Week Twelve

Last Week

- Lab: Count Regression for Bike Data
- Lecture: Multicategory Regression Models (Theory)

This Week: Multicategory Regression Models

- Today: Activity
- Thursday: Lab

Next Week: Generalized Linear Mixed Models

- Tuesday: No Class - Veteran's Day
- Thursday: Activity / Lecture

Multicategory Logit Models

Recall Logistic regression for binary data

$$Y \sim Multinomial(n, \pi)$$

$$\log\left(\frac{\pi_i}{\pi_J}\right) = \alpha_i + \beta_j x$$

- We can also directly estimate $\pi_j(x)$ for any set of covariates.

$$\pi_j = \frac{\exp(\alpha_j + \beta_j x)}{\sum_h \exp(\alpha_h + \beta_h x)},$$

where α_h and $\beta_h = 0$ for the reference category.

Data Analysis

The data set contains variables on 200 high school senior students. This dataset was collected by the National Opinion Research Center with funding from the National Center for Education Statistics.

We will treat `prog` as the outcome variable, where academic is a college preparatory program, general is a basic high school program, and vocation/technical focus on vocational paths.

```
library(foreign)
hsb <- read.dta("https://stats.idre.ucla.edu/stat/data/hsbdemo.dta")
```

1. Let's initially consider the relationship between program type and social economic status. Create a contingency table, run a χ^2 test, and summarize the results.

```
hsb_tab <- table(hsb$prog, hsb$ses)
hsb_tab
```

```
           low middle high
  general   16     20    9
  academic  19     44   42
  vocation  12     31    7
```

```
hsb_chi <- chisq.test(hsb_tab)
hsb_chi
```

```
    Pearson's Chi-squared test

data:  hsb_tab
X-squared = 16.604, df = 4, p-value = 0.002307
```

```
hsb_chi$observed
```

```
          low middle high
  general   16     20    9
  academic  19     44   42
  vocation  12     31    7
```

```
hsb_chi$expected
```

```
             low middle  high
  general  10.575 21.375 13.05
  academic 24.675 49.875 30.45
  vocation 11.750 23.750 14.50
```

This test concludes that it is unlikely that program a student enters and their socioeconomic status are independent. Looking at the observed and expected (under the null hypothesis of independence) that the high ses class tends to see more students in academic programs; whereas the low and middle ses classes tend to see fewer students than expected in academic programs.

2. Now let's consider a multicategory GLM for program, using SES. For a ML approach you can use `nnet::multinom`. You can also use `brms::brm` for a Bayesian approach, but it might require compiling a stan program. Fit the model, interpret the coefficients, and summarize the results graphically.

```
library(nnet)
library(reshape2)
```

```
Attaching package: 'reshape2'
```

```
The following object is masked from 'package:tidyr':

    smiths
```

```r
multi_glm <- multinom(prog ~ ses -1 , data = hsb)
```

```
# weights:  12 (6 variable)
initial  value 219.722458
iter  10 value 195.705190
iter  10 value 195.705188
iter  10 value 195.705188
final  value 195.705188
converged
```

```r
summary(multi_glm)
```

```
Call:
multinom(formula = prog ~ ses - 1, data = hsb)

Coefficients:
             seslow sesmiddle    seshigh
academic  0.1718634 0.7884384  1.5404455
vocation -0.2876754 0.4382459 -0.2512923

Std. Errors:
            seslow sesmiddle   seshigh
academic 0.3393106 0.2696792 0.3673160
vocation 0.3818819 0.2868055 0.5039502

Residual Deviance: 391.4104
AIC: 403.4104
```

```r
library(brms)
```

```
Loading required package: Rcpp

Loading 'brms' package (version 2.23.0). Useful instructions
can be found by typing help('brms'). A more detailed introduction
to the package is available through vignette('brms_overview').
```

```
Attaching package: 'brms'

The following object is masked from 'package:stats':

    ar
```

```
multi_bayes <- brm(
  prog ~ ses - 1,
  data = hsb,
  family = categorical(link = "logit"),
  refresh = 0
)
```

```
Compiling Stan program...

Start sampling
```

```
multi_bayes
```

```
 Family: categorical
  Links: muacademic = logit; muvocation = logit
Formula: prog ~ ses - 1
   Data: hsb (Number of observations: 200)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Regression Coefficients:
                    Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
muacademic_seslow       0.18      0.35    -0.50     0.86 1.00     3716
muacademic_sesmiddle    0.81      0.27     0.29     1.33 1.00     3296
muacademic_seshigh      1.58      0.38     0.89     2.36 1.00     3665
muvocation_seslow      -0.29      0.39    -1.07     0.47 1.00     3531
muvocation_sesmiddle    0.45      0.28    -0.10     1.00 1.00     3288
muvocation_seshigh     -0.28      0.52    -1.34     0.71 1.00     3741
                    Tail_ESS
muacademic_seslow       3289
muacademic_sesmiddle    3301
muacademic_seshigh      3221
muvocation_seslow       3073
```

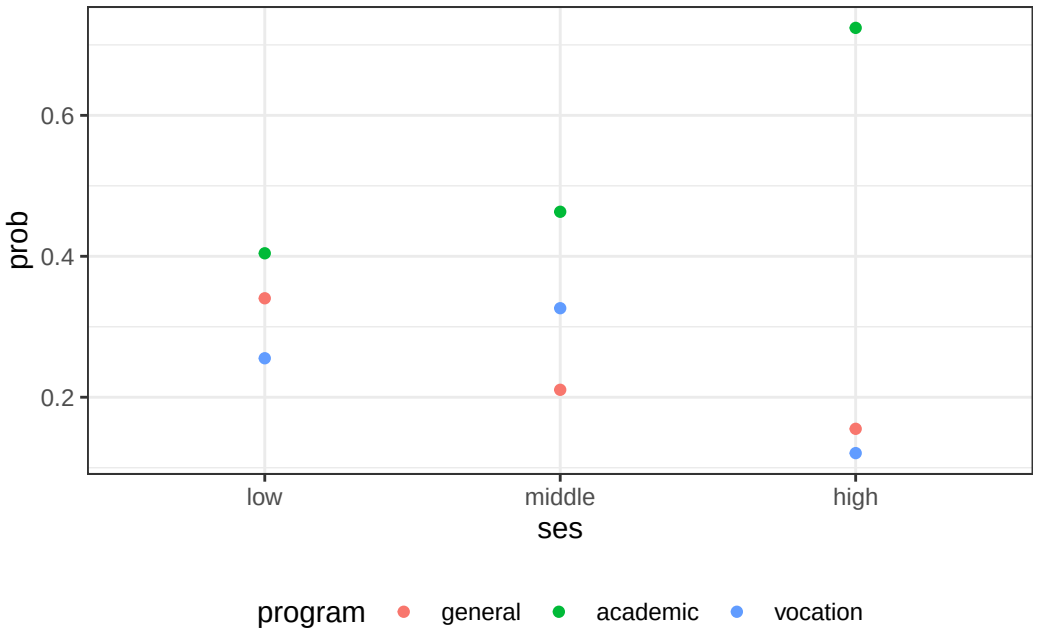
```
muvocation_sesmiddle     2803
muvocation_seshigh       3194

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

The values here are log-odds with respect to the reference case(general), but we can also look at odds ratios (relative risk)

```
exp(coef(multi_glm))
```

```
           seslow sesmiddle  seshigh
academic 1.187516  2.199958 4.666669
vocation 0.750005  1.549986 0.777795
```

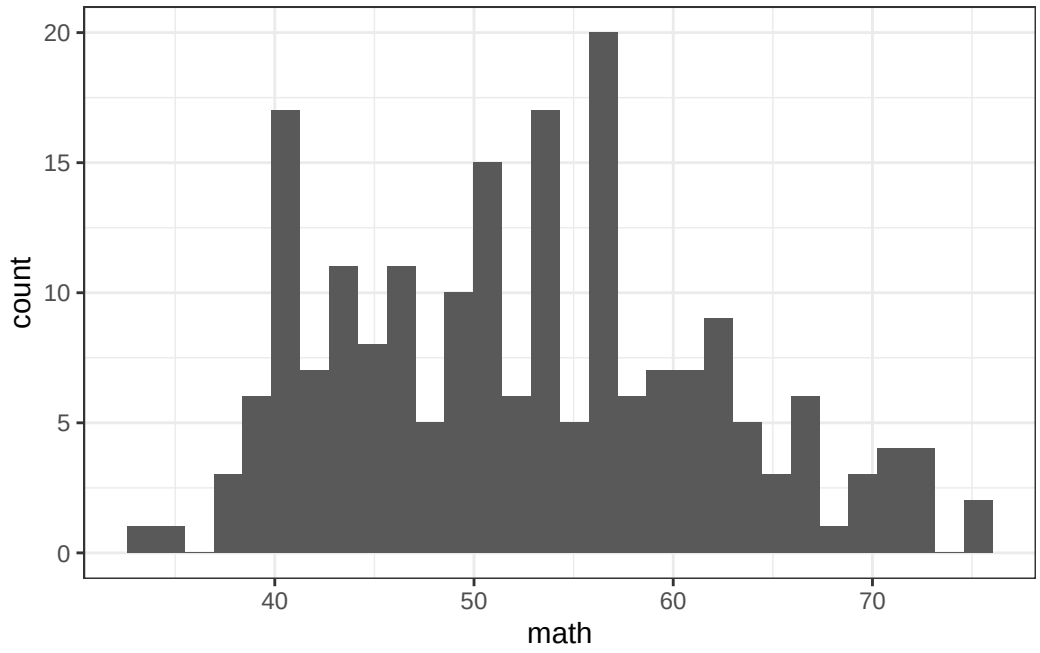
```
ses_levels <- tibble(ses = c("low", "middle", "high"))

ses_probs <- predict(multi_glm, newdata = ses_levels, type = "probs", se = TRUE)

melt(ses_probs) |>
  mutate(Var1 = case_when(
    Var1 == 1 ~ 'low',
    Var1 == 2 ~ 'middle',
    Var1 == 3 ~ 'high'
  ),
  ses = ordered(Var1, levels = c('low','middle','high')),
  program = Var2) |>
  ggplot( aes(y = value, x = ses, color = program )) + geom_point() +
  ylab('prob') +
  theme_bw() +
  theme(legend.position = 'bottom')
```



3. Now let's consider a multicategory GLM for program, using a continuous variable: math. Fit the model, interpret the coefficients, and summarize the results graphically.

```
hsb |>
  ggplot(aes(x = math)) +
  geom_histogram() +
  theme_bw()
```

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




```
hsb <- hsb |>
  mutate(math_centered = math - mean(math))
```

```
multi_glm <- multinom(prog ~ math_centered , data = hsb)
```

```
# weights:  9 (4 variable)
initial  value 219.722458
iter  10 value 178.114228
iter  10 value 178.114228
final  value 178.114228
converged
```

```
summary(multi_glm)
```

```
Call:
multinom(formula = prog ~ math_centered, data = hsb)

Coefficients:
          (Intercept) math_centered
academic    0.7884729    0.09202162
vocation   -0.1784970   -0.06295409
```

```
Std. Errors:
          (Intercept) math_centered
academic    0.1902947    0.02313612
vocation    0.2471549    0.02800092

Residual Deviance: 356.2285
AIC: 364.2285
```

```r
multi_bayes <- brm(
  prog ~ math_centered,
  data = hsb,
  family = categorical(link = "logit"),
  refresh = 0
)
```

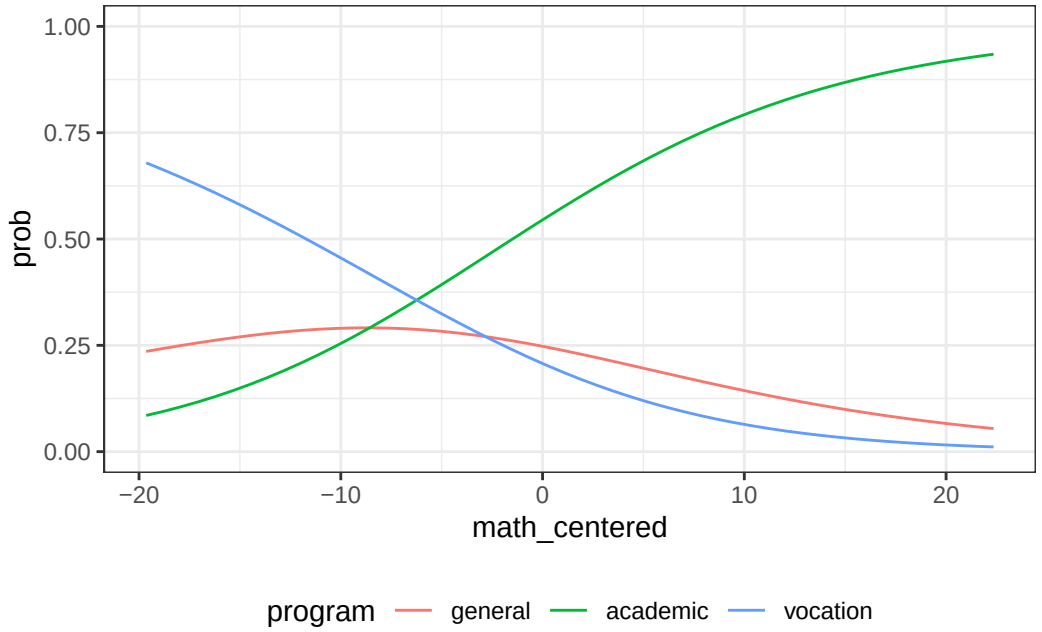
```
Compiling Stan program...

Start sampling
```

```r
multi_bayes
```

```
 Family: categorical
  Links: muacademic = logit; muvocation = logit
Formula: prog ~ math_centered
   Data: hsb (Number of observations: 200)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Regression Coefficients:
                         Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS
muacademic_Intercept         0.79      0.19     0.42     1.17 1.00     2930
muvocation_Intercept        -0.19      0.25    -0.70     0.28 1.01     2613
muacademic_math_centered     0.09      0.02     0.05     0.14 1.00     3115
muvocation_math_centered    -0.06      0.03    -0.12    -0.01 1.00     2711
                         Tail_ESS
muacademic_Intercept         2445
muvocation_Intercept         2755
muacademic_math_centered     2922
muvocation_math_centered     2675
```

```
Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

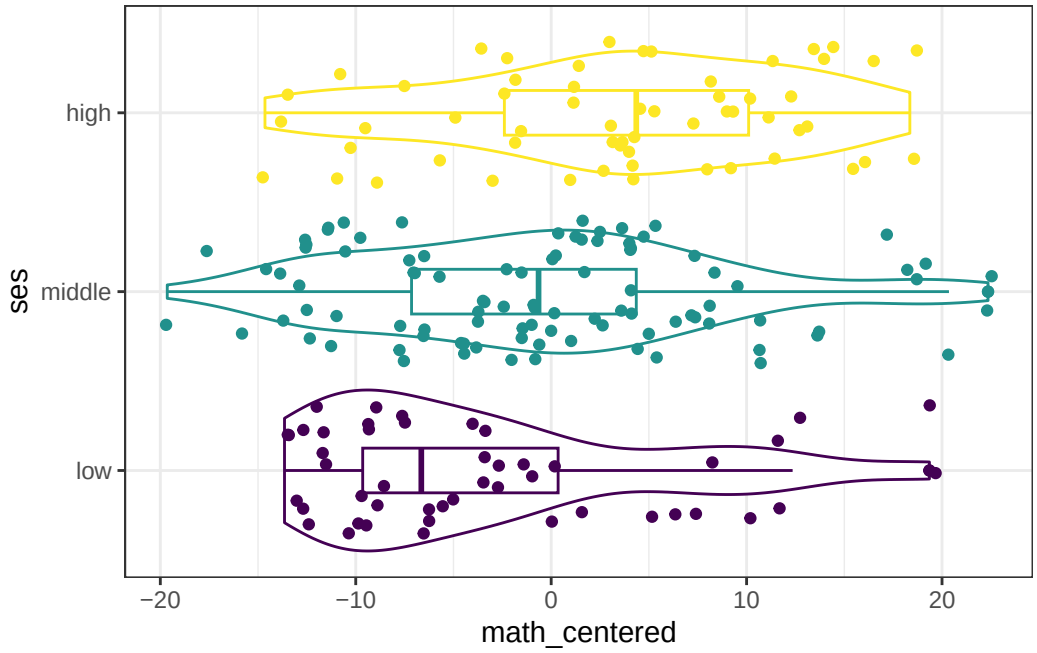
```r
math_levels <- tibble(math_centered = seq(min(hsb$math_centered), max(hsb$math_centered), le

math_probs <- predict(multi_glm, newdata = math_levels, type = "probs", se = TRUE)

melt(math_probs) |>
  bind_cols(math_centered = rep(math_levels$math_centered, 3)) |>
  mutate(ses = ordered(Var1, levels = c('low','middle','high')),
  program = Var2) |>
  ggplot( aes(y = value, x = math_centered, color = program )) + geom_line() +
  ylab('prob') +
  theme_bw() +
  theme(legend.position = 'bottom') +
  ylim(0,1)
```



Ordinal Models

We can also use the `hsb` dataset to fit ordinal regression models

```r
hsb <- hsb |>
  mutate(ses = ordered(ses, levels = c('low','middle','high')))

hsb |>
  ggplot(aes(x = math_centered, y =ses, color = ses)) +
  geom_violin() +
  geom_boxplot(width = .25) +
  geom_jitter() +
  theme_bw() +
  theme(legend.position ='none')
```




```r
library(MASS)
```

```
Attaching package: 'MASS'

The following object is masked from 'package:dplyr':

    select
```

```r
library(rstanarm)
```

```
This is rstanarm version 2.32.1

- See https://mc-stan.org/rstanarm/articles/priors for changes to default priors!

- Default priors may change, so it's safest to specify priors, even if equivalent to the defa

- For execution on a local, multicore CPU with excess RAM we recommend calling

  options(mc.cores = parallel::detectCores())


Attaching package: 'rstanarm'

The following objects are masked from 'package:brms':

    dirichlet, exponential, get_y, lasso, ngrps
```

```r
ordinal_glm <- polr(ses ~ math_centered, data = hsb, method = 'logistic')
ordinal_glm
```

```
Call:
polr(formula = ses ~ math_centered, data = hsb, method = "logistic")

Coefficients:
math_centered
   0.05736345

Intercepts:
 low|middle middle|high
  -1.259069    0.949781

Residual Deviance: 405.8144
AIC: 411.8144
```

```r
ordinal_bayes <- brm(
  ses ~ math_centered,
  data = hsb,
  family = cumulative(link = "logit"),
  refresh = 0
)
```

```
Compiling Stan program...
```

```
Start sampling
```

```r
print(ordinal_bayes)
```

```
 Family: cumulative
  Links: mu = logit
Formula: ses ~ math_centered
   Data: hsb (Number of observations: 200)
  Draws: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup draws = 4000

Regression Coefficients:
              Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept[1]     -1.26      0.17    -1.62    -0.93 1.00     2089     2374
Intercept[2]      0.95      0.16     0.64     1.28 1.00     4945     2971
math_centered     0.06      0.01     0.03     0.09 1.00     2597     2779

Further Distributional Parameters:
     Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
disc     1.00      0.00     1.00     1.00   NA       NA       NA

Draws were sampled using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Recall, we are modeling

$$logit[P(Y \leq J)] = \alpha_j + \beta x$$

- `invlogit(-1.25)` $= 0.2227001$ corresponds to the probability of the low class (at math_score $= 0$)

- `invlogit(.94)` = 0.7190997 corresponds to the probability of the low or middle class (at math_score = 0)

```
math_levels <- tibble(math_centered = seq(min(hsb$math_centered), max(hsb$math_centered), le

math_probs <- predict(ordinal_glm, newdata = math_levels, type = "probs", se = TRUE)

melt(math_probs) |>
  bind_cols(math_centered = rep(math_levels$math_centered, 3)) |>
  mutate(ses = ordered(Var2, levels = c('low','middle','high'))) |>
  ggplot( aes(y = value, x = math_centered, color = ses )) + geom_line() +
  ylab('prob') +
  theme_bw() +
  theme(legend.position = 'bottom') +
  ylim(0,1)
```