SOC542 Statistical Methods in Sociology II

Categorical outcomes

Thomas Davidson

Rutgers University

April 11, 2022

Course updates

- Homework 4 will be released on Wednesday
  - Count outcomes
  - Categorical and ordered outcomes
- Replication project workshops (instead of lab final two weeks)

Plan

- Categorical outcomes
- Multinomial logistic regression
- Ordered logistic regression

Categorical outcomes

Categories of categories

- A categorical outcome consists of *three or more discrete categories*
- *Ordered* categorical outcomes
  - e.g. Very good, good, okay, bad, very bad.
- *Unordered* (or nominal) categorical outcomes
  - e.g. Single, in a relationship, married, its complicated, etc.

Categorical outcomes

Intervals

- If a categorical variable is *ordered* there is some sense of an **interval** between categories such that each category can be positioned on a single dimension.
  - These intervals may vary between categories:
    - e.g. The difference between good and very good may be larger than difference between good and okay.
- Categories without *order* do not have clearly defined intervals between categories.

Categorical outcomes

Modeling categories using existing approaches

- OLS regression
  - Only suitable if there are many categories and intervals are *even*
- *One-versus-rest* logistic regression models
  - One model for each category with a binary outcome
  - Limitations: Loss of information

Data

GSS 2018

- Two outcomes from the GSS 2018:
  - Unordered: Marital status
    - Married, widowed, divorced, separated, never
  - Ordered: Self-reported health
    - Excellent, good, fair, poor

Models for categorical outcomes

- We will be considering two different approaches using variations of logistic regression:
  1. Unordered outcomes modeled using **multinomial logistic regression**
  2. Ordered outcomes modeled using **ordinal logistic regression**

Multinomial logistic regression

- **Multinomial logistic regression** models allow us to generalize logistic regression to categorical outcomes and is suitable for *unordered* categories.
- For a set of K outcomes, we can model the linear propensity for outcome k using a linear model with n predictors.

$$\lambda_k = \beta_{0k} + \beta_{1k} x_1 + ... + \beta_{nk} x_n$$

- Rather than estimating a series of separate models, we can jointly estimate a set of equations.

Multinomial logistic regression

- The probability of outcome y_k is represented by the **softmax** link function.[1] The probability of outcome k is the exponentiated linear propensity of outcome k relative to the sum of exponentiated linear propensities of all outcomes in the set K (Kruschke 2015: 650).

$$P(y = k|X) = \text{softmax}_K(\lambda_k) = \frac{e^{\lambda_k}}{\sum_{i \in K} e^{\lambda_i}}$$

[1] The approach is therefore sometimes referred to as **softmax regression**.

Multinomial logistic regression

- Due to the constraints on the system, one category will always produce the following equation:

$$\lambda_r = \beta_{0r} + \beta_{1r}x_1 + ... + \beta_{nr}x_n = 0 + 0x_1 + ... + 0x_n = 0$$

- We therefore select a category to leave out as the *reference category*.
- Model coefficients can then be considered as the log odds of each outcome, relative to the reference category.

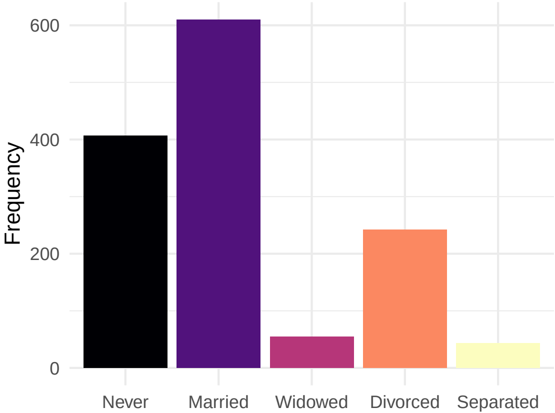
Multinomial logistic regression

Estimation

- The standard `glm` function cannot be used for multinomial outcomes
- Maximum likelihood models can be estimated using the `multinom` function from the `nnet` package[2]
- Bayesian models can be estimated by supplying the `family = categorical(link = "logit")` argument to `brms` models.

[2] Other packages are available but require additional data manipulation before modeling. See this blog for further discussion.

Data: Marital status



Multinomial logistic regression

Estimation

```
library(nnet)
gss$marital <- relevel(gss$marital, ref = "Never")
m1 <- multinom(marital ~ age + sex + log(realrinc) + educ, data = gss)
```

```
## # weights:  30 (20 variable)
## initial  value 2184.007247
## iter  10 value 1667.335362
## iter  20 value 1459.416635
## iter  30 value 1441.935116
## final  value 1441.935011
## converged
```

Multinomial logistic regression

		Married	Widowed	Divorced	Separated
Model 1	(Intercept)	-6.546***	-10.986***	-8.047***	-7.817***
		(0.669)	(1.500)	(0.860)	(1.544)
	age	0.092***	0.187***	0.122***	0.096***
		(0.007)	(0.015)	(0.008)	(0.014)
	sexMale	-0.365*	-1.422***	-0.900***	-0.689*
		(0.153)	(0.347)	(0.196)	(0.347)
	log(realrinc)	0.385***	0.262*	0.413***	0.500**
		(0.069)	(0.128)	(0.087)	(0.167)
	educ	-0.014	-0.125*	-0.081*	-0.197***
		(0.028)	(0.058)	(0.035)	(0.055)

Ref: Never married.

$+\ p < 0.1$, $*\ p < 0.05$, $**\ p < 0.01$, $***\ p < 0.001$

Multinomial logistic regression

		Married	Widowed	Divorced	Separated
Model 1	(Intercept)	0.001***	0.000***	0.000***	0.000***
		(0.001)	(0.000)	(0.000)	(0.001)
	age	1.096***	1.206***	1.130***	1.100***
		(0.007)	(0.018)	(0.009)	(0.015)
	sexMale	0.694*	0.241***	0.407***	0.502*
		(0.107)	(0.084)	(0.080)	(0.174)
	log(realrinc)	1.470***	1.299*	1.511***	1.649**
		(0.102)	(0.166)	(0.131)	(0.275)
	educ	0.987	0.882*	0.922*	0.821***
		(0.027)	(0.051)	(0.032)	(0.045)

Ref: Never married.

$+$ $p < 0.1$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$

Multinomial logistic regression

Interpretation

- Each column is a model comparing a group to the *baseline* (Never married).
- For example, the first column represents the following equation:

$$log(\frac{y = \text{married}}{y = \text{never married}}) = \beta_{10} + \beta_{11}Age + \beta_{12}Sex + \beta_{13}Income + \beta_{14}Educ$$

Multinomial logistic regression

Interpretation

- β_{11} indicates that a one-year increase in age is associated with a .092 change in the log odds of being married compared to never married.
- Like standard logistic regression $e^{\beta_{11}}$ allows us to interpret the coefficient as an odds ratio.
  - This is sometimes interpreted as the **relative risk ratio** of being married vs. never married.

Multinomial logistic regression

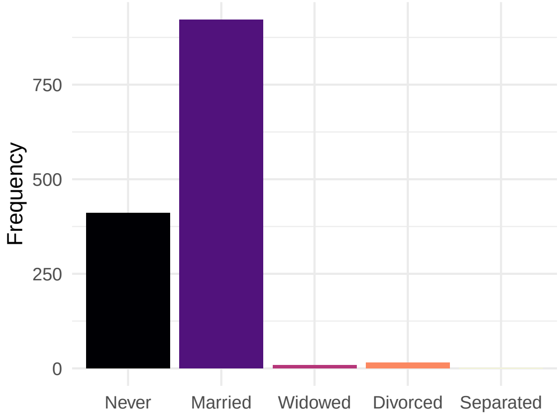
Predictions

The `predict` function returns a factor variable containing the highest probability category for each observation.

```
preds <- predict(m1, gss %>% drop_na(age, sex, realrinc, educ, marital)
preds %>% head(20)
```

```
##  [1] Married  Married  Married  Married  Divorced Married  Married  
##  [9] Married  Widowed  Married  Married  Married  Married  Married  
## [17] Divorced Never    Married  Married
## Levels: Never Married Widowed Divorced Separated
```

Multinomial logistic regression



Multinomial logistic regression

Predictions

- This shows that the model predicts almost all people to be never married or married.
- The model rarely predicts widowed or divorced and did not predict any people to be separated.
- Data imbalances make never/married the most likely categories and omitted variables may help to predict other categories.

Multinomial logistic regression

Predictions

Setting `type = "probs"` returns a vector of probabilities for each observation. Each element indicates $P(y_i = k)$.

```
probs <- predict(m1, type = "probs", gss %>% drop_na())
probs %>% round(3) %>% head(5)
```

```
##   Never Married Widowed Divorced Separated
## 1 0.052   0.459   0.117    0.331     0.042
## 2 0.278   0.522   0.010    0.148     0.042
## 3 0.059   0.692   0.025    0.205     0.019
## 4 0.215   0.611   0.007    0.140     0.027
## 5 0.008   0.265   0.370    0.328     0.029
```

Multinomial logistic regression

Predictions

The probabilities for each observation all sum to one.

```
probs %>% head(5) %>% rowSums() %>% as.numeric()

## [1] 1 1 1 1 1
```

Multinomial logistic regression

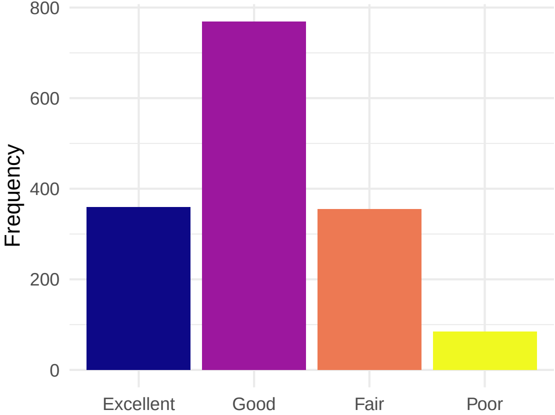
Limitations

- Larger samples required compared to more simple models
- Difficult to evaluate model fit
- Unstable if some variables perfectly predict category membership or have no overlap with certain categories.

Ordinal logistic regression

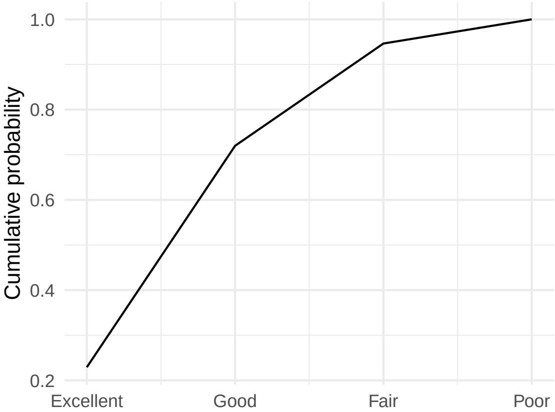
- The multinomial framework could be used for ordinal data, but it ignores any information about the order of categories.
- **Ordinal logistic regression** accounts for ordering by using **cutpoints** to map the intervals between categories onto a linear scale.
- Process:
  - Map categorical outcome onto cumulative probability scale using cumulative link.
  - Convert to log-cumulative-odds, analogue of the logit link for cumulative scale.
  - Construct a linear model to examine association between predictors and outcome, while maintaining information on order.

Data: Self-reported health



Ordinal logistic regression

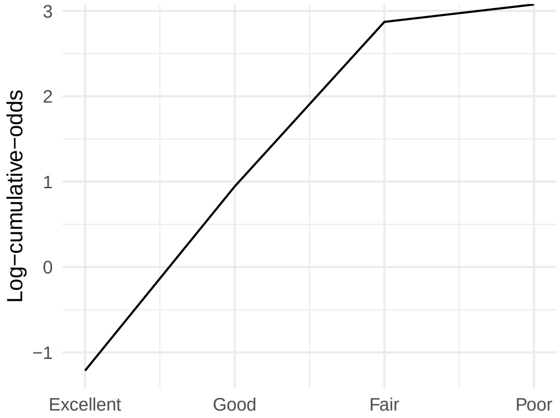
Cumulative probabilities of each class




```
## [1] 0.229 0.720 0.946 1.000
```

Ordinal logistic regression

Log cumulative odds




```
## [1] -1.213  0.944  2.871    Inf
```

Ordinal logistic regression

Estimation

- Each cutpoint on the previous graph representing the log-cumulative-odds that y_i is less than or equal to some value k. These can be considered as group-level *intercepts*.

$$log(\frac{P(y_i \leq k)}{1 - P(y_i \leq k)}) = \alpha_k$$

- The intercept for the final value is ∞ since $log(\frac{1}{1-1}) = \infty$. Therefore we only need $K - 1$ intercepts.

Ordinal logistic regression

Estimation

- If we use the inverse link, we can go back from cumulative-log-odds to cumulative probabilities. The likelihood of k is expressed as

$$p_k = P(y_i = k) = P(y_i \leq k) - P(y_i \leq k - 1)$$

- In the context of your example, we could express the likelihood of "Good" health as

$$p_{\text{good}} = P(y_i = \text{good}) = P(y_i \leq \text{good}) - P(y_i \leq \text{excellent})$$

Ordinal logistic regression

Estimation

- Given this $K-1$ length vector of intercepts, $\alpha_{k \in K-1}$, we can use a linear model to predict the log-cumulative-odds that $y_i = k$ given a matrix of predictors X:

$$\phi_i = \beta X_i$$

$$log(\frac{P(y_i \leq k)}{1 - P(y_i \leq k)}) = \alpha_k - \phi_i$$

Ordinal logistic regression

Estimation

- Once again, we cannot fit such models using `glm`. Instead, we can use the `polr` function from the `MASS` package.
- `rstanarm` includes a `stan_polr` function, which implements a Bayesian version of `polr`.

Ordinal logistic regression

Estimation

The argument `Hess = TRUE` ensures the Hessian matrix is stored, which is necessary for subsequent model evaluation.

```
library(MASS)
m2 <- polr(health ~ age + I(log(realrinc)) + educ + sex + race,
                     data = gss, Hess = TRUE)
```

Ordinal logistic regression[3]

	Log odds	Odds ratios
age	0.004	1.004
	(0.005)	(0.005)
I(log(realrinc))	-0.204	0.815
	(0.058)	(0.047)
educ	-0.102	0.903
	(0.024)	(0.022)
sexMale	0.098	1.102
	(0.130)	(0.144)
raceBlack	0.148	1.160
	(0.174)	(0.201)
raceOther	0.248	1.282
	(0.207)	(0.265)
Num.Obs.	906	906
Log.Lik.	-984.749	-984.749

[3] Significance tests are not provided as standard in ordinal regression output from `polr` so no stars are displayed here.

Ordinal logistic regression

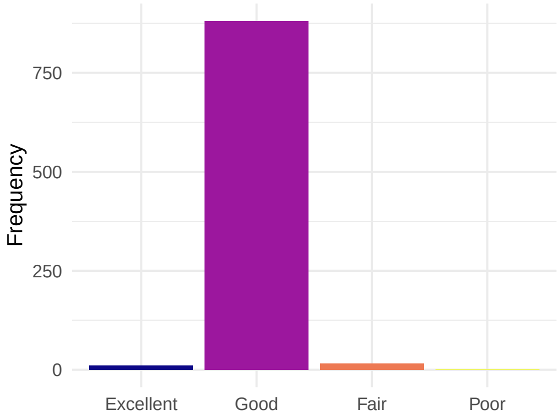
Predictions

```
preds2 <- predict(m2, gss %>% drop_na(health, age, sex, race, realrinc,
preds2 %>% head(20)
```

```
##  [1] Good Good Good Good Good Good Good Good Good Good Good Good Goo
## [16] Good Good Good Good Good
## Levels: Excellent Good Fair Poor
```

Ordinal logistic regression

Predictions



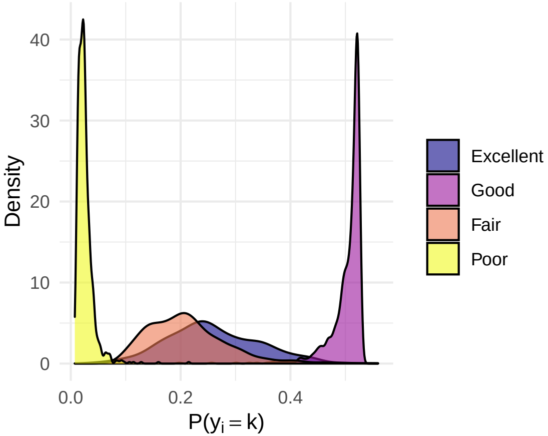
Ordinal logistic regression

Predictions

```
probs2 <- predict(m2, type = "prob",
                  gss %>%
                      drop_na(health, age, sex, race, realrinc, educ))
probs2 %>% round(3) %>% head(5)
```

```
##   Excellent  Good  Fair  Poor
## 1     0.193 0.517 0.258 0.032
## 2     0.219 0.523 0.231 0.027
## 3     0.404 0.470 0.114 0.011
## 4     0.307 0.512 0.163 0.017
## 5     0.174 0.509 0.281 0.036
```

Ordinal logistic regression



Ordinal logistic regression

More predictions

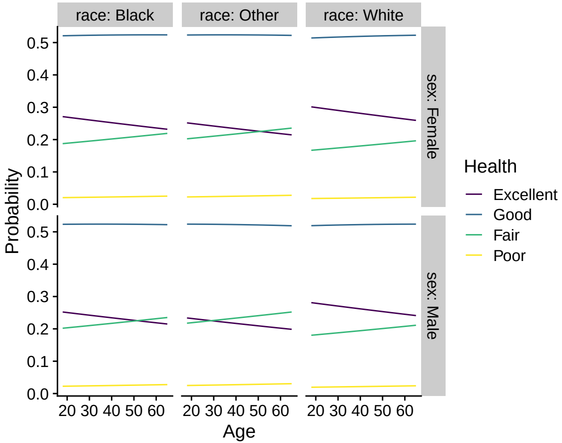
We can easily generate predictions for all combinations of predictors.

```
newdat <- expand_grid(
  race = c("Black", "White", "Other"),
  sex = c("Female", "Male"),
  educ = 12,
  realrinc = c(50000),
  age = 18:65)

newpreds <- predict(m2, newdat, type = "probs")
head(newpreds, 5) %>% round(3)
```

```
##   Excellent  Good  Fair  Poor
## 1     0.271 0.521 0.187 0.021
## 2     0.270 0.521 0.188 0.021
## 3     0.269 0.521 0.189 0.021
## 4     0.268 0.521 0.189 0.021
## 5     0.268 0.522 0.190 0.021
```

Ordinal logistic regression



Ordinal logistic regression

Cutpoints

The cutpoints can be extracted from the model using the `zeta` parameter.

```
cuts <- m2$zeta
print(cuts)
```

```
## Excellent|Good      Good|Fair      Fair|Poor
##     -4.1986283     -1.8720014      0.6567534
```

Ordinal logistic regression

Cutpoints

We can obtain the probability associated with each cutpoint by using the inverse logit function, $\frac{e^x}{1+e^x}$.

```
inv.logit <- function(x) {
    return(exp(1)^x / (1 + exp(1)^x))
    }

cut.probs  <- inv.logit(cuts)
cut.probs %>% round(3) %>% print()
```

```
## Excellent|Good      Good|Fair      Fair|Poor
##          0.015          0.133          0.659
```

Ordinal logistic regression

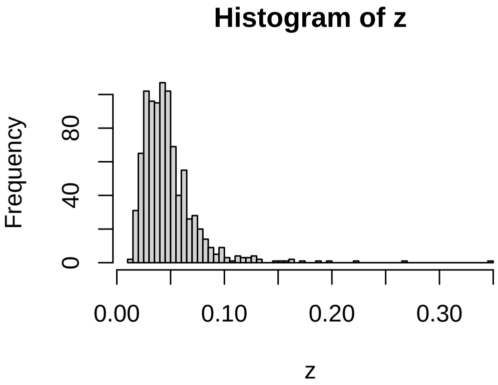
Latent variables

One way to understand the model is to extract a latent variable representing the predicted position of each outcome on the cumulative probability scale without subtracting the intercepts. We can then observe where each observation falls between the cutpoints.

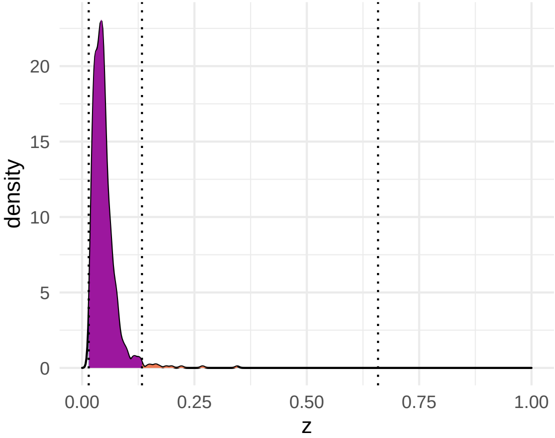
```
z <- m2$lp  %>% inv.logit()
z %>% head(10) %>% round(3)
```

```
##     7     8    11    14    16    19    21    24    25    27
## 0.059 0.051 0.022 0.033 0.067 0.018 0.033 0.048 0.038 0.051
```

Ordinal logistic regression



Ordinal logistic regression



Ordinal logistic regression

Limitations

- Similar to multinomial logistic regression
  - Larger samples required compared to more simple models
  - Difficult to evaluate model fit
  - Unstable if some variables perfectly predict category membership or have no overlap with certain categories
- Additionally, the models assume that the relationship between the predictors and each pair of outcomes is the same (hence on set of coefficients). This is known as the **proportional odds assumption**. Additional tests are required to verify this is met.[4]

[4] See the UCLA stats blog for details.

Categorical outcomes

Frequentist and Bayesian approaches

- Due to the complexity of the models, many frequentist approaches require additional testing and analysis to diagnose issues and assess model fit
- In contrast, we can use the same tools to evaluate Bayesian models:
    - Trace plots and MCMC diagnostics for estimation issues
    - LOO-CV and WAIC for fit
    - PSIS diagnostics for outliers
    - Posterior predictive checks for predictions and fit
- Either way, these models are more cumbersome to work with than other single-equation GLMs

Summary

- Categorical outcomes can be modeled using specialized types of generalized linear models
- Unordered categories
  - Multinomial logistic regression
- Ordered categories
  - Ordinal logistic regression
  - OLS if many categories and equal intervals
- These models are complex and more difficult to fit and interpret than previous models we have covered

Next week

- Data structures
  - Clustering and nesting
    - Standard errors
    - Fixed effects
    - Random effects
  - Autocorrelation
    - Time
    - Space
    - Networks
- Project workshop