PSM II Tutorial 7: Multinomial logistic regression

1 Introduction

This week we'll use data from the British Crime Survey 2007–08 to investigate what factors influence people's beliefs about the main causes of crime in Britain. This dataset is quite old, but has been made available for teaching students how to analyse survey data of this type (access to up-to-date data from this survey requires an extensive vetting process for each researcher).

This tutorial will use the easystats collection of R packages for working with regression output and the tidyverse collection of packages for wrangling data. We will also use the nnet package to fit multinomial logistic regression models.

You might need to install some packages that haven't been used in previous weeks:

```
install.packages("labelled", "nnet", "skimr")
```

To get started, download the `bcs_2007_8_teaching_data_unrestricted.dta` data file from Moodle. Put this file in the folder on your computer that you are using for this module, then create a new R script in RStudio (`File > New File > R Script`) and save it as `tutorial_07.R` in the same folder as the data file.

Start your script file by loading the packages we will need:

```
library(easystats)
library(nnet)
library(tidyverse)
```

You can now load the data using the `read_dta()` function from the haven package:

```
bcs_data <- haven::read_dta("bcs_2007_8_teaching_data_unrestricted.dta")
```

2 Get to know the data

You already know lots of R functions for inspecting data in different ways in R. For example:

- `summary()` to see a summary of each variable,
- `nrow()` to find out how many rows there are in a dataset,
- `View()` to see the data in a separate window,
- `glimpse()` (from the dplyr package that is part of tidyverse) to see the first few rows of each column, and
- `count()` (also from dplyr) to count unique values in a variable.

One function we can use to get a fairly comprehensive summary of a dataset is the `skim()` function from the skimr package.

```
skimr::skim(bcs_data)
```

Question 1: Look at the column `complete_rate`: what columns in the data have a substantial proportion of missing values?

Since our data is in `.dta` format (which has explanatory labels attached to each column and value), we can also use the `look_for()` function from the labelled package to get an explanation of what each variable measures.

```
labelled::look_for(bcs_data)
```

Question 2: What does the variable `winsult` stand for? How is this variable measured?

3 Prepare the data for modelling

Before we can use the data to create a model, we need to make some minor changes to the data. First, use the `count()` function to find all the unique values of the `causem` column in the data. That column contains the answer each participant gave when asked what they considered to be the main cause of crime in Britain.

```
count(bcs_data, causem)
```

We could use this variable without making any changes, but since there are so many categories in the data that would make the resulting model quite hard to interpret. For the sake of simplicity, we will first aggregate this variable so that causes of crime are described as being 'social', 'personal' or 'other'.

At the same time as doing this, we will also:

1. Specify we want the model to treat the 'other' value in the new variable as the reference value.

2. Convert categorical variables where the numbers are represented by numeric codes into factors (the variable type R uses to store categorical variables), since this makes it easier to interpret the model.
3. Remove the 'refused' and 'don't know' values from some columns that we will use as explanatory variables, together with any missing values.

```
bcs_data_for_model <- bcs_data |>
  mutate(
    # Aggregate the outcome variable so it has fewer categories
    cause_crime = case_match(
      causem,
      c(3, 4, 8) ~ "personal",
      c(2, 5, 6, 7) ~ "social",
      .default = "other"
    ),
    # Set the reference value to be 'other'
    cause_crime = fct_relevel(cause_crime, "other"),
    # Convert categorical variables to factors
    across(c(ethgrp2, sex, work), as_factor)
  ) |>
  # Remove 'refused' and 'don't know' values from variables we want to use
  filter(work %in% c("yes", "no")) |>
  droplevels() |>
  # Remove missing values from variables we want to use
  drop_na(age, ethgrp2, sex, work)
```

Question 3: How many participants said they thought crime was mainly caused by personal, social or other causes? Use `count(bcs_data_for_model, cause_crime)` in the R console to find out.

4 Fitting the model

Now we can start to create some models to understand how participants' beliefs about causes of crime are related to some explanatory variables. Let's start with a very basic model that just suggests beliefs about causes of crime will vary between men and women.

```
bcs_model_sex <- multinom(cause_crime ~ sex, data = bcs_data_for_model)
```

We can use `test_performance()` from the performance package (part of the easystats collection of packages)

```
test_performance(bcs_model_sex)
```

Question 4: Is this model better than a 'null' or 'empty' model containing no predictors? Look at the *p*-value associated with the test for differences between the 'full model' (contained in the `bcs_model_sex` object) and a null model. Type `?test_performance` in the R console if you need more help.

Let's see how other demographic factors might be related to beliefs about the main causes of crime. Create a model that includes sex, age and whether the person did any paid work in the past week.

```
bcs_model_age_sex_work <- multinom(
  cause_crime ~ sex + age + work,
  data = bcs_data_for_model
)
```

Question 5: Which explanatory variables are associated with significant increases or decreases in the likelihood of people saying crime is due to personal or social issues? Use `model_parameters(bcs_model_age_sex_work, exponentiate = TRUE)` to inspect the model coefficients and/or `plot(model_parameters(bcs_model_age_sex_work, exponentiate = TRUE), log_scale = TRUE)` to see a dot-and-whisker plot.

Now add ethnic group as another explanatory variable.

```
bcs_model_age_sex_work_eth <- multinom(
  cause_crime ~ sex + age + work + ethgrp2,
  data = bcs_data_for_model
)
```

Question 6: What is the reference value of the variable `ethgrp2`? Use `levels(pull(bcs_data_for_model, "ethgrp2"))` to see which level is first, since the model will use the first level as the reference value.

5 Using predicted values to understand relationships between variables

One way to understand the results produced by regression models is to apply them to specific combinations of values of the explanatory variables. For example, we can work out what a woman aged 35 who works and is Black is most likely to think is the main cause of crime in Britain. In a previous

week we used the `estimate_expectation()` function from the modelbased package (part of the easystats family of packages) to do this automatically. `estimate_expectation()` does not work for multinomial models, but in any case it is sometimes useful to calculate predicted values manually because this gives us the ability to control which combinations of explanatory variables we are interested in.

The first step in calculating predicted values is to create a new dataset containing all the combinations of explanatory variables that we are interested in. We can create a dataset with the `expand_grid()` function from the tidyr package (part of the tidyverse).

We must make sure that there is column in our new dataset with the same name and data type (numeric, character, factor, etc.) as each of the explanatory variables in the model. We can specify the values we want to predict results for either manually (e.g. `sex = c("female", "male")`) or by extracting the values from the existing data using `pull()` and then finding all the possible values using either `unique()` (for categorical variables) or `range()` for numeric variables.

```
bcs_newdata <- expand_grid(
  sex = c("female", "male"),
  age = 18:64,
  work = unique(pull(bcs_data_for_model, "work")),
  ethgrp2 = unique(pull(bcs_data_for_model, "ethgrp2"))
)

head(bcs_newdata)
```

Now we have the values we want to use as the basis for our predictions, we can predict how likely the model estimates it is that each combination of explanatory variables will produce each possible value of the outcome variable.

```
bcs_predictions <- bcs_model_age_sex_work_eth |>
  # Predict the probability of each outcome
  predict(newdata = bcs_newdata, type = "probs") |>
  # Join the values of the explanatory variables
  bind_cols(bcs_newdata) |>
  # Convert the result to 'long' format because this is easier for plotting
  pivot_longer(
    cols = c("other", "personal", "social"),
    names_to = "category",
    values_to = "prob"
  )
```

Finally, we can create a plot to show the relationship between the explanatory variables and the outcome variable. To do this we have to assign each colum in the `bcs_predictions` dataset to either an aesthetic (e.g. x, y, colour) or to create rows/columns of small-multiple charts using `facet_grid()`.

```
ggplot(
  bcs_predictions,
  aes(x = age, y = prob, colour = category, linetype = sex)
) +
  geom_line() +
  facet_grid(rows = vars(work), cols = vars(ethgrp2)) +
  theme_minimal()
```

Question 7: What can we say from these charts about how the likelihood of a White person believing the main cause of crime is a social issue varies with age compared to a Black person?