Exercises: Categorical data in the General Social Survey

YSC2210 - DAVis with R

Michael T. Gastner

Job opportunity (early February 2022): earn $10 for 1 hour

Yale-NUS research study on data visualisation

One of the MCS capstone students is conducting an experiment about thematic map design. As participant, you can earn S$10 or 60 minutes of course credit (e.g. for 'Introduction to Psychology').

The experiment takes place on the Yale-NUS campus and lasts approximately 50-60 minutes in a single session. You will be asked to perform several map reading tasks. If you are interested, please sign up at https://bit.ly/cartogram-study.

General Social Survey

The General Social Survey is a biannual survey conducted by the National Opinion Research Center at the University of Chicago. The survey collects demographic information from representative respondents of the US population. It also records information about their opinions, attitudes and behaviours.

The **forcats** package contains a data frame `gss_cat` with responses to some of the survey questions. Most columns in `gss_cat` contain categorical data encoded as factors. In this exercise, we use some of these data to practise working with factors. For background information about the data, please have a look at the documentation (`?gss_cat`) and the web site of the General Social Survey (https://gss.norc.org/About-The-GSS).

For this exercise, we need the packages **formattable**, **kableExtra** and various tidyverse packages; thus, please put the following code immediately below your setup chunk.

```
library(formattable)
library(kableExtra)
library(tidyverse)
```

1 'Race' of respondents

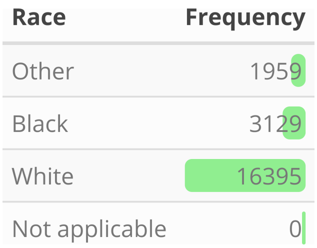
The 'race' of the respondents is the factor `gss_cat$race`.

(a) What are the levels of `gss_cat$race`?

(b) Tabulate the frequency of the 'races' with `table()`.

(c) We can obtain the table in a more readable format with the help of **formattable** and **kableExtra**. You do not need to understand the following code chunk in detail. Just accept it as a recipe for making nice tables. The table produced by this code is shown in figure 1.

```
table(gss_cat$race) |>
  as.data.frame() |>
  mutate(Freq = color_bar("lightgreen")(Freq)) |>
```



Race	Frequency
Other	1959
Black	3129
White	16395
Not applicable	0

Figure 1: We can produce attractive tables with the packages **formattable** and **kableExtra**. The code for this table is given in the exercise.

```
kbl(
  col.names = c("Race", "Frequency"),
  align = c("l", "r"),
  escape = FALSE
) |>
kable_styling(c("striped", "condensed"), full_width = FALSE)
```

Because we also want to apply the same typesetting technique to most of the other tables in this exercise, let us write a function `nice_table()` that takes two arguments:

(i) the name of the column in `gss_cat` (`"race"` in the example above)
(ii) the column name to be printed in the table (`"Race"` in this example)

(d) From the table in (c), we can see that one level is not present in the data. Create a new column `gss_cat$race_drop_unused` that has the same data as `gss_cat$race`, but drop the unused level.

(e) Make a 'nice' frequency table of `race_drop_unused`. Confirm that the unused level was dropped.

(f) Sort the levels in the order of frequency. Put the most frequent 'race' at the top of the 'nice' table.

2 Investigating the reported income

(a) The reported income is the factor `gss_cat$rincome`. What are the levels?
(b) Without editing labels and levels, tabulate the frequency of the income levels with the `nice_table()` function from (1)-c. Explain why this table would not be suitable for a published report.
(c) Improve the table. You may want to look at the **forcats** cheat sheet for help: https://raw.githubusercontent.com/rstudio/cheatsheets/main/factors.pdf.

3 Party affiliation

(a) What are the levels of `gss_cat$partyid`?
(b) Add a factor column `gss_cat$partyid_aggregated` in which (in this order):
  (i) `"Strong republican"` and `"Not str republican"` become `"Republican"`.
  (ii) `"Ind,near rep"`, `"Independent"` and `"Ind,near dem"` become `"Independent"`
  (iii) `"Not str democrat"` and `"Strong democrat"` become `"Democrat"`.
  (iv) `"Other party"` becomes `"Other`".
  (v) `"No answer"` and `"Don't know"` become `"Missing"`.
(c) Make a frequency table of `gss_cat$partyid_aggregated` with the `nice_table()` function from (1)-c.

4 Religion and denomination

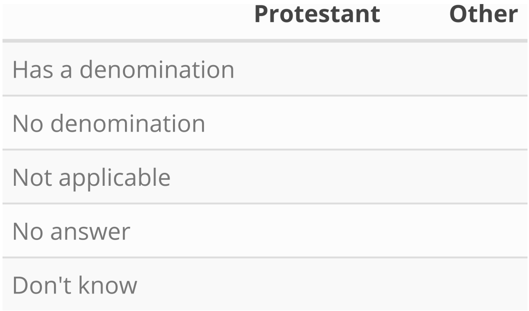
In this section, we want to demonstrate that 'denomination' is predominantly a Protestant concept.

(a) Add a factor column `gss_cat$is_protestant` with levels `"Protestant"` and `"Other"`.

(b) Add a factor column `gss_cat$has_denom` with levels (in this order):

  (i) `"Has a denomination"`
  (ii) `"No denomination"`
  (iii) `"Not applicable"`
  (iv) `"Don't know"`
  (v) `"No answer"`

(c) Make a two-way table as indicated in figure 2. The function `nice_table()` makes one-way table; thus, the function is not directly applicable. Be creative and search for help online!



	Protestant	**Other**
Has a denomination		
No denomination		
Not applicable		
No answer		
Don't know		

Figure 2: In part 4 of the exercise, you are asked to fill in the numbers in this two-way table.