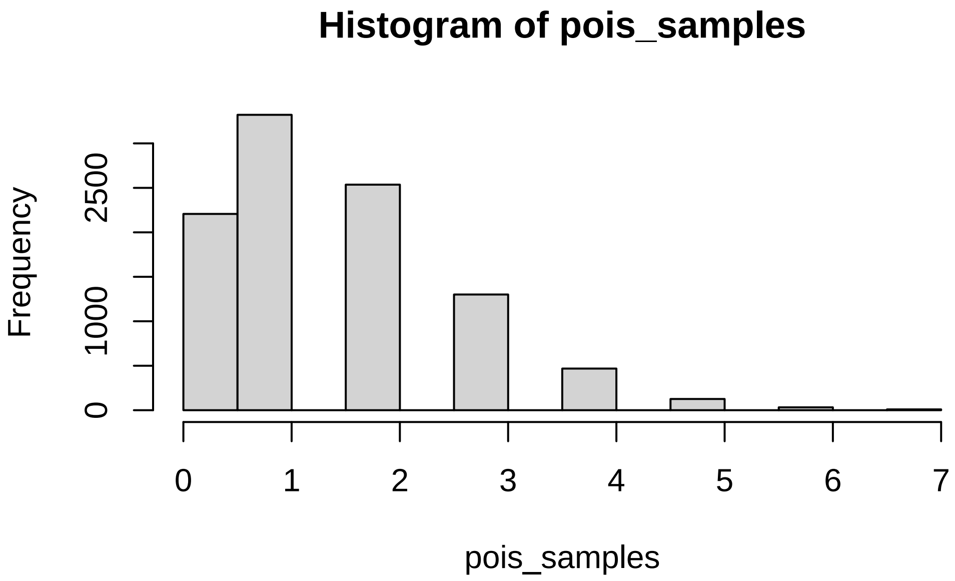
d) The estimated coefficients do not change at all between this model and the previous one while the standard errors increase by a multiplicative factor of 23.7943463

Exercise 4

a)

```r
pois_samples <- rpois(10000, 1.5)

hist(pois_samples)
```



```
mean(pois_samples)
```

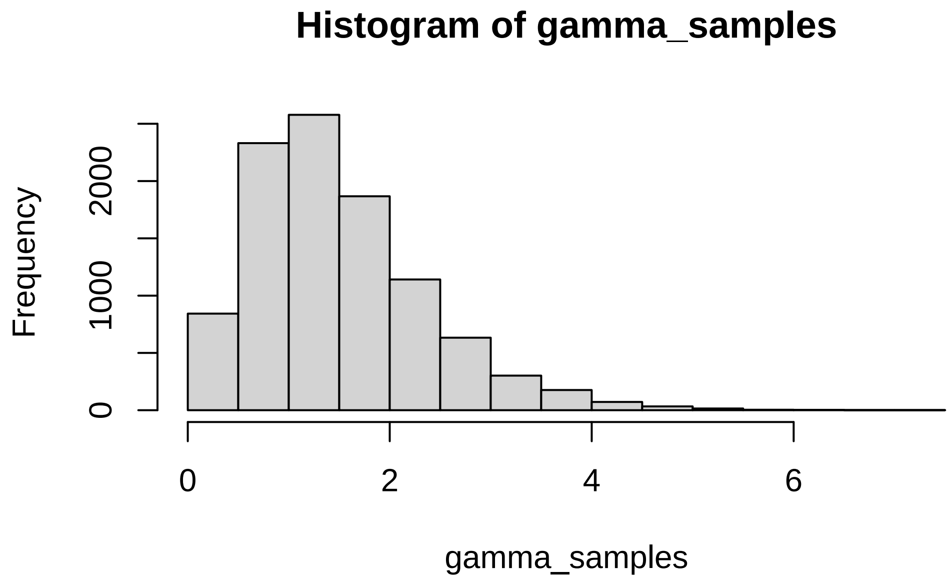
```
[1] 1.5053
```

```
var(pois_samples)
```

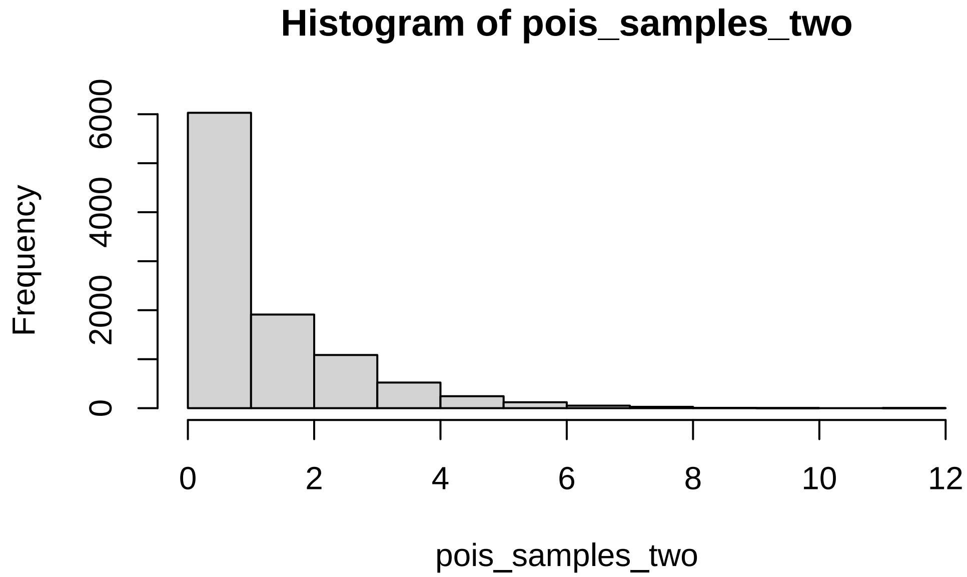
```
[1] 1.474719
```

Our mean and variance are 1.48 and 1.47 respectively, which is closed to our theoretical mean and variance of $\lambda = 1.5$.

```
gamma_samples <- rgamma(10000, 3, 2)
hist(gamma_samples)
```



```
pois_samples_two <- rpois(10000, gamma_samples)
hist(pois_samples_two)
```



```
mean(pois_samples_two)
```

```
[1] 1.4737
```

```
var(pois_samples_two)
```

```
[1] 2.248933
```

The variance is larger here, while the mean is pretty similar, only slightly larger. Our histogram makes our distribution appear less symmetric than it did before.

b) In class, we showed that if we have

$$\lambda \sim Gamma(r, \frac{p}{1-p})$$
$$Y|\lambda \sim Poisson(\lambda)$$

Then, we can say that

$$Y \sim NegBinom(r, p)$$

If we have a

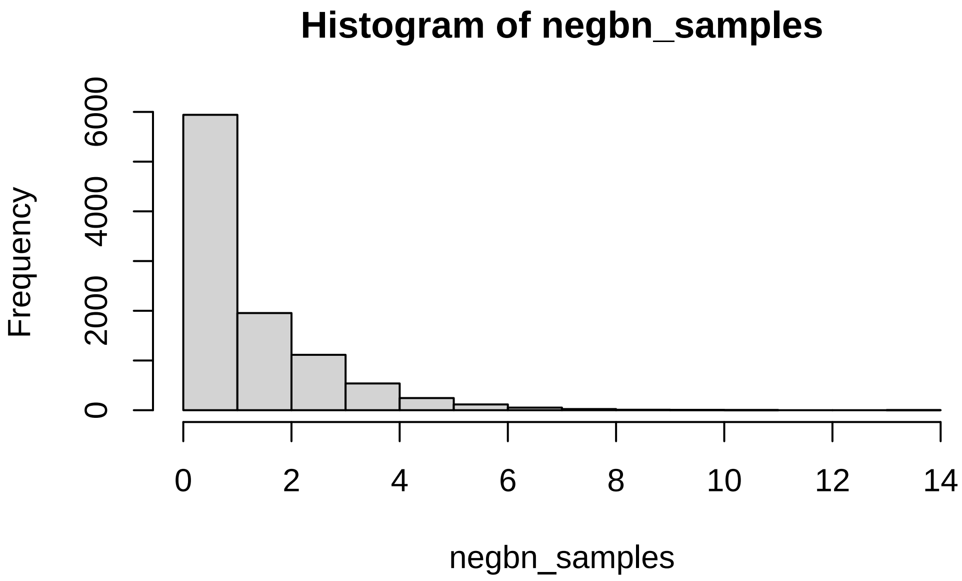
$$\lambda \sim Gamma(3, 2)$$

Then, it's clear that $r = 3$ and $\frac{p}{1-p} = 2$.

Some minor algebra makes it clear that $p = \frac{2}{3}$.

```
negbn_samples <- rnbinom(10000, 3, 2/3)

hist(negbn_samples)
```



```
mean(negbn_samples)
```

```
[1] 1.5081
```

```
var(negbn_samples)
```

```
[1] 2.280962
```

The histogram for the NegBinom(3, 2/3) appears very similar to our Poisson-Gamma distribution. The mean and variance va lues are also very similar.

c)

I will make the mathematical argument that $Y \sim NegBinom(r, p)$ is equivalent to sampling from $Y|\lambda \sim Poisson(\lambda)$ where $\lambda \sim Gamma(r, \frac{p}{1-p})$

This can be shown by deriving the marginal density of Y.

$$f_Y = \int f_{Y,\lambda}\, d\lambda$$

$$= \int f_Y f_\lambda\, d\lambda$$

$$= \int \frac{\lambda^Y e^{-\lambda}}{y!} \cdot \frac{(\frac{p}{1-p})^r}{\Gamma(r)} \cdot \lambda^{r-1} \cdot e^{-(\frac{p}{1-p})\lambda} d\lambda$$

$$= \frac{\left(\frac{p}{1-p}\right)^r}{y!\,\Gamma(r)} \int \lambda^{y+r-1} e^{-\left(1+\frac{p}{1-p}\right)\lambda}\, d\lambda$$

We can utilize the kernel rule to rearrange and find that.

$$= \frac{(y+r-1)!}{y!(r-1)!} p^r (1-p)^y$$

This is equivalent to

$$Y \sim NegBinom(r, p)$$

Exercise 5

a)

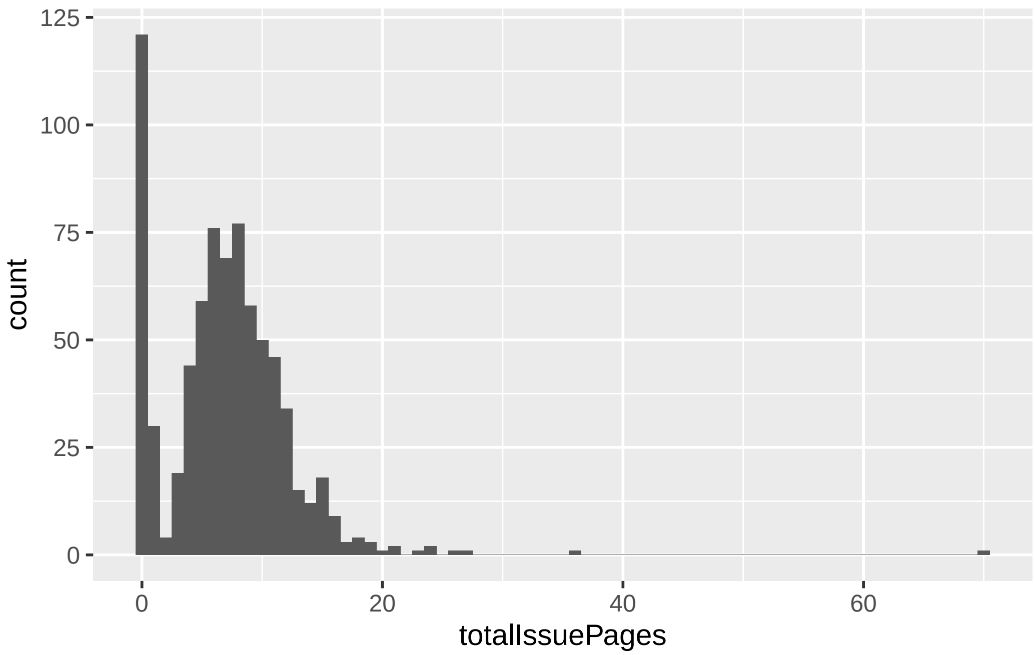
```
ambiguity <- read_csv("data/ambiguity.csv")
```

```
Rows: 870 Columns: 11
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (1): name
dbl (10): ambiguity, distID, ideology, totalIssuePages, democrat, mismatch, ...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
ggplot(ambiguity, aes(x = totalIssuePages)) +
  geom_histogram(binwidth = 1)
```

```
Warning: Removed 109 rows containing non-finite outside the scale range
(`stat_bin()`).
```



We might consider using a hurdle model because the number of zeroes is very large, more so than we would expect in a Poisson model. A zero-inflated model is not appropriate because ZIP assumes that the zeroes are comprised of two different groups– the true always zeoroes and the zeroes on occasion. Here, the context of our observations doesn't indicate that this sort of analysis would make sense. We are counting number of issues candidates commented on, so there are not really two groups of zeroes since the study involves every statement a candidate made, not just on a single occasion.

b)

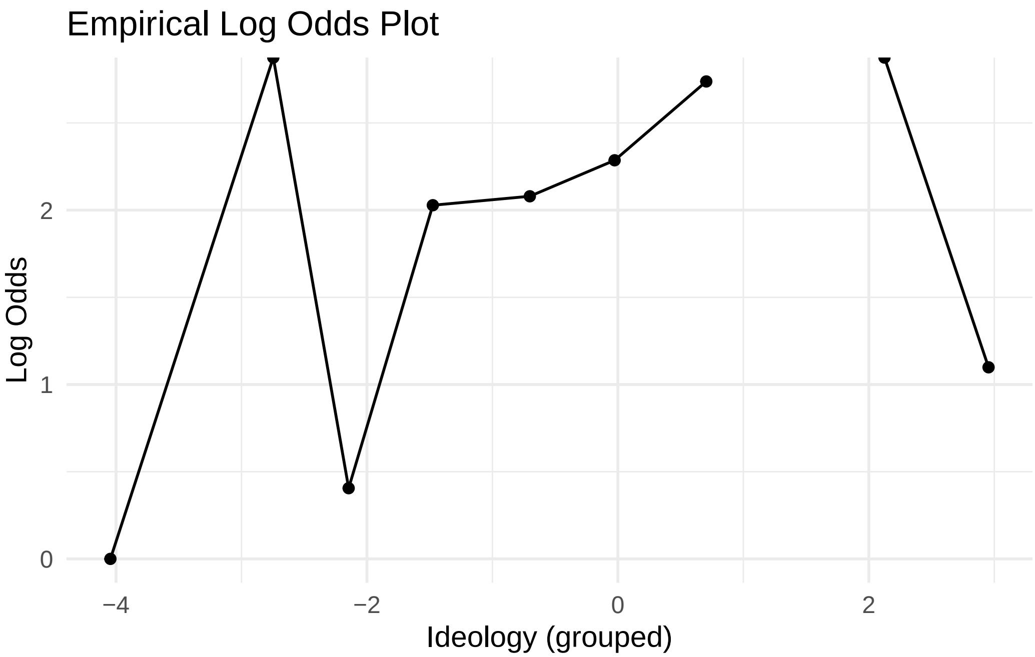
```
ambiguity <- ambiguity |>
  mutate(
    atLeastOne = if_else(totalIssuePages >=1, 1, 0))
```

```
df_binned <- ambiguity %>%
  mutate(bin = cut(ideology, breaks = 10)) %>%
  group_by(bin) %>%
  summarize(mean_ideology = mean(ideology), p_hat = mean(atLeastOne),
            .groups = "drop") %>%
  mutate(log_odds = log(p_hat / (1 - p_hat)))

# Plot
ggplot(df_binned, aes(x = mean_ideology, y = log_odds)) +
  geom_point() +
  geom_line() +
  labs(title = "Empirical Log Odds Plot", x = "Ideology (grouped)", y = "Log Odds") +
  theme_minimal()
```

```
Warning: Removed 2 rows containing missing values or values outside the scale range
(`geom_point()`).
```

```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_line()`).
```



https://chatgpt.com/c/67b67f41-86b4-8011-89ad-ff17749876d0

Used chatgpt to get started on figuring out this empirical logit plot.

c)

```
install.packages("pscl")
```

```
Installing package into '/home/guest/R/x86_64-pc-linux-gnu-library/4.4'
(as 'lib' is unspecified)
```

```r
library(pscl)
```

```
Classes and Methods for R originally developed in the
Political Science Computational Laboratory
Department of Political Science
Stanford University (2002-2015),
by and under the direction of Simon Jackman.
hurdle and zeroinfl functions by Achim Zeileis.
```

```r
hurdle(totalIssuePages ~ ideology + democrat,
       dist = "poisson", data = ambiguity)
```

```
Call:
hurdle(formula = totalIssuePages ~ ideology + democrat, data = ambiguity,
    dist = "poisson")

Count model coefficients (truncated poisson with log link):
(Intercept)     ideology     democrat
   2.093167    -0.005902     0.041379

Zero hurdle model coefficients (binomial with logit link):
(Intercept)     ideology     democrat
     2.1266       0.5746       0.4279
```

Interpret ideology -

For candidates that commented on at least one issue, for each unit increase in ideology, the mean number of issues commented on is expected to change by a multiplicative factor of 0.9941154, holding party status constant.

The odds that a candidate spoke on zero issues changes by a multiplicative factor 1.7764198 for each unit increase in ideology, holding party status constant.

d)

```r
hurdle(totalIssuePages ~ ideology + democrat + ideology*democrat,
       dist = "poisson", data = ambiguity)
```

```
Call:
hurdle(formula = totalIssuePages ~ ideology + democrat + ideology * democrat,
    data = ambiguity, dist = "poisson")

Count model coefficients (truncated poisson with log link):
      (Intercept)           ideology           democrat  ideology:democrat
          2.05817            0.03387            0.05736           -0.07592

Zero hurdle model coefficients (binomial with logit link):
      (Intercept)           ideology           democrat  ideology:democrat
           1.8129             1.3667             0.3581            -1.3995
```

For democrat candidates that commented on at least one issue, the mean number of issues a candidate spoke on is expected to change by a multiplicative factor of 0.9588218 for each unit increase in ideology.

The odds that a candidate spoke on zero issues changes by a multiplicative factor 0.9677321 for each unit increase in ideology, which is 0.2467203 times the multiplicative rate of change we'd expect for a candidate who is not a democrat.