432 Class 16 Slides

thomaselove.github.io/432

2022-03-15

Multinomial Logistic Regression: An Introduction

Setup

```
library(here); library(magrittr); library(janitor)
library(knitr); library(naniar); library(broom)
library(rms)
library(nnet)
library(conflicted)
library(tidyverse)

conflict_prefer("summarize", "dplyr")

theme_set(theme_bw())
```

Regression on Multi-categorical Outcomes

Suppose we have a nominal, multi-categorical outcome of interest. Multinomial (also called multicategory or polychotomous) logistic regression models describe the odds of response in one category instead of another.

- Such models pair each outcome category with a baseline category, the choice of which is arbitrary.
- The model consists of J-1 logit equations (for an outcome with J categories) with separate parameters for each.

Working with `gator1`

The gator1 data: Alligator Food Choice

The gator1 data are from a study by the Florida Game and Fresh Water Fish Commission of factors influencing the primary food choice of alligators[1].

The data include the following data for 59 alligators:

- length (in meters)
- choice = primary food type, in volume, found in the alligator's stomach, specifically...
  - Fish,
  - Invertebrates (mostly apple snails, aquatic insects and crayfish, and I'll abbreviate this category as Inverts in what follows)
  - Other (which includes reptiles, amphibians, mammals, plant material and stones or other debris.)

We'll be trying to predict primary food choice using length.

[1] My Source: Agresti's 1996 first edition of An Introduction to Categorical Data Analysis, Table 8.1. These were provided by Delany MF and Moore CT.

Today's Data

Today's data relates to alligator food choices. We'll actually work with two different data sets.

In each case, we'll read in the data, and set some key variables to be factors and, if needed, actively select the baseline category.

```
gator1 <- read_csv(here("data", "gator1.csv")) %>%
    mutate(choice = fct_relevel(factor(choice), "Other"),
           choice = fct_recode(choice,
                               "Inverts" = "Invertebrates"))
```

Alligator Food Choice, Part 1

```
gator1
```

```
# A tibble: 59 x 3
      id length choice
   <dbl>  <dbl> <fct>
 1     1   1.24 Inverts
 2     2   1.3  Inverts
 3     3   1.3  Inverts
 4     4   1.32 Fish
 5     5   1.32 Fish
 6     6   1.4  Fish
 7     7   1.42 Inverts
 8     8   1.42 Fish
 9     9   1.45 Inverts
10    10   1.45 Other
# ... with 49 more rows
```

Summarizing the gator1 data

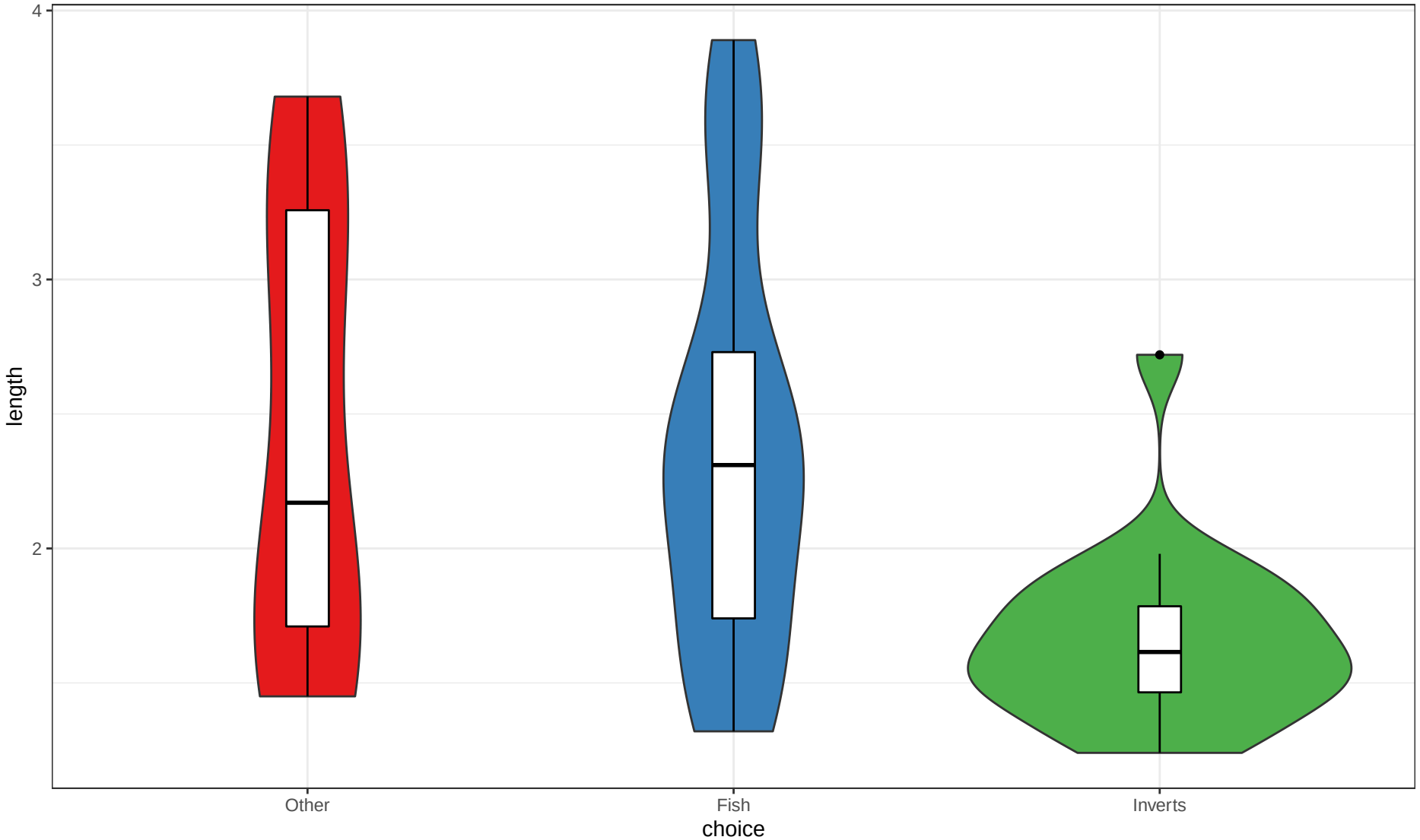
```
mosaic::favstats(length ~ choice, data = gator1) %>%
    kable(digits = 2)
```

choice	min	Q1	median	Q3	max	mean	sd	n	missing
Other	1.45	1.71	2.17	3.26	3.68	2.42	0.88	8	0
Fish	1.32	1.74	2.31	2.73	3.89	2.36	0.76	31	0
Inverts	1.24	1.47	1.61	1.78	2.72	1.66	0.33	20	0

```
n_miss(gator1)
```

```
[1] 0
```

Plotting Length by Primary Food Choice



Plotting Length by Primary Food Choice (code)

```
ggplot(gator1, aes(x = choice, y = length, fill = choice)) +
    geom_violin(trim = TRUE) +
    geom_boxplot(fill = "white", col = "black",
                 width = 0.1) +
    scale_fill_brewer(palette = "Set1") +
    guides(fill = "none")
```

Fitting a Multinomial Logistic Regression

- We'll start by setting "Other" as the first (reference) level for the `choice` outcome

```
gator1 <- gator1 %>%
    mutate(choice = fct_relevel(choice, "Other"))
```

For our first try, we'll use the `multinom` function from the `nnet` package. . .

```
try1 <- multinom(choice ~ length, data=gator1)
```

```
# weights:  9 (4 variable)
initial  value 64.818125
iter  10 value 49.170785
final  value 49.170622
converged
```

Looking over the first try

```
try1
```

```
Call:
multinom(formula = choice ~ length, data = gator1)

Coefficients:
        (Intercept)     length
Fish       1.617952 -0.1101836
Inverts    5.697543 -2.4654695

Residual Deviance: 98.34124
AIC: 106.3412
```

Our R output suggests the following models:

- log odds of Fish rather than Other = 1.62 - 0.110 Length
- log odds of Invertebrates rather than Other = 5.70 - 2.465 Length

Estimating Response Probabilities from our First Try

We can express the multinomial logistic regression model directly in terms of outcome probabilities:

$$\pi_j = \frac{exp(\beta_{0j} + \beta_{1j}x)}{\Sigma_j exp(\beta_{0j} + \beta_{1j}x)}$$

Our models contrast "Fish" and "Invertebrates" to "Other" as the reference category.

- log odds of Fish rather than Other = 1.62 - 0.110 Length
- log odds of Invertebrates rather than Other = 5.70 - 2.465 Length
- For the reference category we use $\beta_{0j} = 0$ and $\beta_{1j} = 0$ so that $exp(\beta_{0j} + \beta_{1j}x) = 1$ for that category.

Estimated Response Probabilities

- log odds of Fish rather than Other = 1.62 - 0.110 Length
- log odds of Invertebrates rather than Other = 5.70 - 2.465 Length

and so our estimates (which will sum to 1) are:

$$Pr(Fish|Length = L) = \frac{exp(1.62 - 0.110L)}{1 + exp(1.62 - 0.110L) + exp(5.70 - 2.465L)}$$

$$Pr(Invert.|Length = L) = \frac{exp(5.70 - 2.465L)}{1 + exp(1.62 - 0.110L) + exp(5.70 - 2.465L)}$$

$$Pr(Other|Length = L) = \frac{1}{1 + exp(1.62 - 0.110L) + exp(5.70 - 2.465L)}$$

Making a Prediction

For an alligator of 3.9 meters, for instance, the estimated probability that primary food choice is "other" equals:

$$\hat{\pi}(Other) = \frac{1}{1 + exp(1.62 - 0.110[3.9]) + exp(5.70 - 2.465[3.9])} = 0.232$$

Storing Predicted Probabilities from `try1`

```
try1_fits <-
    predict(try1, newdata = gator1, type = "probs")

gator1_try1 <- cbind(gator1, try1_fits)

head(gator1_try1, 3)
```

```
  id length  choice      Other      Fish   Inverts
1  1   1.24 Inverts 0.05150117 0.2265417 0.7219571
2  2   1.30 Inverts 0.05727232 0.2502677 0.6924600
3  3   1.30 Inverts 0.05727232 0.2502677 0.6924600
```

Tabulating Response Probabilities

```
gator1_try1 %>% group_by(choice) %>%
    summarise(mean(Other), mean(Fish), mean(Inverts))
```

```
# A tibble: 3 x 4
  choice  `mean(Other)` `mean(Fish)` `mean(Inverts)`
  <fct>           <dbl>        <dbl>           <dbl>
1 Other          0.155        0.580           0.265
2 Fish           0.155        0.590           0.255
3 Inverts        0.0973       0.404           0.499
```

Pivot the Wide data to make it longer

We need to have this data organized differently in order to build the plot I want to build.

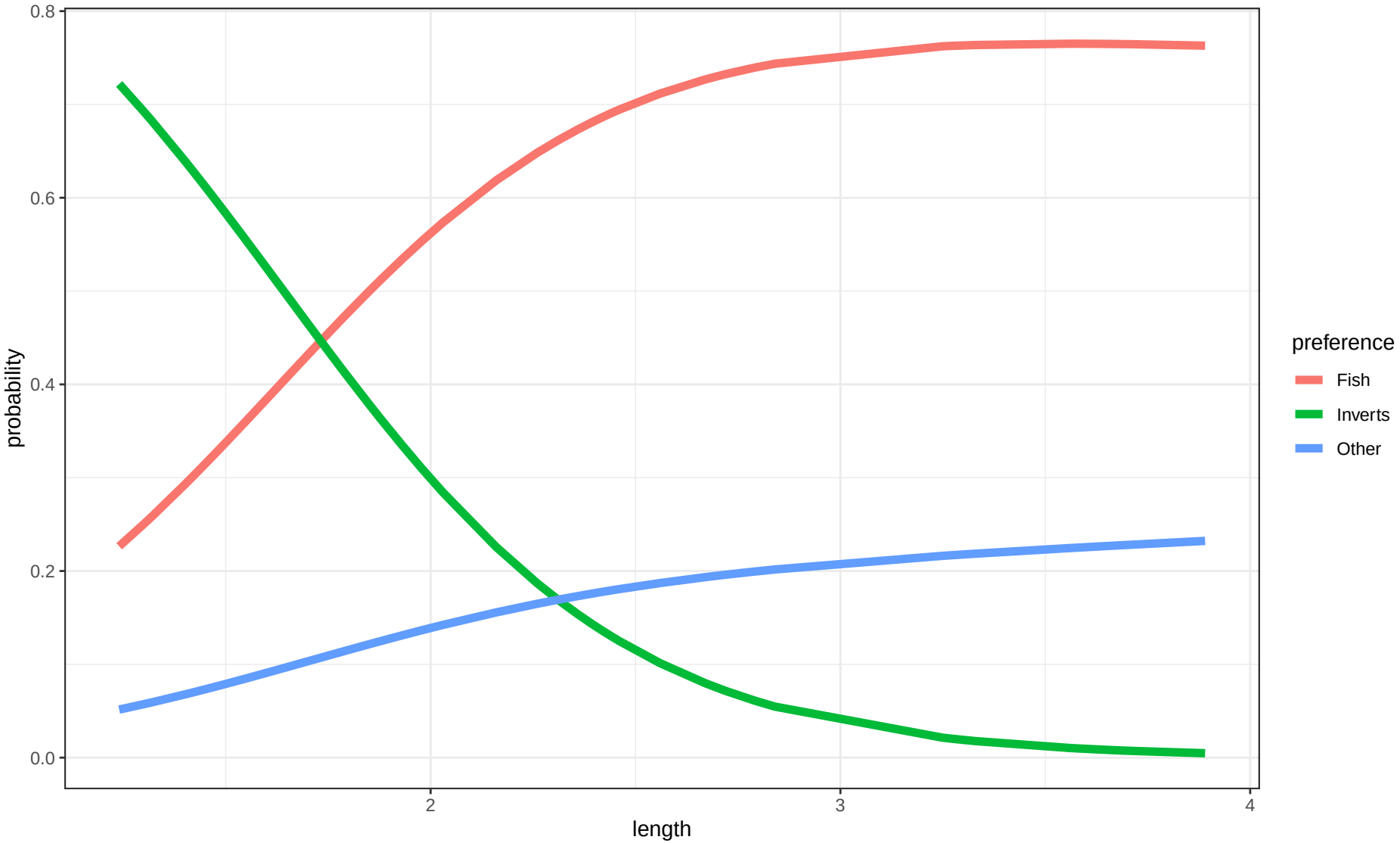
```
gator1_try1long <-
  pivot_longer(gator1_try1,
               cols = c("Other", "Fish", "Inverts"),
               names_to = "preference",
               values_to = "probability") %>%
  mutate(preference = factor(preference))
```

What does this pivoting accomplish?

```
head(gator1_try1long)
```

```
# A tibble: 6 x 5
     id length choice  preference probability
  <dbl>  <dbl> <fct>   <fct>             <dbl>
1     1   1.24 Inverts Other            0.0515
2     1   1.24 Inverts Fish             0.227
3     1   1.24 Inverts Inverts          0.722
4     2   1.3  Inverts Other            0.0573
5     2   1.3  Inverts Fish             0.250
6     2   1.3  Inverts Inverts          0.692
```

Graphing the Model's Response Probabilities



Graphing the Response Probabilities (code)

```
ggplot(gator1_try1long, aes(x = length, y = probability,
                            col = preference)) +
    geom_line(size = 2) +
    scale_fill_brewer(palette = "Set1")
```

summary of try1

```
Call:
multinom(formula = choice ~ length, data = gator1)

Coefficients:
        (Intercept)     length
Fish       1.617952 -0.1101836
Inverts    5.697543 -2.4654695

Std. Errors:
        (Intercept)    length
Fish       1.307291 0.5170838
Inverts    1.793820 0.8996485

Residual Deviance: 98.34124
AIC: 106.3412
```

Assess the try1 model as a whole with a drop in deviance test

Compare the model (try1) to the null model with only an intercept (try0)

```
try0 <- multinom(choice ~ 1, data=gator1)
```

```
# weights:  6 (2 variable)
initial  value 64.818125
final  value 57.570928
converged
```

AIC and BIC to compare try0 to try1

```
AIC(try0, try1)
```

```
     df       AIC
try0  2 119.1419
try1  4 106.3412
```

```
BIC(try0, try1)
```

```
     df       BIC
try0  2 123.2969
try1  4 114.6514
```

Does the inclusion of `length` produce a meaningfully better fit to the data than simply fitting an intercept?

- If you'd prefer a hypothesis testing approach, use `anova`...

ANOVA to compare try0 to try1

```
anova(try0, try1)
```

```
Likelihood ratio tests of Multinomial Models

Response: choice
   Model Resid. df Resid. Dev   Test    Df LR stat.
1      1       116  115.14186                      
2 length       114   98.34124 1 vs 2     2 16.80061
       Pr(Chi)
1             
2 0.0002247985
```

Does the inclusion of `length` produce a meaningfully better fit to the data than simply fitting an intercept?

Wald Z tests for individual predictors

By default, `tidy` exponentiates multinomial coefficients. . .

```
tidy(try1) %>% kable(digits = 3)
```

y.level	term	estimate	std.error	statistic	p.value
Fish	(Intercept)	1.618	1.307	1.238	0.216
Fish	length	-0.110	0.517	-0.213	0.831
Inverts	(Intercept)	5.698	1.794	3.176	0.001
Inverts	length	-2.465	0.900	-2.740	0.006

Working with a larger example: `gator2`

A Larger Alligator Food Choice Example

The `gator2.csv` data[2] considers the stomach contents of 219 alligators, aggregated into 5 categories by primary food choice:

- fish
- invertebrates
- reptiles
- birds
- other (including amphibians, plants, household pets, stones, and debris)

The 219 alligators are also categorized by sex, and by length (< 2.3 and ≥ 2.3 meters) and by which of four lakes they were captured in (Hancock, Oklawaha, Trafford or George.) Table on next slide.

[2] Source: https://onlinecourses.science.psu.edu/stat504/node/226

Lake	Sex	Size	*Primary Food Choice* Fish	Inv.	Rept.	Bird	Other
Hancock	M	small	7	1	0	0	5
		large	4	0	0	1	2
	F	small	16	3	2	2	3
		large	3	0	1	2	3
Oklawaha	M	small	2	2	0	0	1
		large	13	7	6	0	0
	F	small	3	9	1	0	2
		large	0	1	0	1	0
Trafford	M	small	3	7	1	0	1
		large	8	6	6	3	5
	F	small	2	4	1	1	4
		large	0	1	0	0	0
George	M	small	13	10	0	2	2
		large	9	0	0	1	2
	F	small	3	9	1	0	1
		large	8	1	0	0	1

Model Setup

$$\pi_1 = Pr(Fish), \pi_2 = Pr(Invert.), \pi_3 = Pr(Reptiles),$$
$$\pi_4 = Pr(Birds), \pi_5 = Pr(Other)$$

We'll use Fish as the baseline, so our regression equations take the form

$$log(\frac{\pi_j}{\pi_1}) = \beta_0 + \beta_1[Lake = Hancock] + \beta_2[Lake = Oklawaha] +$$
$$\beta_3[Lake = Trafford] + \beta_4[Length \geq 2.3] + \beta_5[Sex = Female]$$

for $j = 2, 3, 4, 5$.

- We have six coefficients to estimate in each of four logit equations (one each for $j = 2, 3, 4, 5$) so there are 24 parameters to estimate.

Loading the gator2 data

```
gator2 <- read_csv(here("data/gator2.csv")) %>%
    type.convert() # converts characters to factors
```

```
Warning in type.convert.default(x[[i]], ...): 'as.is' should
be specified by the caller; using TRUE

Warning in type.convert.default(x[[i]], ...): 'as.is' should
be specified by the caller; using TRUE

Warning in type.convert.default(x[[i]], ...): 'as.is' should
be specified by the caller; using TRUE

Warning in type.convert.default(x[[i]], ...): 'as.is' should
be specified by the caller; using TRUE

Warning in type.convert.default(x[[i]], ...): 'as.is' should
be specified by the caller; using TRUE
```

Rearranging the gator2 data

We rearrange factor levels as needed to get our reference categories to appear first.

```
gator2 <- gator2 %>%
    mutate(food = fct_relevel(food, "fish", "invert",
                              "rep", "bird", "other"),
           size = fct_relevel(size, ">=2.3"),
           gender = fct_relevel(gender, "m"))
```

Now, gator2 matches our order. . .

```
summary(gator2)
```

```
       id              food        size      gender
 Min.   :  1.0   fish  :94   >=2.3: 95   m:130
 1st Qu.: 55.5   invert:61   <2.3 :124   f: 89
 Median :110.0   rep   :19
 Mean   :110.0   bird  :13
 3rd Qu.:164.5   other :32
 Max.   :219.0
     lake
 Length:219
 Class :character
 Mode  :character
```

Complete Set of Models We Will Fit

- Response: Category of Primary Food Choice
- Predictors: L = lake, G = gender, S = size

Specifically, we'll fit (using the `multinom` function in the `nnet` package)

- A *saturated* model, including all three predictors and all two-way interactions and the three-way interaction
- A *null* model, with the intercept alone
- Simple logistic regression models for each of the three predictors as a main effect alone
- The model including both L(ake) and S(ize) but nothing else
- The model including all three predictors as main effects, but no interactions

Our Models (Code)

```
options(contrasts=c("contr.treatment", "contr.poly"))
fit_SAT <- multinom(food ~ lake*size*gender, data=gator2)
        # saturated
fit_1<-multinom(food~1,data=gator2)                          # null
fit_G<-multinom(food~gender,data=gator2)                     # G
fit_L<-multinom(food~lake,data=gator2)                       # L
fit_S<-multinom(food~size,data=gator2)                       # S
fit_LS<-multinom(food~lake+size,data=gator2)                  # L+S
fit_GLS<-multinom(food~gender+lake+size,data=gator2) # G+L+S
```

What You'll See When Fitting the models

```
options(contrasts=c("contr.treatment", "contr.poly"))
fit_SAT <- multinom(food ~ lake*size*gender, data=gator2)
```

```
# weights:  85 (64 variable)
initial  value 352.466903
iter  10 value 261.200857
iter  20 value 245.788420
iter  30 value 244.090612
iter  40 value 243.812122
iter  50 value 243.801212
final  value 243.800899
converged
```

and we'll see something similar for each of the other models. . .

etc.

etc.

Summarizing the Models: Intercept only

```
summary(fit_1)
```

```
Call:
multinom(formula = food ~ 1, data = gator2)

Coefficients:
       (Intercept)
invert  -0.4324211
rep     -1.5988558
bird    -1.9783458
other   -1.0775589

Std. Errors:
       (Intercept)
invert   0.1644133
rep      0.2515350
bird     0.2959078
```

Tidying this summary

```
tidy(fit_1, exponentiate = FALSE) %>% kable(digits = 3)
```

y.level	term	estimate	std.error	statistic	p.value
invert	(Intercept)	-0.432	0.164	-2.630	0.009
rep	(Intercept)	-1.599	0.252	-6.356	0.000
bird	(Intercept)	-1.978	0.296	-6.686	0.000
other	(Intercept)	-1.078	0.205	-5.265	0.000

```
glance(fit_1) %>% kable()
```

edf	deviance	AIC	nobs
4	604.3629	612.3629	219

Summarizing the Models: Size only

```
summary(fit_S)
```

```
Call:
multinom(formula = food ~ size, data = gator2)

Coefficients:
       (Intercept)   size<2.3
invert   -1.034070  0.9489120
rep      -1.241705 -0.8583649
bird     -1.727214 -0.5551882
other    -1.241709  0.2943162

Std. Errors:
       (Intercept)  size<2.3
invert   0.2910708 0.3568648
rep      0.3148729 0.5349960
bird     0.3836949 0.6063277
```

Size only model

```
tidy(fit_S, exponentiate = FALSE) %>% kable(digits = 3)
```

y.level	term	estimate	std.error	statistic	p.value
invert	(Intercept)	-1.034	0.291	-3.553	0.000
invert	size<2.3	0.949	0.357	2.659	0.008
rep	(Intercept)	-1.242	0.315	-3.944	0.000
rep	size<2.3	-0.858	0.535	-1.604	0.109
bird	(Intercept)	-1.727	0.384	-4.502	0.000
bird	size<2.3	-0.555	0.606	-0.916	0.360
other	(Intercept)	-1.242	0.315	-3.944	0.000
other	size<2.3	0.294	0.415	0.709	0.478

```
glance(fit_S) %>% kable()
```

edf	deviance	AIC	nobs
8	589.2134	605.2134	219

Gender only model

```
tidy(fit_G, exponentiate = FALSE) %>% kable(digits = 3)
```

y.level	term	estimate	std.error	statistic	p.value
invert	(Intercept)	-0.581	0.217	-2.673	0.008
invert	genderf	0.358	0.334	1.072	0.284
rep	(Intercept)	-1.513	0.306	-4.937	0.000
rep	genderf	-0.251	0.538	-0.467	0.641
bird	(Intercept)	-2.132	0.400	-5.332	0.000
bird	genderf	0.368	0.596	0.618	0.537
other	(Intercept)	-1.187	0.269	-4.409	0.000
other	genderf	0.271	0.415	0.652	0.514

```
glance(fit_G) %>% kable()
```

edf	deviance	AIC	nobs
8	602.2589	618.2589	219

Lake only model (part 1 of 2)

```
tidy(fit_L, exponentiate = FALSE) %>% slice(1:12) %>% kable(di
```

y.level	term	estimate	std.error	statistic	p.value
invert	(Intercept)	-0.501	0.283	-1.767	0.077
invert	lakehancock	-1.514	0.603	-2.511	0.012
invert	lakeoklawaha	0.555	0.434	1.278	0.201
invert	laketrafford	0.826	0.461	1.791	0.073
rep	(Intercept)	-3.496	1.015	-3.445	0.001
rep	lakehancock	1.194	1.182	1.010	0.312
rep	lakeoklawaha	2.552	1.108	2.302	0.021
rep	laketrafford	3.011	1.110	2.713	0.007
bird	(Intercept)	-2.398	0.603	-3.976	0.000
bird	lakehancock	0.607	0.773	0.785	0.432
bird	lakeoklawaha	-0.492	1.191	-0.413	0.680
bird	laketrafford	1.220	0.831	1.468	0.142

Lake only model (part 2 of 2)

```
tidy(fit_L, exponentiate = FALSE) %>% slice(13:16) %>% kable(d
```

y.level	term	estimate	std.error	statistic	p.value
other	(Intercept)	-1.705	0.444	-3.841	0.000
other	lakehancock	0.869	0.554	1.567	0.117
other	lakeoklawaha	-0.087	0.765	-0.114	0.910
other	laketrafford	1.443	0.611	2.359	0.018

```
glance(fit_L)
```

```
# A tibble: 1 x 4
    edf deviance   AIC  nobs
  <dbl>    <dbl> <dbl> <int>
1    16     561.  593.   219
```

The Saturated Model

We'll show the complete output on the next slide.

```
fit_SAT
```

```
Call:
multinom(formula = food ~ lake * size * gender, data = gator2)

Coefficients:
       (Intercept) lakehancock lakeoklawaha laketrafford
invert  -22.731435  -7.6997047     22.11245    22.443706
rep     -29.030622   4.5446124     28.25748    28.742943
bird     -2.196705   0.8106289    -18.76043     1.215771
other    -1.503884   0.8107459    -25.23128     1.033839
         size<2.3     genderf lakehancock:size<2.3
invert 22.4691578  20.6519880            6.0160287
rep    -2.1497924  -1.5018889          -15.0175978
bird    0.3248760 -17.2683965          -22.8201143
other  -0.3675892  -0.5756885            0.7242536
```

```
> fit_SAT
Call:
multinom(formula = food ~ lake * size * gender, data = gator2)

Coefficients:
       (Intercept) lakehancock lakeoklawaha laketrafford   size<2.3     genderf lakehancock:size<2.3
invert  -22.731435  -7.6997047     22.11245    22.443706 22.4691578  20.6519880            6.0160287
rep     -29.030622   4.5446124     28.25748    28.742943 -2.1497924  -1.5018889          -15.0175978
bird     -2.196705   0.8106289    -18.76043     1.215771  0.3248760 -17.2683965          -22.8201143
other    -1.503884   0.8107459    -25.23128     1.033839 -0.3675892  -0.5756885            0.7242536
       lakeoklawaha:size<2.3 laketrafford:size<2.3 lakehancock:genderf lakeoklawaha:genderf
invert             -21.85028           -21.3342850           -3.946342             4.226498
rep                -17.43950             1.3387310           24.889170           -13.585689
bird               -25.18859           -25.8829682           18.248790            62.485154
other               26.40938            -0.2614093            1.268734            -1.758853
       laketrafford:genderf size<2.3:genderf lakehancock:size<2.3:genderf
invert            25.465169      -19.2913107                    2.857688
rep              -18.078274       31.5836415                  -15.396895
bird              16.978562        0.6638064                   20.157737
other             -7.586589        1.3479978                   -3.378585
       lakeoklawaha:size<2.3:genderf laketrafford:size<2.3:genderf
invert                     -4.488351                    -26.979637
rep                         2.767887                    -11.597631
bird                      -24.265617                     25.472087
other                       1.274620                      8.606604

Residual Deviance: 487.6018
AIC: 615.6018
```

Building a Model Comparison Table

For a model `fitX`, we find the:

- Effective degrees of freedom with `fitX$edf`
- Deviance with `deviance(fitX)` or by listing or summarizing the model
- AIC with `AIC(fitX)` or by listing or summarizing the model

```
fit_SAT$edf
```

```
[1] 64
```

```
deviance(fit_SAT)
```

```
[1] 487.6018
```

```
AIC(fit_SAT)
```

```
[1] 615.6018
```

Label	Model	Effective df	Deviance	AIC
fit_SAT	G*S*L (saturated)	64	487.6	615.6

Results across all of the models we've fit

fit	Model	Effective df	Deviance	AIC
1	Intercept only	4	604.4	612.4
G	Gender only	8	602.3	618.3
S	Size only	8	589.2	605.2
L	Lake only	16	561.2	593.2
LS	Lake and Size	20	540.1	580.1
GLS	G, L, S main effects	24	537.9	585.9
SAT	G*S*L (saturated)	64	487.6	615.6

Which model looks like it fits the data best?

- Here, AIC = Deviance + 2(EDF)

Drop in deviance tests (example 1)

Compare Model `G` to intercept-only

```
anova(fit_G, fit_1)
```

```
Likelihood ratio tests of Multinomial Models

Response: food
   Model Resid. df Resid. Dev   Test    Df LR stat.
1      1       872   604.3629                      
2 gender       868   602.2589 1 vs 2     4 2.104069
    Pr(Chi)
1          
2 0.7166248
```

Drop in deviance tests (example 2)

Compare Model `SAT` to Model `GLS`

```
anova(fit_SAT, fit_GLS)
```

```
Likelihood ratio tests of Multinomial Models

Response: food
                   Model Resid. df Resid. Dev   Test    Df
1 gender + lake + size         852   537.8655
2 lake * size * gender         812   487.6018 1 vs 2    40
  LR stat.    Pr(Chi)
1
2 50.26368 0.1281851
```

Results of testing

fit	Model	edf	Deviance	versus	*p*-value
1	Intercept only	4	604.4	–	–
G	Gender only	8	602.3	1	0.717
S	Size only	8	589.2	1	0.004
L	Lake only	16	561.2	1	0
LS	Lake and Size	20	540.1	L	0
GLS	G, L, S main effects	24	537.9	LS	0.696
SAT	G*S*L (saturated)	64	487.6	GLS	0.128

- Which model looks like it fits the data best?

Results of testing

fit	Model	edf	Deviance	versus	*p*-value
1	Intercept only	4	604.4	–	–
G	Gender only	8	602.3	1	0.717
S	Size only	8	589.2	1	0.004
L	Lake only	16	561.2	1	0
LS	Lake and Size	20	540.1	L	0
GLS	G, L, S main effects	24	537.9	LS	0.696
SAT	G*S*L (saturated)	64	487.6	GLS	0.128

- Which model looks like it fits the data best?
- The best model (of these) is apparently the model which collapses on Gender, and uses only Lake and Size as predictors for Food Choice.

The start of the L+S Model

```
tidy(fit_LS, exponentiate = FALSE) %>%
  slice(1:5) %>% kable(digits = 3)
```

y.level	term	estimate	std.error	statistic	p.value
invert	(Intercept)	-1.549	0.425	-3.645	0.000
invert	lakehancock	-1.658	0.613	-2.706	0.007
invert	lakeoklawaha	0.937	0.472	1.986	0.047
invert	laketrafford	1.122	0.491	2.287	0.022
invert	size<2.3	1.458	0.396	3.683	0.000

- So, for instance, log odds of invertebrates rather than fish are:

```
-1.54 + 1.46 Small - 1.66 Hancock
      + 0.94 Oklawaha + 1.12 Trafford
```

etc. For the baseline category, log odds of fish = 0, so exp(log odds) = 1.

Response Probabilities in the L+S Model

To keep things relatively simple, we'll look at the class of Large size alligators (so the small size indicator is 0, in Lake George, so the three Lake indicators are all 0, also).

- The estimated probability of Fish in Large size alligators in Lake George according to our model is:

$$\hat{\pi}(Fish) = \frac{1}{1 + exp(-1.54) + exp(-3.31) + exp(-2.09) + exp(-1.90)}$$

$$= \frac{1}{1.524} = 0.66$$

Response Probabilities in the L+S Model

- The estimated probability of Invertebrates in Large size alligators in Lake George according to our model is:

$$\hat{\pi}(Inv.) = \frac{exp(-1.54)}{1 + exp(-1.54) + exp(-3.31) + exp(-2.09) + exp(-1.90)}$$

$$= \frac{0.214}{1.524} = 0.14$$

The estimated probabilities for the other categories in Large size Lake George alligators are:

- 0.02 for Reptiles, 0.08 for Birds, and 0.10 for Other
- And the five probabilities will sum to 1, at least within rounding error.

Comparing Model Estimates to Observed Counts

For large size alligators in Lake George, we have. . .

Food Type	Fish	Inverts	Reptiles	Birds	Other
Observed #	17	1	0	1	3
Observed Prob.	0.77	0.045	0	0.045	0.14
L+S Model Prob.	0.66	0.14	0.02	0.08	0.10

We could perform similar calculations for all other combinations of size and lake, but I'll leave that to the dedicated.

Storing Predicted Probabilities from `fit_LS`

```
fitLS_fits <-
    predict(fit_LS, newdata = gator2, type = "probs")

gator2_fit_LS <- cbind(gator2, fitLS_fits)

head(gator2_fit_LS, 3)
```

```
  id food size gender    lake      fish     invert
1  1 fish <2.3      m hancock 0.5352844 0.09311221
2  2 fish <2.3      m hancock 0.5352844 0.09311221
3  3 fish <2.3      m hancock 0.5352844 0.09311221
         rep       bird     other
1 0.04745855 0.07040277 0.2537421
2 0.04745855 0.07040277 0.2537421
3 0.04745855 0.07040277 0.2537421
```

Tabulating Response Probabilities

```
gator2_fit_LS %>% group_by(food) %>%
    summarize(mean(fish), mean(invert), mean(rep),
              mean(bird), mean(other))
```

```
# A tibble: 5 x 6
  food  `mean(fish)` `mean(invert)` `mean(rep)` `mean(bird)`
  <fct>        <dbl>          <dbl>       <dbl>        <dbl>
1 fish         0.481          0.230      0.0763       0.0631
2 inve~        0.361          0.393      0.0858       0.0395
3 rep          0.381          0.258      0.148        0.0641
4 bird         0.452          0.197      0.0960       0.0841
5 other        0.426          0.246      0.0791       0.0733
# ... with 1 more variable: `mean(other)` <dbl>
```

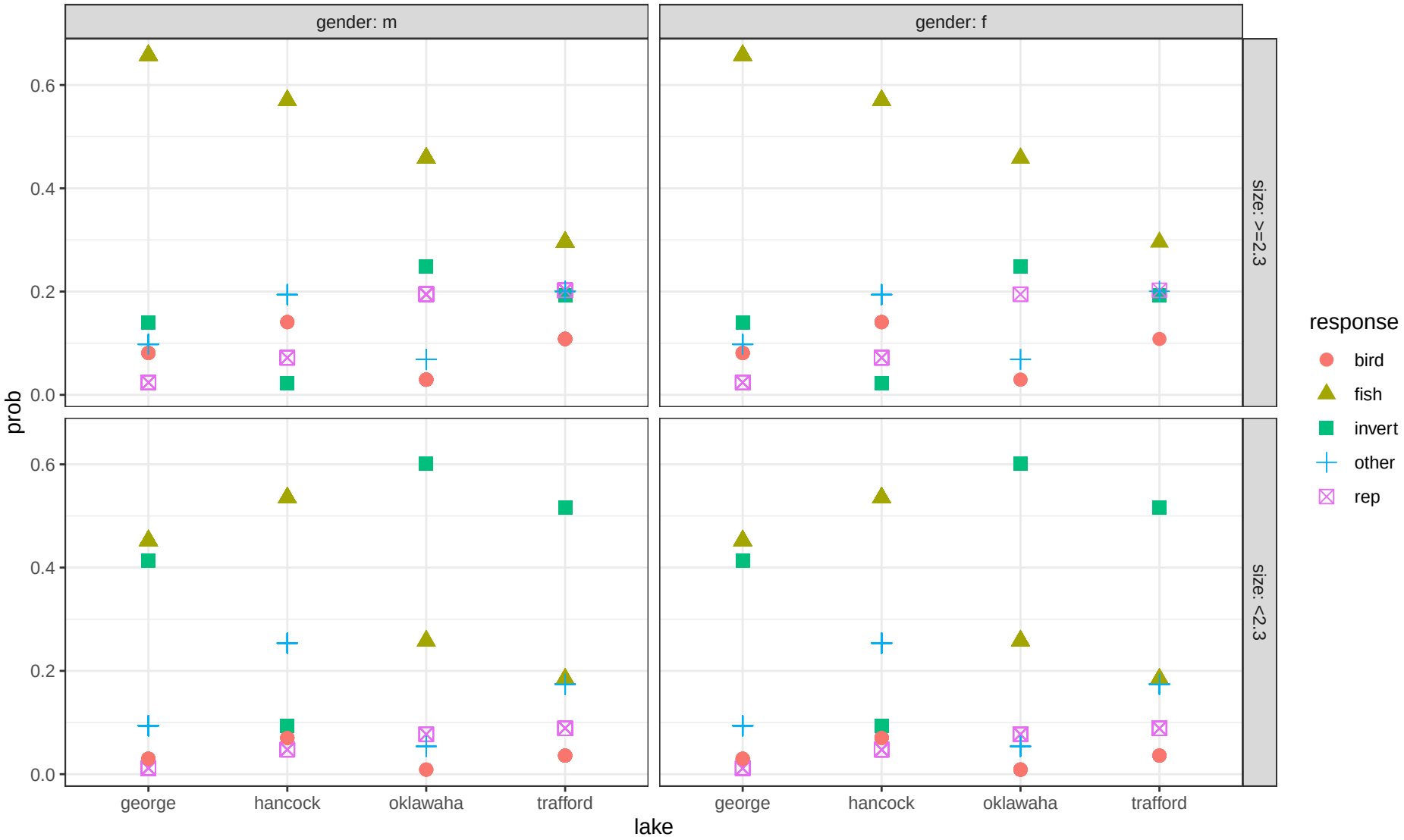
Turn Wide Data into Long

```
gator2_fitLSlong <-
  pivot_longer(gator2_fit_LS,
               cols = fish:other,
               names_to = "response",
               values_to = "prob")

head(gator2_fitLSlong,3)
```

```
# A tibble: 3 x 7
     id food  size  gender lake    response   prob
  <int> <fct> <fct> <fct>  <chr>   <chr>     <dbl>
1     1 fish  <2.3  m      hancock fish      0.535
2     1 fish  <2.3  m      hancock invert    0.0931
3     1 fish  <2.3  m      hancock rep       0.0475
```

Graphing the Model's Response Probabilities



Graphing the Model's Response Probabilities (code)

```
ggplot(gator2_fitLSlong, aes(x = lake, y = prob,
                             col = response,
                             shape = response)) +
    geom_point(size = 3) +
    scale_fill_brewer(palette = "Set1") +
    facet_grid(size ~ gender, labeller = "label_both")
```

Some Sources for Multinomial Logistic Regression

In addition to the example found in our Course Notes. . .

- A good source of information on fitting these models is https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/
- Using the tidymodels structure to fit these models is another good idea. Julia Silge has a very nice example at https://juliasilge.com/blog/multinomial-volcano-eruptions/
- More mathematically oriented sources include the following texts:
  - Hosmer DW Lemeshow S Sturdivant RX (2013) Applied Logistic Regression, 3rd Edition, Wiley
  - Agresti A (2007) An Introduction to Categorical Data Analysis, 2nd Edition, Wiley.

Next Time

Time-to-event data, survival analysis.