PSM II Tutorial 8: Ordinal logistic regression

1 Introduction

This week we will use data from the Crime Survey for England and Wales to identify what factors are associated with people being more or less worried about having their car stolen.

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 MASS and VGAM packages to fit ordinal logistic regression models.

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

```
install.packages(c("brant", "MASS"))
```

To get started, download the `csew1314teachingopen.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_08.R` in the same folder as the data file.

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

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

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

```
csew_data <- haven::read_dta("csew1314teachingopen.dta")
```

2 Get to know the data

As in previous weeks, we can use functions such as `skimr::skim()`, `labelled::look_for()` and `dplyr::count()` to understand the dataset.

Question 1: Use these functions to find out: (a) what is the name of the column in `csew_data` that contains information about how worried people feel about having their car stolen, and (b) what values this variable can have.

You might want to check the effect of setting the `details` argument to the `labelled::look_for()` function to either `FALSE` or `TRUE` in terms of what information it produces.

3 Prepare the data for modelling

Before we can create a model, we need to wrangle the data so it is suitable for the model we want to create. As in previous tutorials, the first step is to convert labelled numeric columns in the `.dta` data file into factors.

Use `count(csew_data, wcarstol)` to inspect the outcome variable. Two things are notable about this how this variable is formatted:

1. The `wcarstol` column contains 30 rows with the value 'Not applicable' (for those respondents who didn't have access to a car) and 7,080 rows with missing values (for those respondents who were not asked this question). These values won't work with ordinal regression because neither 'Not applicable' nor a missing value fit into an ordered scale. For example, it is meaningful to say that the value 'very high' is higher than the value 'high', but not meaningful to say that the value 'not applicable' is higher or lower than the value 'low'.
2. The remaining values of `wcarstol` have the levels 'Very worried', 'Fairly worried', 'Not very worried' and 'Not at all worried' in that order. This means that as you go 'up' the scale of values from the first level to the last, people are *less* worried. If we keep the levels in this order then we will find it harder to interpret the odds ratios produced by our models later. Fortunately, we can easily reverse the order of the levels of this variable using the `fct_rev()` function from the forcats package.

As part of the modelling process, functions in R such as `lm()` and `glm()` automatically remove rows in a dataset that have missing values in any of the columns that will be used in the model. This means it isn't necessary to remove missing values manually, but it can be useful to do so. This is particularly useful when it's necessary to compare multiple models, since some of the R functions we can use to compare models only work on two models that are based on the same data. For this reason, we will remove the missing values of the outcome and explanatory variables using `drop_na()`, as well as removing the 'not applicable' values using `filter()`.

```
csew_data_for_model <- csew_data |>
  mutate(
    across(where(labelled::is.labelled), as_factor),
    wcarstol = fct_rev(wcarstol)
  ) |>
  drop_na(wcarstol, sex, agegrp7, ethgrp2a, bcsvictim) |>
  filter(wcarstol != "Not applicable") |>
  droplevels()
```

Question 2: What levels does the `wcarstol` column in the `csew_data_for_model` object have, and in what order?

4 Fitting a model

Now we have prepared the data, we can fit a model. Ordinal logistic regression models are created using the `polr()` function from the MASS package. The first two arguments to this function provide the model formula and the data we want to use. The argument `Hess = TRUE` tells `polr()` to return some extra information that makes it faster to work with the model object later.

Let's say that we want to see if sex, age, ethnic group and whether a person has been a victim of crime in the past 12 months is related to the likelihood that a person will be worried about their car being stolen. We can put all of these explanatory variables in a model.

```
wcarstol_model <- MASS::polr(
  wcarstol ~ sex + agegrp7 + ethgrp2a + bcsvictim,
  data = csew_data_for_model,
  Hess = TRUE
)
```

Question 3: What is the null hypothesis for this model?

5 Checking the model assumptions

Most of the assumptions of ordinal regression are the same as those for other types of regression model. For example, the model must make sense, the observations must have been made independently, and the results must not be unduly influenced by highly correlated explanatory variables or outliers. One assumption that is particular to ordinal regression is the *proportional odds assumption*. We can check this assumption by using the `brant()` function from the brant package to conduct a Brant test.

```
brant::brant(wcarstol_model)
```

Question 4: What do the Brant test results tell us about whether our model is compatible with the proportional odds assumption?

6 Interpreting the model

As in previous weeks, we can use the `model_parameters()` function to look at the odds ratios for the explanatory variables in this model.

```
model_parameters(wcarstol_model, exponentiate = TRUE)
```

Remember you can create a plot of these coefficients and the 95% confidence intervals, if you find a chart easier to interpret:

```
plot(model_parameters(wcarstol_model, exponentiate = TRUE), log_scale = TRUE)
```

Question 5: How would you interpret the odds ratios for the different levels of `sex`, `agegrp7`, `ethgrp2a` and `bcsvictim`, using the 'Odds Ratio' and 'p' columns in the table?

7 Fitting alternative models

The Brant test results above suggest our model is not compatible with the proportional odds assumption. When this assumption is violated, it often means it is necessary to use a different type of model. There are several options, of which we will just try one: linear regression. This might seem a strange choice, since we know linear regression requires an outcome variable that is continuous and normally distributed. However, it can sometimes be useful to fit a linear model to use as a comparison with another type of model.

Linear regression requires a continuous outcome variable. Fortunately, it is easy to convert an ordered categorical variable to a numeric variable in R using the `as.numeric()` function. This will replace the four levels 'Not at all worried', 'Not very worried', 'Fairly worried' and 'Very worried' into the numbers 1 to 4. Otherwise, we don't need to do any other data preparation to fit a linear model to our existing data.

```
wcarstol_model_lm <- lm(
  as.numeric(wcarstol) ~ sex + agegrp7 + ethgrp2a + bcsvictim,
  data = csew_data_for_model
)
```

We can compare this model to our ordinal model using the `compare_performance()` function from the performance package.

```
compare_performance(wcarstol_model, wcarstol_model_lm)
```

This function produces several statistics for comparing which model provides the best fit for the data. The statistic to focus on is the column marked AIC. We don't need to understand this statistic in any detail, we just need to remember that the model with the lowest AIC value is the model that best fits the data. The actual AIC value cannot be interpreted directly: it is only used by *comparing* two or more AIC values for different models.

Question 6: Which model best fits the data, the ordinal model or the linear model?

8 Predicting values

An alternative way to try to understand the results from a regression model is to look at what values the model predicts for the outcome variable given particular values of the explanatory variables. We covered how to do this in more detail in last week's tutorial, so this week we can use very similar code.

```
# Step 1: specify what values of the explanatory variables we're interested in
# Remember the new data object we create must include values for **all** of the
# explanatory variables in our model. This week, we will use `unique()` to
# include every unique value in the original data in the new data we want to
# apply the model to.
csew_newdata <- expand_grid(
  sex = unique(pull(csew_data_for_model, "sex")),
  agegrp7 = unique(pull(csew_data_for_model, "agegrp7")),
  ethgrp2a = unique(pull(csew_data_for_model, "ethgrp2a")),
  bcsvictim = unique(pull(csew_data_for_model, "bcsvictim"))
)

# Step 2: produce predicted values of outcome variable
csew_predicted <- wcarstol_model |>
  # Predict the probability of each outcome
  predict(newdata = csew_newdata, type = "probs") |>
  # Join the values of the explanatory variables
  bind_cols(csew_newdata) |>
  # Convert the result to 'long' format because this is easier for plotting
  pivot_longer(
    cols = contains("worried"),
    names_to = "category",
    values_to = "prob"
  ) |>
  # Match ordering of values between original data and new `category` variable
  mutate(
    category = factor(
      category,
      levels = levels(pull(csew_data_for_model, "wcarstol"))
    )
  )

# Step 3: plot the predicted values
ggplot(
  csew_predicted,
  aes(x = agegrp7, y = prob, fill = category, group = category)
) +
  geom_col(position = position_fill()) +
  facet_grid(cols = vars(ethgrp2a), rows = vars(sex, bcsvictim)) +
  scale_fill_ordinal() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = -55, hjust = 0))
```

Question 7: What does this tell us about which groups in society are most likely to be more worried about having their car stolen?