Week 2: Some more tidy stuff

SOC6302 Statistics for Sociologists

Table of contents

1 Load packages and read in GSS

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.0      v purrr   0.3.5
v tibble  3.1.8      v dplyr   1.0.10
v tidyr   1.2.1      v stringr 1.4.1
v readr   2.1.2      v forcats 0.5.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
gss <- read_csv(file = "../data/gss.csv")
```

```
Rows: 20602 Columns: 85
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (63): sex, place_birth_canada, place_birth_father, place_birth_mother, p...
dbl (21): caseid, age, age_first_child, age_youngest_child_under_6, total_ch...
lgl  (1): main_activity

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

2 Creating your own dataset

Note that you can also create your own dataset using the `tibble` function. Note that each column is defined as a vector:

```
my_fruit <- tibble(fruit = c("banana", "apple", "pear"),
                   count = c(2,4,1))
my_fruit
```

```
# A tibble: 3 x 2
  fruit  count
  <chr>  <dbl>
1 banana     2
2 apple      4
3 pear       1
```

3 More important functions

3.1 The `group_by` function

The `group_by` function allows you to get key summary statistics by group (levels of a categorical variable). Use in combination with `summarize` etc

e.g. mean age and standard deviation by marital status in GSS

```r
gss |>
  group_by(marital_status) |>
  summarize(mean_age = mean(age),
            sd_age = sd(age)) |>
  arrange(mean_age)
```

```
# A tibble: 7 x 3
  marital_status        mean_age sd_age
  <chr>                    <dbl>  <dbl>
1 Single, never married     38.1  17.2
2 Living common-law         44.6  14.5
3 Separated                 54.5  13.7
4 Married                   54.9  14.8
5 Divorced                  61.0  11.4
6 <NA>                      65.8  12.9
7 Widowed                   73.0   8.47
```

Note that the above table shows the mean and sd of age for when marital status is missing (NA). We may want to remove those. To do this, use the `drop_na` function

```r
gss |>
  drop_na(marital_status) |>
  group_by(marital_status) |>
  summarize(mean_age = mean(age),
            sd_age = sd(age)) |>
  arrange(mean_age)
```

```
# A tibble: 6 x 3
  marital_status        mean_age sd_age
  <chr>                    <dbl>  <dbl>
1 Single, never married     38.1  17.2
2 Living common-law         44.6  14.5
3 Separated                 54.5  13.7
4 Married                   54.9  14.8
5 Divorced                  61.0  11.4
6 Widowed                   73.0   8.47
```

4 Calculating the correlation coefficient

To calculate the correlation coefficient between two quantitative variables, e.g. age and age at first marriage, use the `summarize` function. Notice that we need to remove rows with any NA values before doing the calculation. We can do this using `drop_na()`

```
gss |>
  select(age, age_at_first_marriage) |>
  drop_na() |>
  summarise(correlation = cor(age, age_at_first_marriage))
```

```
# A tibble: 1 x 1
  correlation
        <dbl>
1      -0.154
```

5 Counts and proportions

5.1 Counting the number of observations

Often we would like to include counts of observations in particular groups. To do this, use the `tally()` or `count()` function.

e.g. the number of people by province of residence in the GSS

```
gss |>
  group_by(province) |>
  tally()
```

```
# A tibble: 10 x 2
   province                     n
   <chr>                    <int>
 1 Alberta                   1728
 2 British Columbia          2522
 3 Manitoba                  1192
 4 New Brunswick             1337
 5 Newfoundland and Labrador 1094
 6 Nova Scotia               1425
 7 Ontario                   5621
 8 Prince Edward Island       708
```

```
 9 Quebec                     3822
10 Saskatchewan               1153
```

5.2 Getting the proportion in each group

Also often useful to get proportion of total in each group:

```
gss |>
  group_by(province) |>
  tally() |>
  mutate(prop = n / sum(n))
```

```
# A tibble: 10 x 3
   province                     n   prop
   <chr>                    <int>  <dbl>
 1 Alberta                   1728 0.0839
 2 British Columbia          2522 0.122
 3 Manitoba                  1192 0.0579
 4 New Brunswick             1337 0.0649
 5 Newfoundland and Labrador 1094 0.0531
 6 Nova Scotia               1425 0.0692
 7 Ontario                   5621 0.273
 8 Prince Edward Island       708 0.0344
 9 Quebec                    3822 0.186
10 Saskatchewan              1153 0.0560
```

6 Review questions

1. How many respondents were born in Canada?
2. What proportion of respondents were born in Canada?
3. Calculate respondents mean age by whether or not they were born in Canada