Grouping and summarizing

Steve Huguenin

2025-10-13

Identify the duplicated row

Identify the duplicated subject.

Compute basic statistics for all moral dilemma columns.

Compute the mean, the median, the standard deviation as well as min and max values. Find meaningful short names for the functions such as `med` for the median.

Assign `judgments_condition_stats` to the results.

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.1     v stringr   1.5.2
v ggplot2   4.0.0     v tibble    3.3.0
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.1.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
```

```r
judgments <- read_tsv("https://biostat2.uni.lu/practicals/data/judgments.tsv")
```

```
Rows: 188 Columns: 158
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr   (5): start_date, end_date, condition, gender, logbook
dbl (153): finished, subject, age, mood_pre, mood_post, STAI_pre_1_1, STAI_p...
```

```
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.
```

```
judgments_condition_stats <- judgments |>
  reframe(
    across(starts_with("moral_dilemma"), list(avg = mean, med = median, std = sd, min = mi
  )
judgments_condition_stats
```

```
# A tibble: 1 x 35
  moral_dilemma_dog_avg moral_dilemma_dog_med moral_dilemma_dog_std
                  <dbl>                 <dbl>                 <dbl>
1                  7.35                     8                  2.17
# i 32 more variables: moral_dilemma_dog_min <dbl>,
#   moral_dilemma_dog_max <dbl>, moral_dilemma_wallet_avg <dbl>,
#   moral_dilemma_wallet_med <dbl>, moral_dilemma_wallet_std <dbl>,
#   moral_dilemma_wallet_min <dbl>, moral_dilemma_wallet_max <dbl>,
#   moral_dilemma_plane_avg <dbl>, moral_dilemma_plane_med <dbl>,
#   moral_dilemma_plane_std <dbl>, moral_dilemma_plane_min <dbl>,
#   moral_dilemma_plane_max <dbl>, moral_dilemma_resume_avg <dbl>, ...
```

Sort by groups

Find the number of subjects by `age`, `gender` and `condition`, e.g. how many 20 years of age females are in the `stress` group.

Sort the resulting tibble such that the condition that contains the most populous group is sorted first (i.e. `stress` or `control` appear together).

```
judgments |>
  group_by(age, gender, condition, .add = FALSE, .drop = TRUE) |>
  tally(sort=TRUE)
```

```
# A tibble: 33 x 4
# Groups:   age, gender [20]
     age gender condition     n
   <dbl> <chr>  <chr>     <int>
 1    18 female stress       27
 2    18 female control      25
 3    19 female stress       19
 4    19 female control      17
 5    20 female stress       14
 6    22 female stress        9
 7    18 male   control       7
 8    21 female control       7
 9    20 male   control       6
10    17 female stress        5
# i 23 more rows
```

Ensure that the resulting tibble does not contain groups.