Categorical data and regression

Frank Edwards

Categorical data

Categorical data falls into a fixed set of categories. It may be *unordered*, meaning that there is no inherent ranking of categories, or it may be *ordered*. Ordered categorical data has an explicit hierarchical ranking of values.

Categorical data, examples

Are these variables ordered or unordered?

Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election

Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election
- Zip code for people choosing a place to move

Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election
- Zip code for people choosing a place to move
- Cause of death

Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election
- Zip code for people choosing a place to move
- Cause of death
- Opinions on a political issue on a thermometer / Likert scale (e.g. Strongly oppose, oppose, neutral, support, strongly support)

Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election
- Zip code for people choosing a place to move
- Cause of death
- Opinions on a political issue on a thermometer / Likert scale (e.g. Strongly oppose, oppose, neutral, support, strongly support)
- Ranking of academic progrms

Categorical data

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

```
##    id female    ses schtyp     prog read write math science socst       honors
## 1  45 female    low public vocation   34    35   41      29    26 not enrolled
## 2 108   male middle public  general   34    33   41      36    36 not enrolled
## 3  15   male   high public vocation   39    39   44      26    42 not enrolled
## 4  67   male    low public vocation   37    37   42      33    32 not enrolled
## 5 153   male middle public vocation   39    31   40      39    51 not enrolled
## 6  51 female   high public  general   42    36   42      31    39 not enrolled
##   awards cid
## 1      0   1
## 2      0   1
## 3      0   1
## 4      0   1
## 5      0   1
## 6      0   1
```

Crosstabs are often the best

```r
table(dat$prog)
```

```
##
##  general academic vocation
##       45      105       50
```

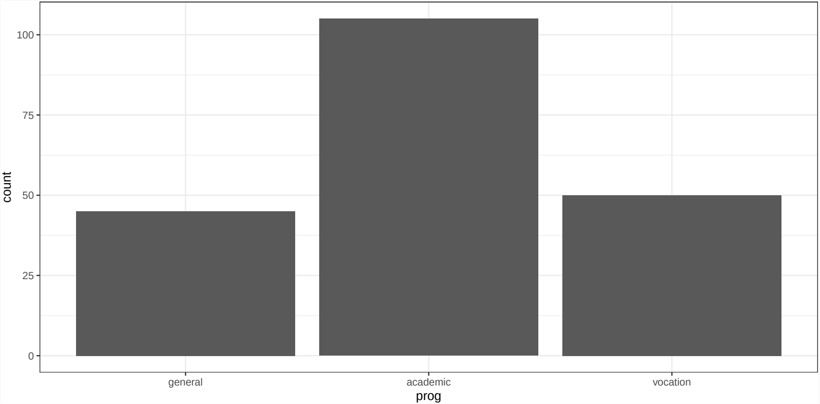
Visualzing categorical data (cont.)

```r
dat %>%
    group_by(prog, ses) %>%
    summarize(n = n()) %>%
    mutate(prop = n/sum(n))
```

```
## # A tibble: 9 x 4
## # Groups:   prog [3]
##   prog     ses        n  prop
##   <fct>    <fct>  <int> <dbl>
## 1 general  low       16 0.356
## 2 general  middle    20 0.444
## 3 general  high       9 0.2
## 4 academic low       19 0.181
## 5 academic middle    44 0.419
## 6 academic high      42 0.4
## 7 vocation low       12 0.24
## 8 vocation middle    31 0.62
## 9 vocation high       7 0.14
```

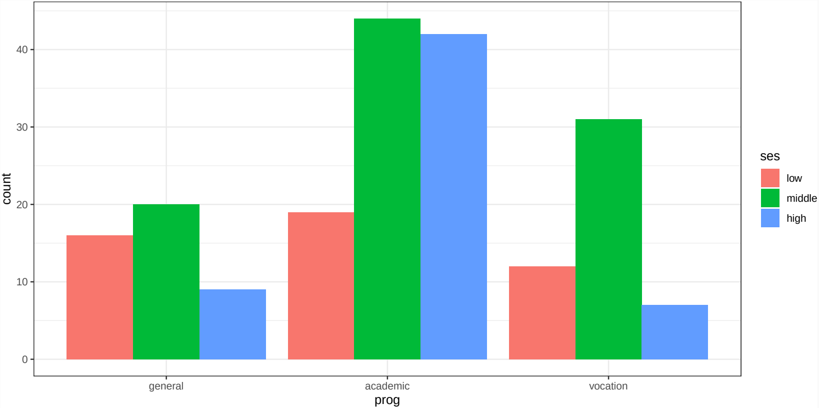
Visualizing categorical data - frequency barplots

```
ggplot(dat, aes(x = prog)) + geom_bar()
```



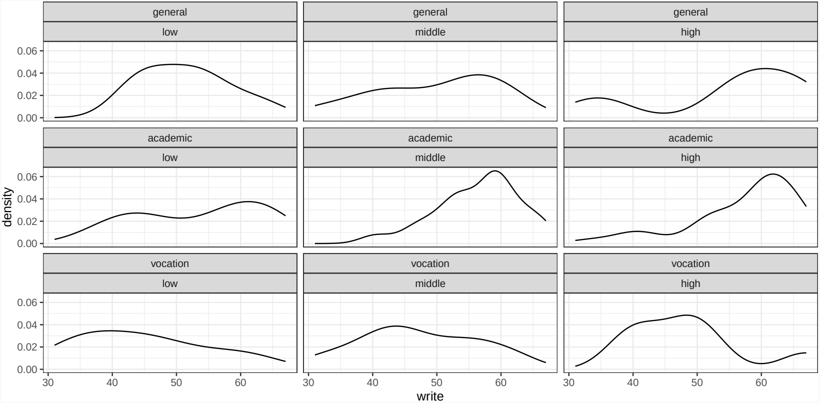
Visualizing categorical data - frequency barplots

```r
ggplot(dat, aes(x = prog, fill = ses)) + geom_bar(position = position_dodge())
```



Visualizing categorical data, facets

```
ggplot(dat, aes(x = write)) + geom_density() + facet_wrap(prog ~ ses)
```



Multinomial logistic regression

Multinomial logistic regression is a GLM that models the log odds of a categorical outcome as a function of a linear combination of a set of predictors.

Multinomial logistic regression is a GLM that models the log odds of a categorical outcome as a function of a linear combination of a set of predictors.

Multinomial logistic regression: basics

For a categorical outcome with K categories, estimate $K-1$ models where 1,2,3 stand in for membership in group 1, 2, 3, ... K:

$$log \frac{Pr(y_i = 1)}{Pr(y_i = K)} = \beta_{k=1} X_i$$

$$log \frac{Pr(y_i = 2)}{Pr(y_i = K)} = \beta_{k=2} X_i$$

$$\ldots$$

$$log \frac{Pr(y_i = K-1)}{Pr(y_i = K)} = \beta_{k=3} X_i$$

Key assumtion: Independence of irrelevant alternatives. Odds of choice do not depend on the presence or absence of other alternatives (i.e. car vs bus or car vs red bus vs blue bus)

Implementation

1. Choose a reference category. This is arbitrary, but changes the interpretation. Remember that we're modeling the log odds of membership in one group relative to another.
2. Estimate a model
3. Interpret results

Implementation

Multinomial logistic regression is easy to estimate using `brms`, an package for estimating Bayesian models using Stan, very similar to `rstanarm`. Simply use `family = categorical` with a call to `brm`.

Estimation

Let's predict high school program choice as a function of socio-economic status and math standardized test score

```
library(brms)
m0 <- brm(prog ~ ses + math, data = dat, family = categorical, refresh = 0)
```

Interpretation

Remember how to interpret logit coefficients? It just got harder!

```
m0
```

```
##  Family: categorical
##   Links: muacademic = logit; muvocation = logit
## Formula: prog ~ ses + math
##    Data: dat (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    -4.10      1.26    -6.59    -1.76 1.00     4103
## muvocation_Intercept     3.03      1.44     0.28     5.93 1.00     3205
## muacademic_sesmiddle     0.33      0.47    -0.61     1.27 1.00     2754
## muacademic_seshigh       0.95      0.55    -0.10     2.05 1.00     2682
## muacademic_math          0.09      0.02     0.04     0.13 1.00     3600
## muvocation_sesmiddle     0.97      0.51    -0.04     1.98 1.00     2484
## muvocation_seshigh       0.37      0.68    -0.98     1.70 1.00     2490
## muvocation_math         -0.07      0.03    -0.13    -0.02 1.00     2720
##                      Tail_ESS
## muacademic_Intercept     3204
## muvocation_Intercept     3322
## muacademic_sesmiddle     3285
## muacademic_seshigh       2760
## muacademic_math          3152
## muvocation_sesmiddle     2646
## muvocation_seshigh       2684
```

Options

Change in log odds of option *k* versus the reference category for a one unit change in *x*

```r
fixef(m0)[, 1]
```

```
## muacademic_Intercept muvocation_Intercept muacademic_sesmiddle
##          -4.10053988           3.02505861           0.32783552
##   muacademic_seshigh      muacademic_math muvocation_sesmiddle
##           0.94943179           0.08536433           0.96640678
##   muvocation_seshigh      muvocation_math
##           0.36564876          -0.07272188
```

Change in odds ratio of option *k* versus the reference category for a one unit change in *x*

```r
exp(fixef(m0)[, 1])
```

```
## muacademic_Intercept muvocation_Intercept muacademic_sesmiddle
##           0.01656373          20.59521171           1.38796066
##   muacademic_seshigh      muacademic_math muvocation_sesmiddle
##           2.58424085           1.08911379           2.62848276
##   muvocation_seshigh      muvocation_math
##           1.44144886           0.92985941
```

Or - we could simulate!

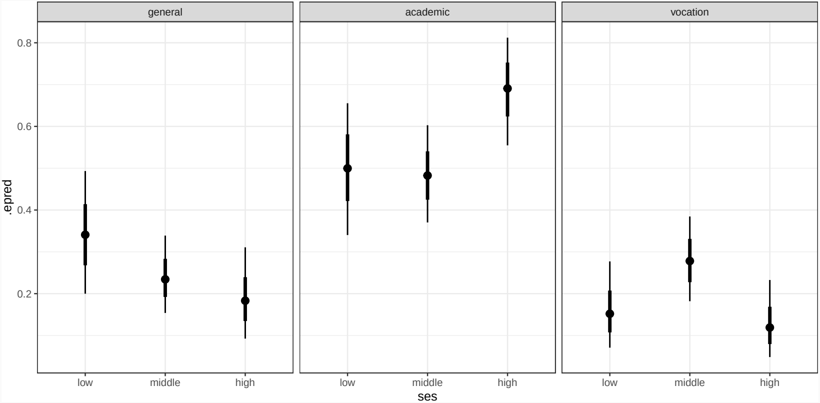
```r
plot_dat <- expand_grid(ses = unique(dat$ses), math = mean(dat$math)) %>%
    add_epred_draws(m0)

head(plot_dat)
```

```
## # A tibble: 6 x 8
## # Groups:   ses, math, .row, .category [1]
##   ses    math  .row .chain .iteration .draw .category .epred
##   <fct> <dbl> <int>  <int>      <int> <int> <fct>      <dbl>
## 1 low    52.6     1     NA         NA     1 general    0.201
## 2 low    52.6     1     NA         NA     2 general    0.284
## 3 low    52.6     1     NA         NA     3 general    0.297
## 4 low    52.6     1     NA         NA     4 general    0.317
## 5 low    52.6     1     NA         NA     5 general    0.317
## 6 low    52.6     1     NA         NA     6 general    0.366
```

Visualize

```
ggplot(plot_dat, aes(y = .epred, x = ses)) + stat_pointinterval() + facet_wrap(~.category)
```



Ordinal regression

The data

```
dat <- read.dta("https://stats.idre.ucla.edu/stat/data/ologit.dta")
head(dat)
```

```
##             apply pared public  gpa
## 1     very likely     0      0 3.26
## 2 somewhat likely     1      0 3.21
## 3        unlikely     1      1 3.94
## 4 somewhat likely     0      0 2.81
## 5 somewhat likely     0      0 2.53
## 6        unlikely     0      1 2.59
```

Ordinal logistic regression

Ordinal logistic regression is a GLM that models the log odds of a rank-ordered categorical outcome as a function of a linear combination of a set of predictors.

Ordinal logistic regression

Ordinal logistic regression is a GLM that models the log odds of a rank-ordered categorical outcome as a function of a linear combination of a set of predictors.

Multinomial logistic regression: basics

For an ordinal outcome with K categories, estimate $K-1$ models where 1,2,3 stand in for membership in group 1, 2, 3, ... K:

$$log \frac{Pr(y_i > 1)}{Pr(y_i = K)} = \beta X_i$$

$$log \frac{Pr(y_i > 2)}{Pr(y_i = K)} = \beta X_i - c_2$$

$$\cdots$$

$$log \frac{Pr(y_i = K-1)}{Pr(y_i = K)} = \beta X_i - c_{k-1}$$

Estimation

We can use `rstanarm` for this with a new function

```
m_ord <- stan_polr(apply ~ pared + gpa, data = dat, prior = NULL, refresh = 0)
```

Model output

```
m_ord
```

```
## stan_polr
##  family:       ordered [logistic]
##  formula:      apply ~ pared + gpa
##  observations: 400
## ------
##       Median MAD_SD
## pared 1.0    0.3   
## gpa   0.6    0.2   
## 
## Cutpoints:
##                             Median MAD_SD
## unlikely|somewhat likely    2.1    0.7   
## somewhat likely|very likely 4.2    0.7   
## 
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

Interpretation

Log odds again!

```
coef(m_ord)
```

```
##     pared       gpa
## 1.0286727 0.5773669
```

Or odds ratios

```
exp(coef(m_ord))
```

```
##    pared      gpa
## 2.797350 1.781342
```

But why not just simulate!

```
expand_grid(gpa = unique(dat$gpa), pared = unique(dat$pared)) %>%
    add_epred_draws(m_ord, ndraws = 500) %>%
    ggplot(aes(x = gpa, y = .epred)) + stat_lineribbon(.width = c(0.5, 0.8, 0.9)) +
    facet_wrap(~pared) + scale_fill_brewer()
```