Contents

Rdionica: priprema datoteke `podaci_upitnik.csv` za obradu

```
install.packages(c('tidyverse', 'here',
                   'wrapr', 'conflicted',
                   # ovi paketi nisu učitani tijekom ovog
                   # dijela radionice:
                   'DataExplorer', 'naniar', 'visdat',
                   'skimr', 'janitor', 'psych'))
```

U ovom dijelu radionice proći ćemo put od sirovih podataka do podataka na kojima možemo provesti analizu. Prije nego što se bacimo na učitavanje i proučavanje sirovih podataka, učitat ćemo pakete koje ćemo koristiti.

Pakete učitavamo pozivanjem funkcije `library`, koja kao argument prima ime **jednog** paketa.

```
# skupina paketa koja sadrži većinu paketa koje
# ćemo koristiti za baratanje podacima
library(tidyverse)
## -- Attaching packages --------------------------------- tidyverse 1.2.1 --
## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.7
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

# paket koji sadrži 'pipe' operatore
library(magrittr)
##
## Attaching package: 'magrittr'
## The following object is masked from 'package:purrr':
##
##     set_names
## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
# upozorava na konflikte u imenima funkcija
# koji se javljaju kad više paketa koristi isto
# ime
library(conflicted)

# omogućava učitavanje .SAV fielova
library(haven)

# omogućava učitavanje .xlsx fielova
library(readxl)

# paket koji sadrži neke zgodne olakšice
library(wrapr)

# olakšava korištenje relativnih file pathova
library(here)
## here() starts at /home/denis/Documents/rdionica
```

Učitavanje podatka

Za početak, pogledat ćemo kako izgledaju naši sirovi podaci.

A da bismo to učinili, prvo ih moramo učitati u R.

Vidjet ćemo kako učitati tri vrste datoteka: SPSS-ov `.sav`, Excelov `.xls/xlsx` te generički *comma separated values* file - `.csv`.

SPSS - .sav

`.sav` datoteke možemo učitati koristeći funkciju `read_sav` iz paketa `haven` (dio `tidyversea`). Funkcija kao argument prima samo put do datoteke koju želimo učitati.

```r
podaci_spss <- read_sav(here('podaci', 'podaci_upitnik.sav'))
```

Funkcija `here` konstruira relativni put do datoteke `podaci_upitnik.csv`, koji kreće od *root* foldera, a koji je označen prisustvom prazne datoteke imena `.here`. To je jedan od načina koji osigurava reproducibilnost obrada pri prijenosu koda s jednog računala na drugo i lišava nas muke ručnog mijenjanja puteva do datoteka. Isto postižemo stvaranjem projekta u RStudiju. Osim na datoteku `.here`, funkcija `here` reagira i na datoteke sa sufiksom `.Rproj` (koje nastaju pri stvaranju RStudio projekta). Kad pogledamo učitane podatke, primjećujemo nešto neobično kod `pi_` varijabli.

Koristeći funkciju `head` (`tail`) možemo pogledati, po defaultu, prvih (posljednjih) 6 redova tablice.

```r
head(podaci_spss)
## # A tibble: 6 x 65
##   attitudesAndNor~ attitudesAndNor~ attitudesAndNor~ attitudesAndNor~
##              <dbl>            <dbl>            <dbl>            <dbl>
## 1                5                5                5                5
## 2                5                4                2                1
## 3                4                6                5                5
## 4                6                2                3                2
## 5                4                1                2                3
## 6                4                4                4                3
## # ... with 61 more variables: attitudesAndNorms05 <dbl>,
```

```
## #   attitudesAndNorms06 <dbl>, attitudesAndNorms07 <dbl>,
## #   attitudesAndNorms08 <dbl>, callToAction <dbl>,
## #   charitableBehavior01 <dbl>, charitableBehavior02 <dbl>,
## #   descriptiveSocialNorms01 <dbl>, descriptiveSocialNorms02 <dbl>,
## #   descriptiveSocialNorms03 <dbl>, descriptiveSocialNorms04 <dbl>,
## #   mf_AuthoritySubversion <dbl>, mf_CareHarm <dbl>,
## #   mf_FairnessCheating <dbl>, mf_LoyaltyBetrayal <dbl>,
## #   mf_SanctityDegradation <dbl>, moralFoundations01 <dbl>,
## #   moralFoundations02 <dbl>, moralFoundations03 <dbl>,
## #   moralFoundations04 <dbl>, moralFoundations05 <dbl>,
## #   moralFoundations06 <dbl>, moralFoundations07 <dbl>,
## #   moralFoundations08 <dbl>, moralFoundations09 <dbl>,
## #   moralFoundations10 <dbl>, moralFoundations11 <dbl>,
## #   moralFoundations12 <dbl>, moralFoundations13 <dbl>,
## #   moralFoundations14 <dbl>, moralFoundations15 <dbl>,
## #   moralFoundations16 <dbl>, moralFoundations17 <dbl>,
## #   moralFoundations18 <dbl>, moralFoundations19 <dbl>,
## #   moralFoundations20 <dbl>, moralFoundations21 <dbl>,
## #   moralFoundations22 <dbl>, moralFoundations23 <dbl>,
## #   moralFoundations24 <dbl>, moralFoundations25 <dbl>,
## #   moralFoundations26 <dbl>, moralFoundations27 <dbl>,
## #   moralFoundations28 <dbl>, moralFoundations29 <dbl>,
## #   moralFoundations30 <dbl>, moralFoundations31 <dbl>,
## #   moralFoundations32 <dbl>, moralIdentityInternalization01 <dbl>,
## #   moralIdentityInternalization02 <dbl>,
## #   moralIdentityInternalization03 <dbl>,
## #   moralIdentityInternalization04 <dbl>,
## #   moralIdentityInternalization05 <dbl>, pi_age <dbl>,
## #   pi_education <chr>, pi_gender <chr>, pi_ideology <chr>,
## #   pi_income <chr>, pi_nationality <chr>, pi_previousDonations <chr>,
## #   V65 <chr>
```

Kod nekih sudionika, unos pod `pi_education` je razdvojen u dva stupca, pri čemu je jedna vrijednost nasilno gurnuta u `pi_gender`. To je dovelo i do stvaranja nove varijable `V65`, koja sadrži vrijednosti koje bi se trebale javljati pod `pi_previous donations`. Dakle, kod nekih sudionika su vrijednosti iza `pi_education` pomaknute za jedno mjesto udesno.

Do toga je došlo jer je puni naziv jedne razine varijable `pi_education`: "Some professional diploma, no degree". Zbog zareza u nazivu razine dolazi do pogreške u parsanju varijabli, pa dolazi do pomaka udesno i stvaranja varijable viška.

Ovaj problem lako možemo riješiti tako da otvorimo izvornu bazu podataka i samo napravimo find and replace kako bismo uklonili zarez smutnje. Lako ga je riješiti ako imamo SPSS. Druga opcija je korištenje besplatnog online `.sav -> .csv` konvertera (link se nalazi u referencama). Time ćemo dobiti datoteku koju možemo otvoriti u nekom text editoru (recimo, Notepadu), te učiniti potrebne promjene (opet find and replace). Treći način je, naravno, prtljanje po podacima u R-u, što ćemo ostaviti za kraj radionice.

Excel - .xls(x)

Podatke u `.xlsx` (`.xls`) formatu možemo lako učitati pomoću funkcije `read_xlsx` (`read_xls`) iz paketa `readxl`. `readxl` je dio `tidyversea`, ali se ne učitava zajedno njim, tako da ga moramo posebno učitavati.

```
podaci_eksl <- read_xlsx(path = here('podaci', 'podaci_upitnik.xlsx'))

head(podaci_eksl)
```

```
## # A tibble: 6 x 64
##   attitudesAndNor~ attitudesAndNor~ attitudesAndNor~ attitudesAndNor~
##              <dbl>            <dbl>            <dbl>            <dbl>
## 1                5                5                5                5
## 2                5                4                2                1
## 3                4                6                5                5
## 4                6                2                3                2
## 5                4                1                2                3
## 6                4                4                4                3
## # ... with 60 more variables: attitudesAndNorms05 <dbl>,
## #   attitudesAndNorms06 <dbl>, attitudesAndNorms07 <dbl>,
## #   attitudesAndNorms08 <dbl>, callToAction <dbl>,
## #   charitableBehavior01 <dbl>, charitableBehavior02 <dbl>,
## #   descriptiveSocialNorms01 <dbl>, descriptiveSocialNorms02 <dbl>,
## #   descriptiveSocialNorms03 <dbl>, descriptiveSocialNorms04 <dbl>,
## #   mf_AuthoritySubversion <dbl>, mf_CareHarm <dbl>,
## #   mf_FairnessCheating <dbl>, mf_LoyaltyBetrayal <dbl>,
## #   mf_SanctityDegradation <dbl>, moralFoundations01 <dbl>,
## #   moralFoundations02 <dbl>, moralFoundations03 <dbl>,
## #   moralFoundations04 <dbl>, moralFoundations05 <dbl>,
## #   moralFoundations06 <dbl>, moralFoundations07 <dbl>,
## #   moralFoundations08 <dbl>, moralFoundations09 <dbl>,
## #   moralFoundations10 <dbl>, moralFoundations11 <dbl>,
## #   moralFoundations12 <dbl>, moralFoundations13 <dbl>,
## #   moralFoundations14 <dbl>, moralFoundations15 <dbl>,
## #   moralFoundations16 <dbl>, moralFoundations17 <dbl>,
## #   moralFoundations18 <dbl>, moralFoundations19 <dbl>,
## #   moralFoundations20 <dbl>, moralFoundations21 <dbl>,
## #   moralFoundations22 <dbl>, moralFoundations23 <dbl>,
## #   moralFoundations24 <dbl>, moralFoundations25 <dbl>,
## #   moralFoundations26 <dbl>, moralFoundations27 <dbl>,
## #   moralFoundations28 <dbl>, moralFoundations29 <dbl>,
## #   moralFoundations30 <dbl>, moralFoundations31 <dbl>,
## #   moralFoundations32 <dbl>, moralIdentityInternalization01 <dbl>,
## #   moralIdentityInternalization02 <dbl>,
## #   moralIdentityInternalization03 <dbl>,
## #   moralIdentityInternalization04 <dbl>,
## #   moralIdentityInternalization05 <dbl>, pi_age <dbl>,
## #   pi_education <chr>, pi_gender <chr>, pi_ideology <chr>,
## #   pi_income <chr>, pi_nationality <chr>, pi_previousDonations <chr>

str(podaci_eksl)
## Classes 'tbl_df', 'tbl' and 'data.frame':    100 obs. of  64 variables:
##  $ attitudesAndNorms01          : num  5 5 4 6 4 4 6 4 3 5 ...
##  $ attitudesAndNorms02          : num  5 4 6 2 1 4 0 4 7 7 ...
##  $ attitudesAndNorms03          : num  5 2 5 3 2 4 3 5 6 7 ...
##  $ attitudesAndNorms04          : num  5 1 5 2 3 3 3 7 5 6 ...
##  $ attitudesAndNorms05          : num  4 2 3 2 1 4 2 4 4 6 ...
##  $ attitudesAndNorms06          : num  3 2 2 3 2 3 3 3 3 4 ...
##  $ attitudesAndNorms07          : num  4 3 4 5 4 5 6 4 4 5 ...
##  $ attitudesAndNorms08          : num  6 7 5 6 5 5 7 5 3 5 ...
##  $ callToAction                 : num  7 6 7 1 8 7 11 8 3 7 ...
##  $ charitableBehavior01         : num  37 18 7 14 0 37 33 29 16 6 ...
```

```
##  $ charitableBehavior02          : num  4 3 3 5 0 2 4 3 2 3 ...
##  $ descriptiveSocialNorms01      : num  4 3 3 1 3 1 2 4 3 4 ...
##  $ descriptiveSocialNorms02      : num  3 1 3 1 1 1 2 3 3 5 ...
##  $ descriptiveSocialNorms03      : num  2 3 2 2 2 3 3 4 4 5 ...
##  $ descriptiveSocialNorms04      : num  2 1 5 3 4 2 2 2 2 4 ...
##  $ mf_AuthoritySubversion        : num  1 1 2 2 2 0 2 1 1 2 ...
##  $ mf_CareHarm                   : num  3 3 3 3 4 3 4 3 3 4 ...
##  $ mf_FairnessCheating           : num  3 3 4 3 2 4 4 5 3 4 ...
##  $ mf_LoyaltyBetrayal            : num  2 2 2 3 2 1 2 0 0 1 ...
##  $ mf_SanctityDegradation        : num  1 1 1 1 1 -1 1 -1 1 1 ...
##  $ moralFoundations01            : num  4 3 4 3 3 4 5 3 4 4 ...
##  $ moralFoundations02            : num  4 3 4 3 1 4 4 4 2 5 ...
##  $ moralFoundations03            : num  3 0 2 1 1 0 2 0 -1 0 ...
##  $ moralFoundations04            : num  1 0 2 2 2 0 2 0 1 2 ...
##  $ moralFoundations05            : num  2 2 1 3 3 -1 3 1 1 2 ...
##  $ moralFoundations06            : num  0 0 0 2 -1 1 0 0 -1 1 ...
##  $ moralFoundations07            : num  4 3 4 4 5 2 4 4 3 4 ...
##  $ moralFoundations08            : num  4 3 4 3 3 4 5 4 3 5 ...
##  $ moralFoundations09            : num  3 3 2 4 3 3 3 1 -1 1 ...
##  $ moralFoundations10            : num  0 -1 1 3 2 0 2 1 1 1 ...
##  $ moralFoundations11            : num  1 3 1 0 1 -1 2 0 3 3 ...
##  $ moralFoundations12            : num  6 5 4 5 4 4 5 4 3 5 ...
##  $ moralFoundations13            : num  3 5 4 5 5 3 4 5 4 5 ...
##  $ moralFoundations14            : num  4 2 1 1 3 3 3 1 3 2 ...
##  $ moralFoundations15            : num  3 2 2 1 2 3 3 2 5 3 ...
##  $ moralFoundations16            : num  3 1 1 2 -1 1 -2 2 0 1 ...
##  $ moralFoundations17            : num  2 5 3 4 4 4 3 4 3 3 ...
##  $ moralFoundations18            : num  2 3 3 4 5 2 4 5 4 4 ...
##  $ moralFoundations19            : num  0 2 4 2 2 2 4 4 4 2 ...
##  $ moralFoundations20            : num  0 1 0 4 1 3 3 3 2 2 ...
##  $ moralFoundations21            : num  0 1 1 1 3 3 1 2 1 -1 ...
##  $ moralFoundations22            : num  4 4 6 4 4 3 5 3 5 5 ...
##  $ moralFoundations23            : num  3 3 4 2 4 0 3 4 3 3 ...
##  $ moralFoundations24            : num  4 3 1 5 2 3 2 6 2 3 ...
##  $ moralFoundations25            : num  0 0 1 2 0 3 1 2 2 1 ...
##  $ moralFoundations26            : num  1 1 1 5 0 2 2 3 1 1 ...
##  $ moralFoundations27            : num  1 1 0 1 1 1 -1 2 0 1 ...
##  $ moralFoundations28            : num  0 -1 2 -1 -1 1 3 1 4 1 ...
##  $ moralFoundations29            : num  1 1 3 2 4 1 4 2 2 0 ...
##  $ moralFoundations30            : num  1 1 1 1 2 1 1 0 2 2 ...
##  $ moralFoundations31            : num  3 2 1 5 2 2 4 3 3 2 ...
##  $ moralFoundations32            : num  1 0 0 4 2 1 0 1 2 1 ...
##  $ moralIdentityInternalization01: num  5 4 6 6 4 4 5 3 6 5 ...
##  $ moralIdentityInternalization02: num  2 3 5 4 3 6 5 2 4 5 ...
##  $ moralIdentityInternalization03: num  1 1 1 1 1 1 1 1 1 1 ...
##  $ moralIdentityInternalization04: num  2 3 1 3 2 1 3 3 3 1 ...
##  $ moralIdentityInternalization05: num  3 4 5 4 4 4 5 3 4 5 ...
##  $ pi_age                        : num  3 20 20 19 22 25 23 41 16 17 ...
##  $ pi_education                  : chr  "Some professional diploma, no degree" "Master's degree" "Hi
##  $ pi_gender                     : chr  "Male" "Male" "Male" "Male" ...
##  $ pi_ideology                   : chr  "Neither liberal or conservative" "Very liberal (left)" "Nei
##  $ pi_income                     : chr  "Somewhat below the average" "Somewhat above the average" "S
##  $ pi_nationality                : chr  "American" "USA" "Turkish" "United States of America" ...
```

```
##  $ pi_previousDonations          : chr  "Rarely" "Regularly" "Rarely" "Rarely" ...
```

Comma separated values - .csv

Comma separeted value datoteke su točno to što ime kaže - podaci koji su strukturirani kao vrijednosti odvojene zarezima, gdje se svaki unos (na primjer sudionik) nalazi u zasebnom redu, a vrijednosti varijabli koje su uz njega povezane ispisane su redom i odvojene su zarezima. U prvom redu (koji funkcije u R-u često nazivaju **header**) obično se nalaze imena varijabli, a u ostalim redovima su njihove vrijednosti.

Ovako izgledaju prva dva reda i prvih nekoliko stupaca datoteke `podaci_upitnik.csv`:

```
attitudesAndNorms01,attitudesAndNorms02,attitudesAndNorms03, ...
5,5,5,5,4, ...
```

Podatke u `.csv` formatu možemo učitati pomoću funkcije `read_csv` iz `readr` paketa (koji je automatski učitan kad smo učitali `tidyverse`). Osnovni (base) R ima funkciju `read.csv` koja obavlja isti zadatak, ali neki R developeri preporučuju korištenje `read_csv` funkcije (na primjer, Hadley Wickham i Garret Grolemund: http://r4ds.had.co.nz/import.html). U skladu s tom preporukom, koristit ćemo `read_csv`. Podatke iz datoteke `podaci_upitnik.csv` možemo učitati ovako:

```
podaci <- read_csv(here('podaci', 'podaci_upitnik.csv'))
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   pi_education = col_character(),
##   pi_gender = col_character(),
##   pi_ideology = col_character(),
##   pi_income = col_character(),
##   pi_nationality = col_character(),
##   pi_previousDonations = col_character()
## )
## See spec(...) for full column specifications.
```

Poruka koju dobivamo obavještava nas o tome kako su određene varijable reprezentirane. Vidimo da su varijable koje počinju s `pi` reprezentirane kao `character`. Ako pozovemo funkciju `spec`, vidjet ćemo specifikacije svih varijabli. Budući da pozivanjem funkcije `str` zapravo dobivamo manje-više iste podatke, pozvat ćemo samo nju. Njen output pomoći će nam da vidimo jesu li podaci reprezentirani onako kako bismo očekivali.

```
str(podaci)
## Classes 'tbl_df', 'tbl' and 'data.frame':    100 obs. of  64 variables:
##  $ attitudesAndNorms01          : int  5 5 4 6 4 4 6 4 3 5 ...
##  $ attitudesAndNorms02          : int  5 4 6 2 1 4 0 4 7 7 ...
##  $ attitudesAndNorms03          : int  5 2 5 3 2 4 3 5 6 7 ...
##  $ attitudesAndNorms04          : int  5 1 5 2 3 3 3 7 5 6 ...
##  $ attitudesAndNorms05          : int  4 2 3 2 1 4 2 4 4 6 ...
##  $ attitudesAndNorms06          : int  3 2 2 3 2 3 3 3 3 4 ...
##  $ attitudesAndNorms07          : int  4 3 4 5 4 5 6 4 4 5 ...
##  $ attitudesAndNorms08          : int  6 7 5 6 5 5 7 5 3 5 ...
##  $ callToAction                 : int  7 6 7 1 8 7 11 8 3 7 ...
##  $ charitableBehavior01         : int  37 18 7 14 0 37 33 29 16 6 ...
##  $ charitableBehavior02         : int  4 3 3 5 0 2 4 3 2 3 ...
##  $ descriptiveSocialNorms01     : int  4 3 3 1 3 1 2 4 3 4 ...
##  $ descriptiveSocialNorms02     : int  3 1 3 1 1 1 2 3 3 5 ...
##  $ descriptiveSocialNorms03     : int  2 3 2 2 2 3 3 4 4 5 ...
```

```
##  $ descriptiveSocialNorms04      : int  2 1 5 3 4 2 2 2 2 4 ...
##  $ mf_AuthoritySubversion        : int  1 1 2 2 2 0 2 1 1 2 ...
##  $ mf_CareHarm                   : int  3 3 3 3 4 3 4 3 3 4 ...
##  $ mf_FairnessCheating           : int  3 3 4 3 2 4 4 5 3 4 ...
##  $ mf_LoyaltyBetrayal            : int  2 2 2 3 2 1 2 0 0 1 ...
##  $ mf_SanctityDegradation        : int  1 1 1 1 1 -1 1 -1 1 1 ...
##  $ moralFoundations01            : int  4 3 4 3 3 4 5 3 4 4 ...
##  $ moralFoundations02            : int  4 3 4 3 1 4 4 4 2 5 ...
##  $ moralFoundations03            : int  3 0 2 1 1 0 2 0 -1 0 ...
##  $ moralFoundations04            : int  1 0 2 2 2 0 2 0 1 2 ...
##  $ moralFoundations05            : int  2 2 1 3 3 -1 3 1 1 2 ...
##  $ moralFoundations06            : int  0 0 0 2 -1 1 0 0 -1 1 ...
##  $ moralFoundations07            : int  4 3 4 4 5 2 4 4 3 4 ...
##  $ moralFoundations08            : int  4 3 4 3 3 4 5 4 3 5 ...
##  $ moralFoundations09            : int  3 3 2 4 3 3 3 1 -1 1 ...
##  $ moralFoundations10            : int  0 -1 1 3 2 0 2 1 1 1 ...
##  $ moralFoundations11            : int  1 3 1 0 1 -1 2 0 3 3 ...
##  $ moralFoundations12            : int  6 5 4 5 4 4 5 4 3 5 ...
##  $ moralFoundations13            : int  3 5 4 5 5 3 4 5 4 5 ...
##  $ moralFoundations14            : int  4 2 1 1 3 3 3 1 3 2 ...
##  $ moralFoundations15            : int  3 2 2 1 2 3 3 2 5 3 ...
##  $ moralFoundations16            : int  3 1 1 2 -1 1 -2 2 0 1 ...
##  $ moralFoundations17            : int  2 5 3 4 4 4 3 4 3 3 ...
##  $ moralFoundations18            : int  2 3 3 4 5 2 4 5 4 4 ...
##  $ moralFoundations19            : int  0 2 4 2 2 2 4 4 4 2 ...
##  $ moralFoundations20            : int  0 1 0 4 1 3 3 3 2 2 ...
##  $ moralFoundations21            : int  0 1 1 1 3 3 1 2 1 -1 ...
##  $ moralFoundations22            : int  4 4 6 4 4 3 5 3 5 5 ...
##  $ moralFoundations23            : int  3 3 4 2 4 0 3 4 3 3 ...
##  $ moralFoundations24            : int  4 3 1 5 2 3 2 6 2 3 ...
##  $ moralFoundations25            : int  0 0 1 2 0 3 1 2 2 1 ...
##  $ moralFoundations26            : int  1 1 1 5 0 2 2 3 1 1 ...
##  $ moralFoundations27            : int  1 1 0 1 1 1 -1 2 0 1 ...
##  $ moralFoundations28            : int  0 -1 2 -1 -1 1 3 1 4 1 ...
##  $ moralFoundations29            : int  1 1 3 2 4 1 4 2 2 0 ...
##  $ moralFoundations30            : int  1 1 1 1 2 1 1 0 2 2 ...
##  $ moralFoundations31            : int  3 2 1 5 2 2 4 3 3 2 ...
##  $ moralFoundations32            : int  1 0 0 4 2 1 0 1 2 1 ...
##  $ moralIdentityInternalization01: int  5 4 6 6 4 4 5 3 6 5 ...
##  $ moralIdentityInternalization02: int  2 3 5 4 3 6 5 2 4 5 ...
##  $ moralIdentityInternalization03: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ moralIdentityInternalization04: int  2 3 1 3 2 1 3 3 3 1 ...
##  $ moralIdentityInternalization05: int  3 4 5 4 4 4 5 3 4 5 ...
##  $ pi_age                        : int  3 20 20 19 22 25 23 41 16 17 ...
##  $ pi_education                  : chr  "Some professional diploma, no degree" "Master's degree" "Hi
##  $ pi_gender                     : chr  "Male" "Male" "Male" "Male" ...
##  $ pi_ideology                   : chr  "Neither liberal or conservative" "Very liberal (left)" "Nei
##  $ pi_income                     : chr  "Somewhat below the average" "Somewhat above the average" "S
##  $ pi_nationality                : chr  "American" "USA" "Turkish" "United States of America" ...
##  $ pi_previousDonations          : chr  "Rarely" "Regularly" "Rarely" "Rarely" ...
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 64
##   .. ..$ attitudesAndNorms01          : list()
```

```
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ attitudesAndNorms02          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ attitudesAndNorms03          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ attitudesAndNorms04          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ attitudesAndNorms05          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ attitudesAndNorms06          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ attitudesAndNorms07          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ attitudesAndNorms08          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ callToAction                 : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ charitableBehavior01         : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ charitableBehavior02         : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ descriptiveSocialNorms01     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ descriptiveSocialNorms02     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ descriptiveSocialNorms03     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ descriptiveSocialNorms04     : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ mf_AuthoritySubversion       : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ mf_CareHarm                  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ mf_FairnessCheating          : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ mf_LoyaltyBetrayal           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ mf_SanctityDegradation       : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations01           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations02           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations03           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations04           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations05           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations06           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations07           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
```

```
##   .. ..$ moralFoundations08           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations09           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations10           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations11           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations12           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations13           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations14           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations15           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations16           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations17           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations18           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations19           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations20           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations21           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations22           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations23           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations24           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations25           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations26           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations27           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations28           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations29           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations30           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations31           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralFoundations32           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralIdentityInternalization01: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralIdentityInternalization02: list()
```

```
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralIdentityInternalization03: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralIdentityInternalization04: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ moralIdentityInternalization05: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ pi_age                       : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ pi_education                 : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ pi_gender                    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ pi_ideology                  : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ pi_income                    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ pi_nationality               : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ pi_previousDonations         : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

Obrisat ćemo specifikacije varijabli da ne zakrčuju output. `attr` nam omogućuje da pristupimo raznim **atributima** objekata u R-u. Ovdje, dakle, pristupamo atributu `spec` objekta `podaci`, te ga brišemo upisujući vrijednost `NULL`.

```
attr(podaci, 'spec') <- NULL

head(podaci)
## # A tibble: 6 x 64
##   attitudesAndNor~ attitudesAndNor~ attitudesAndNor~ attitudesAndNor~
##              <int>            <int>            <int>            <int>
## 1                5                5                5                5
## 2                5                4                2                1
## 3                4                6                5                5
## 4                6                2                3                2
## 5                4                1                2                3
## 6                4                4                4                3
## # ... with 60 more variables: attitudesAndNorms05 <int>,
## #   attitudesAndNorms06 <int>, attitudesAndNorms07 <int>,
## #   attitudesAndNorms08 <int>, callToAction <int>,
## #   charitableBehavior01 <int>, charitableBehavior02 <int>,
## #   descriptiveSocialNorms01 <int>, descriptiveSocialNorms02 <int>,
## #   descriptiveSocialNorms03 <int>, descriptiveSocialNorms04 <int>,
## #   mf_AuthoritySubversion <int>, mf_CareHarm <int>,
## #   mf_FairnessCheating <int>, mf_LoyaltyBetrayal <int>,
## #   mf_SanctityDegradation <int>, moralFoundations01 <int>,
## #   moralFoundations02 <int>, moralFoundations03 <int>,
## #   moralFoundations04 <int>, moralFoundations05 <int>,
## #   moralFoundations06 <int>, moralFoundations07 <int>,
## #   moralFoundations08 <int>, moralFoundations09 <int>,
```

```
## #   moralFoundations10 <int>, moralFoundations11 <int>,
## #   moralFoundations12 <int>, moralFoundations13 <int>,
## #   moralFoundations14 <int>, moralFoundations15 <int>,
## #   moralFoundations16 <int>, moralFoundations17 <int>,
## #   moralFoundations18 <int>, moralFoundations19 <int>,
## #   moralFoundations20 <int>, moralFoundations21 <int>,
## #   moralFoundations22 <int>, moralFoundations23 <int>,
## #   moralFoundations24 <int>, moralFoundations25 <int>,
## #   moralFoundations26 <int>, moralFoundations27 <int>,
## #   moralFoundations28 <int>, moralFoundations29 <int>,
## #   moralFoundations30 <int>, moralFoundations31 <int>,
## #   moralFoundations32 <int>, moralIdentityInternalization01 <int>,
## #   moralIdentityInternalization02 <int>,
## #   moralIdentityInternalization03 <int>,
## #   moralIdentityInternalization04 <int>,
## #   moralIdentityInternalization05 <int>, pi_age <int>,
## #   pi_education <chr>, pi_gender <chr>, pi_ideology <chr>,
## #   pi_income <chr>, pi_nationality <chr>, pi_previousDonations <chr>

tail(podaci, 3)
## # A tibble: 3 x 64
##   attitudesAndNor~ attitudesAndNor~ attitudesAndNor~ attitudesAndNor~
##              <int>            <int>            <int>            <int>
## 1                5                5                2                3
## 2                4                5                3                2
## 3                7                2                1               -1
## # ... with 60 more variables: attitudesAndNorms05 <int>,
## #   attitudesAndNorms06 <int>, attitudesAndNorms07 <int>,
## #   attitudesAndNorms08 <int>, callToAction <int>,
## #   charitableBehavior01 <int>, charitableBehavior02 <int>,
## #   descriptiveSocialNorms01 <int>, descriptiveSocialNorms02 <int>,
## #   descriptiveSocialNorms03 <int>, descriptiveSocialNorms04 <int>,
## #   mf_AuthoritySubversion <int>, mf_CareHarm <int>,
## #   mf_FairnessCheating <int>, mf_LoyaltyBetrayal <int>,
## #   mf_SanctityDegradation <int>, moralFoundations01 <int>,
## #   moralFoundations02 <int>, moralFoundations03 <int>,
## #   moralFoundations04 <int>, moralFoundations05 <int>,
## #   moralFoundations06 <int>, moralFoundations07 <int>,
## #   moralFoundations08 <int>, moralFoundations09 <int>,
## #   moralFoundations10 <int>, moralFoundations11 <int>,
## #   moralFoundations12 <int>, moralFoundations13 <int>,
## #   moralFoundations14 <int>, moralFoundations15 <int>,
## #   moralFoundations16 <int>, moralFoundations17 <int>,
## #   moralFoundations18 <int>, moralFoundations19 <int>,
## #   moralFoundations20 <int>, moralFoundations21 <int>,
## #   moralFoundations22 <int>, moralFoundations23 <int>,
## #   moralFoundations24 <int>, moralFoundations25 <int>,
## #   moralFoundations26 <int>, moralFoundations27 <int>,
## #   moralFoundations28 <int>, moralFoundations29 <int>,
## #   moralFoundations30 <int>, moralFoundations31 <int>,
## #   moralFoundations32 <int>, moralIdentityInternalization01 <int>,
## #   moralIdentityInternalization02 <int>,
## #   moralIdentityInternalization03 <int>,
```

```
## #   moralIdentityInternalization04 <int>,
## #   moralIdentityInternalization05 <int>, pi_age <int>,
## #   pi_education <chr>, pi_gender <chr>, pi_ideology <chr>,
## #   pi_income <chr>, pi_nationality <chr>, pi_previousDonations <chr>
```

Ove funkcije pomažu nam pri pregledavanju strukture podataka i njihovih sirovih vrijednosti. Osnovnu deskriptivnu statistiku možemo dobiti pomoću generičke funkcije `summary`. Generičke funkcije primaju objekte različitih tipova, a njihov output ovisi o tipu objekta. Primjerice, ako u `summary` stavimo `data.frame`, dobit ćemo grubu deskriptivnu statistiku njegovih stupaca. Ako u funkciju stavimo regresijski model, dobit ćemo informacije o modelu. Idemo vidjeti output tih dviju funkcija kad u nju stavimo neke numeričke i neke kategorijalne stupce iz našeg `data.framea` `podaci`.

```
summary(podaci[, wrapr::qc(attitudesAndNorms01, pi_education, pi_gender)])
##  attitudesAndNorms01 pi_education        pi_gender
##  Min.   :2.00        Length:100         Length:100
##  1st Qu.:4.00        Class :character   Class :character
##  Median :5.00        Mode  :character   Mode  :character
##  Mean   :5.04
##  3rd Qu.:6.00
##  Max.   :8.00
```

Vidimo tri stvari: (1) `summary` nije pretjerano koristan za varijable koje su tipa `character` i (2-3) pojavili su se nova sintaksa i nova funkcija.

`qc` je funkcija iz paketa `wrapr` koja nas oslobađa pisanja navodnika pri korištenju funkcije `c`. `qc` je, dakle, *quoted combine*.

Korištenjem `::` sintakse označili smo da je funkcija `qc` iz paketa `wrapr`. Pri pozivanju funkcija iz paketa **nije nužno** pisati `::`; to smo vidjeli kod pozivanja funkcije `read_csv` iz paketa `readr` (ili `read_xls` ili `read_spss`).

Ipak, važno je znati tu sintaksu iz dva razloga.

Prvo, korištenjem `::` možemo pozvati funkciju iz paketa koji prethodno nismo učitali.

Drugo, u slučaju da dva paketa imaju funkcije koje se jednako zovu, `::` nam omogućava da specificiramo koju funkciju želimo pozvati. Budući da smo učitali paket `conflicted`, R će nas upozoriti ako dođe do konflikta te nas tražiti da specificiramo koju funkciju hoćemo pozvati, koristeći `::`.

Iskoristit ćemo trenutak i upoznati se s još jednom zgodnom funkcijom za dobivanje deskriptivnih podataka: `skim`. Radi preglednosti, probrat ćemo par varijabli različitih tipova.

Nastavit ću koristiti `::` notaciju tako da bude jasno iz kojeg paketa dolazi koja funkcija (osim ako je spomenuto u tekstu ili ako je funkcija iz base R-a).

```
print(skimr::skim(podaci[, qc(pi_education, attitudesAndNorms01,
                              attitudesAndNorms02, attitudesAndNorms03,
                              mf_CareHarm, pi_income)]))
```

Ovdje smo iskoristili `::` notaciju da bismo pozvali funkciju iz paketa koji ranije nije učitan. Dobro je znati i za funkciju `describe` iz paketa `psych`, koja daje nešto detaljniju deskriptivnu statistiku numeričkih varijabli.

```
podaci %>%
dplyr::select(., attitudesAndNorms01:attitudesAndNorms03,
              mf_CareHarm, pi_income, pi_education) %>%
psych::describe(.) %>% print(.)
## Warning in psych::describe(.): NAs introduced by coercion

## Warning in psych::describe(.): NAs introduced by coercion
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning
## Inf
## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning
## -Inf
##                       vars   n mean   sd median trimmed  mad min  max range
## attitudesAndNorms01      1 100 5.04 1.43      5    5.06 1.48   2    8     6
## attitudesAndNorms02      2 100 3.22 2.03      3    3.27 1.48  -2    7     9
## attitudesAndNorms03      3 100 3.06 1.88      3    3.05 1.48  -2    8    10
## mf_CareHarm              4 100 3.56 0.83      3    3.50 1.48   2    6     4
## pi_income*               5 100  NaN   NA     NA     NaN   NA Inf -Inf  -Inf
## pi_education*            6 100  NaN   NA     NA     NaN   NA Inf -Inf  -Inf
##                        skew kurtosis   se
## attitudesAndNorms01 -0.19    -0.59 0.14
## attitudesAndNorms02 -0.17    -0.47 0.20
## attitudesAndNorms03  0.02     0.12 0.19
## mf_CareHarm          0.54     0.23 0.08
## pi_income*             NA       NA   NA
## pi_education*          NA       NA   NA
```

I dalje nije korisno za `character` varijable, ali omogućava digresiju u svijet pipa.

Pipe

Pipe su posebni operatori iz `magrittr` paketa. One omogućavaju kraće i, često, razumljivije pisanje koda. Pipa uzima output izraza sa svoje lijeve strane i daje ga kao argument funkciji na svojoj desnoj strani. Osnovna pipa je `%>%`. Ona se nalazi i u paketu `dplyr` (koji se učitava kad učitamo `tidyverse`) i u paketu `magrittr`. Posebno smo učitali `magrittr` jer s njim dolaze i neke pipe kojih nema u `dplyru`. Sad ćemo proći kroz pipe koje `magrittr` nudi.

%>%

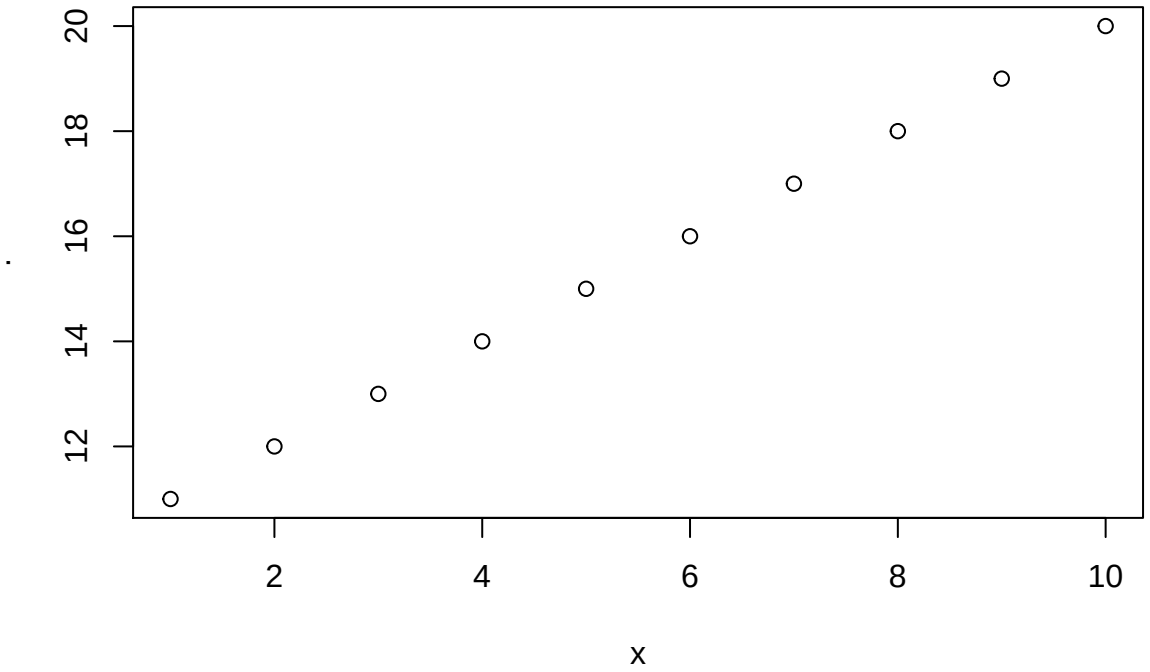
Kao što je rečeno, ovo je osnovna pipa. Ona uzima output izraza s lijeve strane i koristi ga kao input za izraz s desne strane.

Dakle:

```
(2 + 2) %>% sqrt(.)
## [1] 2

(2 + 2) %>% sqrt()
## [1] 2

(2 + 2) %>% sqrt
## [1] 2

(2 + 2)^(1/2)
## [1] 2
```

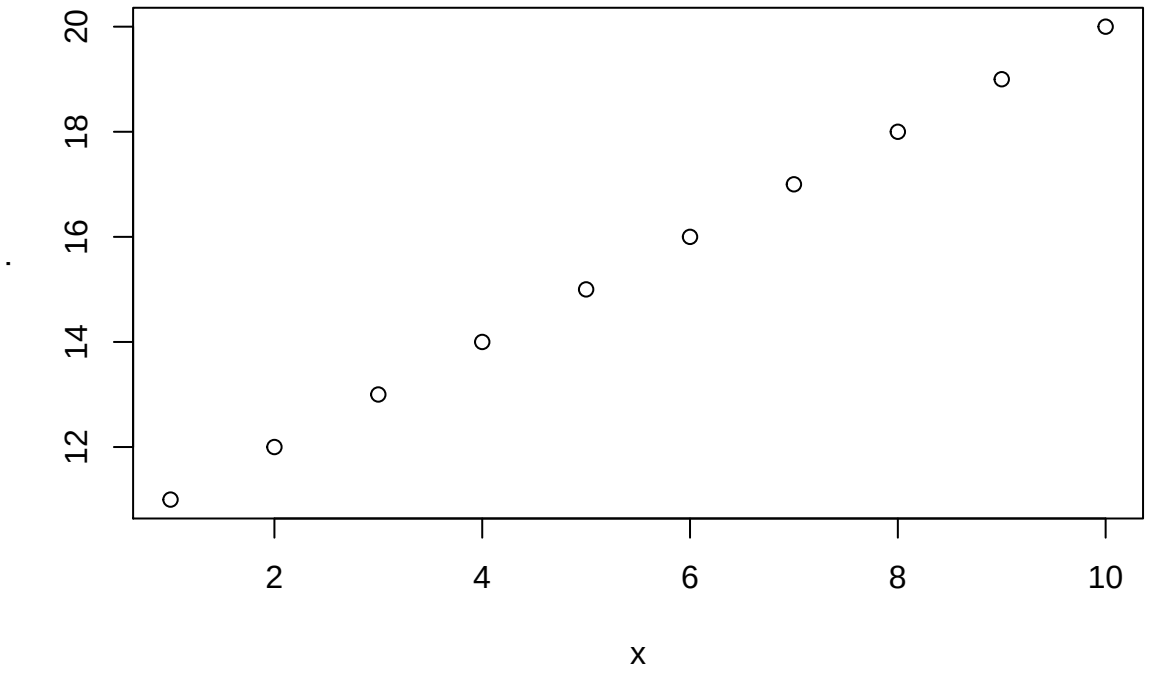
Trenutačno ne izgleda kao neka ušteda, što je u redu. Kasnije ćemo vidjeti primjere u kojima su pipe dosta zgodnije. Kod korištenja pipa, `.` označava output iz funkcije s lijeve strane. Po defaultu, ako `.` nije eksplicitno navedena, pipa će točku staviti na mjesto prvog argumenta. Takav default uglavnom jako dobro funkcionira s funkcijama iz `tidyversea` jer one imaju dosta uniformnu sintaksu, koja je prilagođena za pipe. Ipak, nekad takvo ponašanje nije poželjno. Pokušajmo grafirati dva brojčana vektora - jedan od njih ćemo spremiti u varijablu, a drugi ćemo direktno dati pipi.

```
# ova vrijednost treba ići na x os
x <- 1:10

# vektor koji ćemo sad stvoriti treba ići na y os
# ovaj kod jednak je ovom -> 11:20 %>% plot(., x)
# plot je također generička funkcija
11:20 %>% plot(x,.)
```



Da bismo izbjegli takvo ponašanje, možemo eksplicitno napisati točku, ili izraz opkoliti vitičastim zagradama.

```
11:20 %>% {plot(x, .)}
```

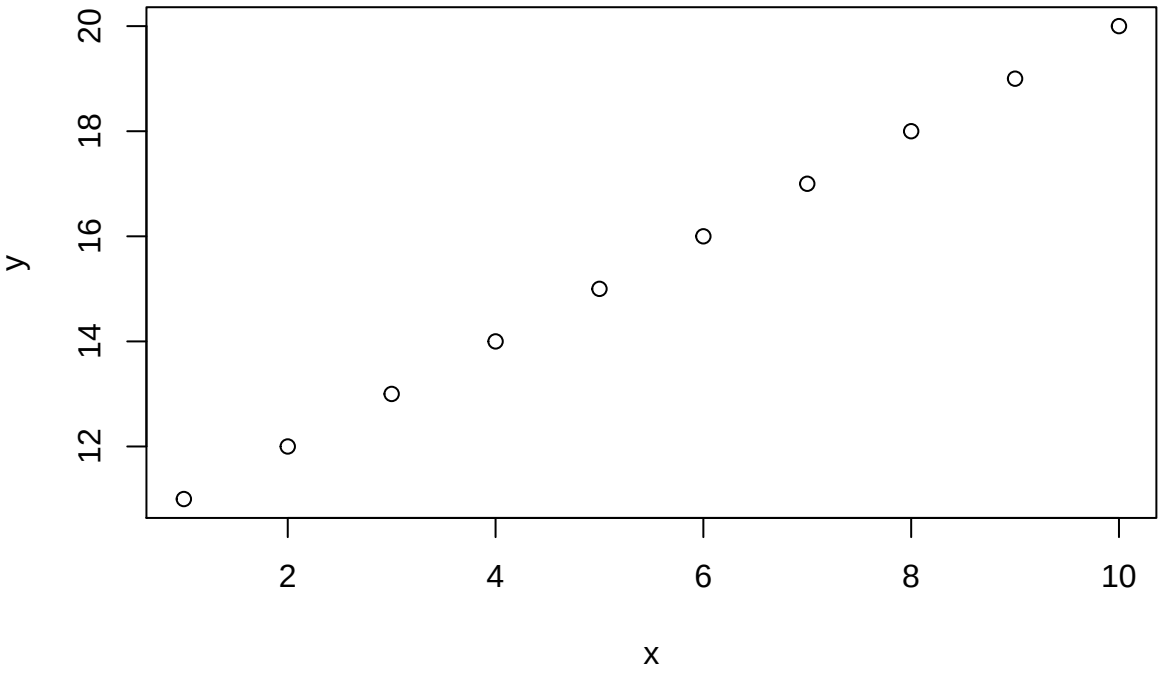


Sad ćemo `x` staviti u `data.frame` i pridružiti mu `y`.

```
za_graf <- data.frame(x = 1:10, y = 11:20)

str(za_graf)
## 'data.frame':    10 obs. of  2 variables:
##  $ x: int  1 2 3 4 5 6 7 8 9 10
##  $ y: int  11 12 13 14 15 16 17 18 19 20
```

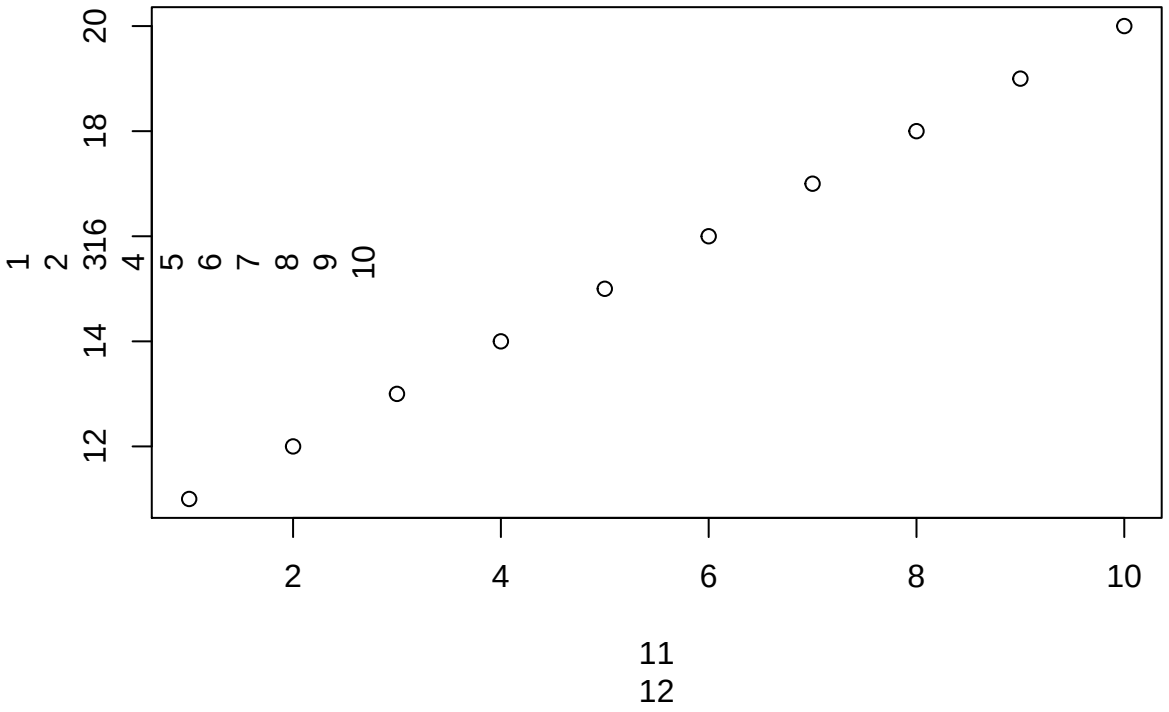
Ponovno ćemo pokušati plotati vrijednosti tako što `za_graf` stavimo u pipu.

```
za_graf %>% plot(.)
```



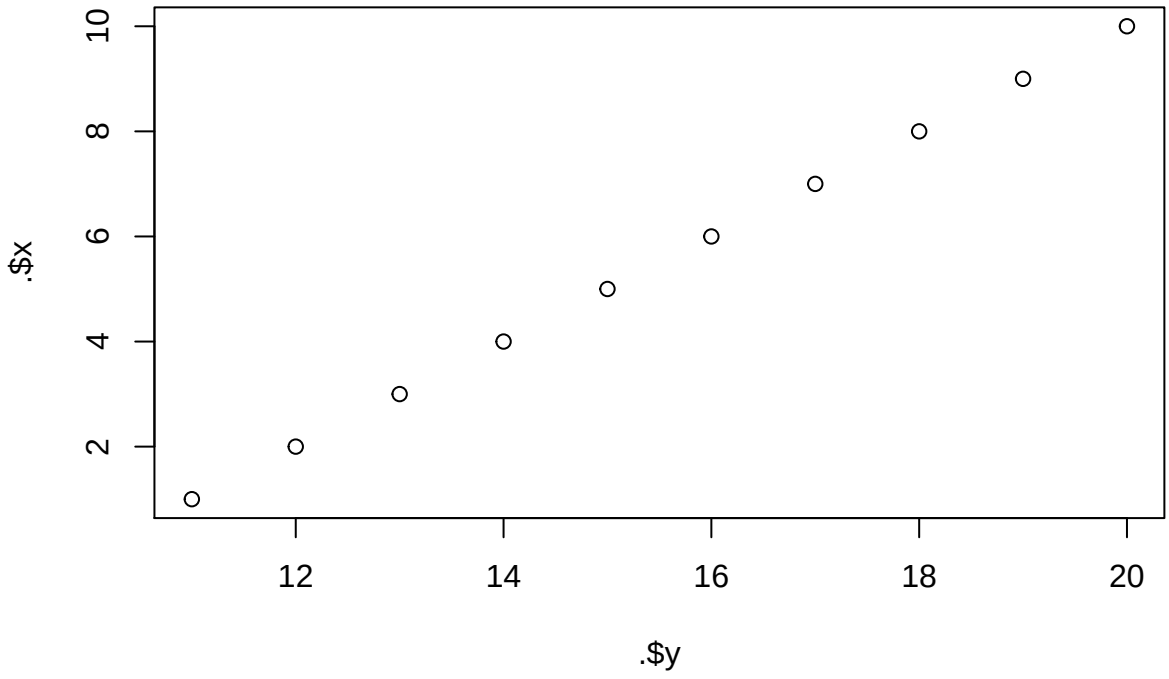
Da bismo zamijenili osi, možemo učiniti sljedeće:

```
za_graf %>% plot(.$y, .$x)
```



To jest, ne možemo jer se gornji kod interpretira kao `za_graf %>% plot(., .$y, .$x)` Dakle, ponovno možemo izraz s desne strane opkoliti vitičastim zagradama.

```
za_graf %>% {plot(.$y, .$x)}
```

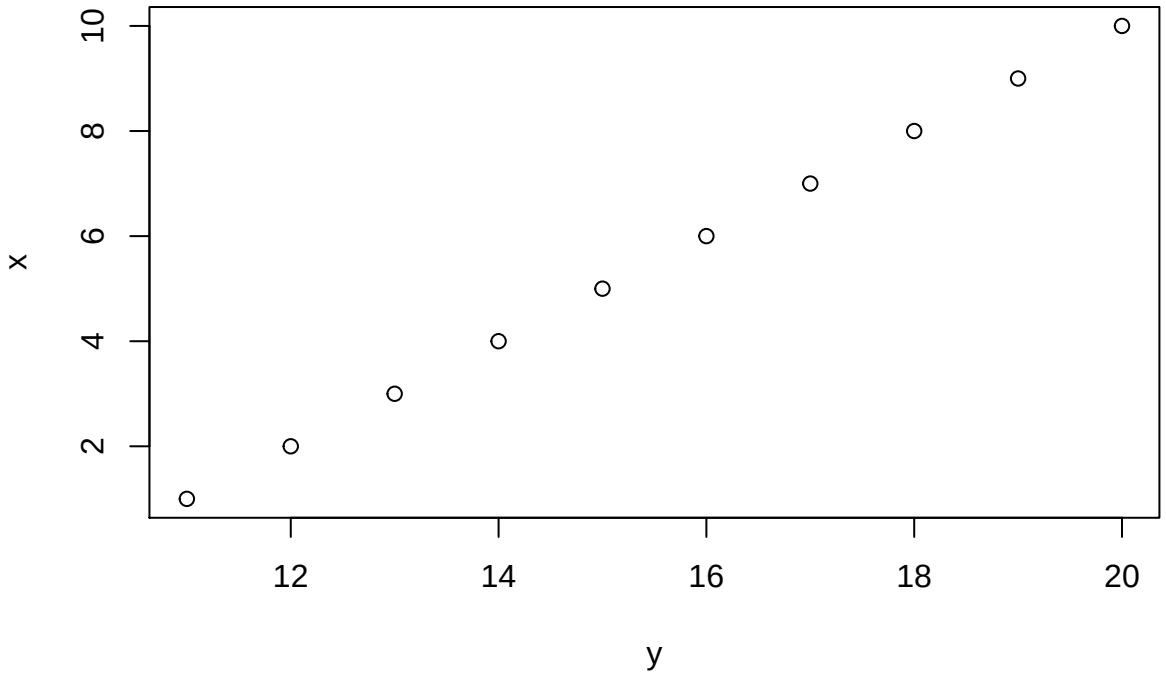


Da se smanje takve konfuzije, neki preporučuju da se `.` uvijek piše, tako da je to praksa koju ćemo ovdje usvojiti. No, osim zatvaranja izraza s desne strane u zagrade, možemo iskoristiti jednu drugu pipu.

%$%

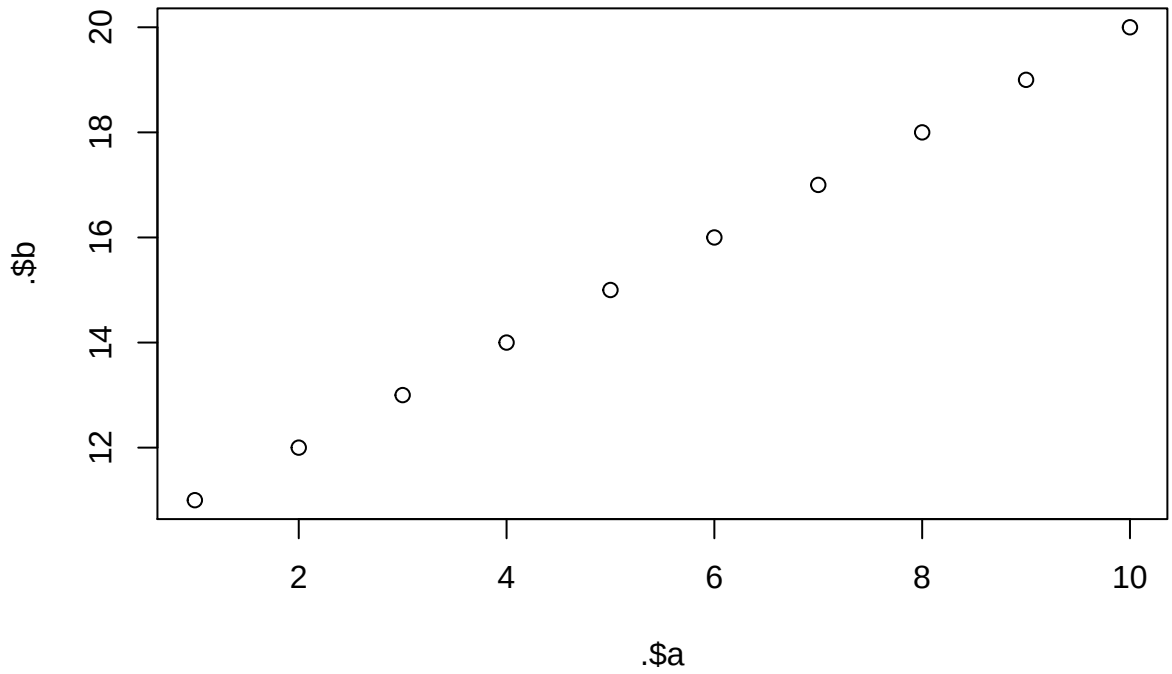
%$% je *variable exposition* pipa. Ona nam daje direktan pristup varijablama koje se nalaze u objektu kojim baratamo. Gornji problem mogli bismo riješiti i ovako:

```
za_graf %$% plot(y, x)
```



Možemo kombinirati različite pipe. Na primjer:

```
1:10 %>% data.frame (a = ., b = 11:20) %>% {plot(y = .$b, x = .$a)}
```



U gornjem primjeru bi nam možda bilo zgodno da možemo pogledati strukturu `data.framea` nakon što ga

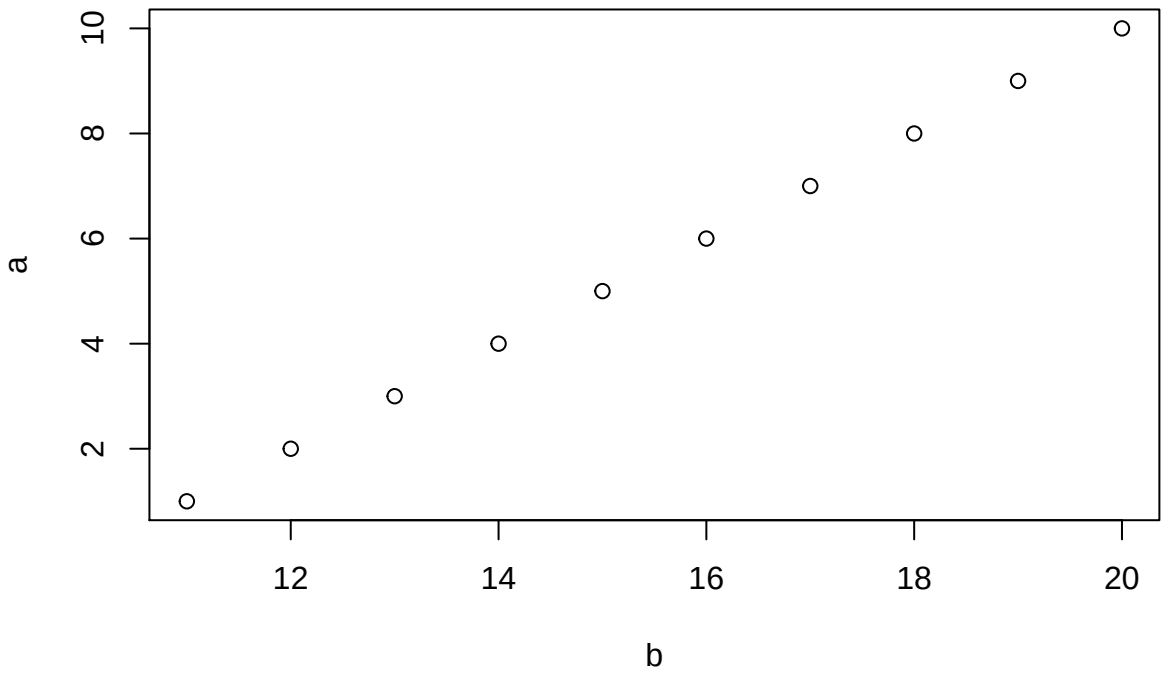
stvorimo ili napraviti još neke operacije nakon što plotamo varijable.

Imamo pipu i za to.

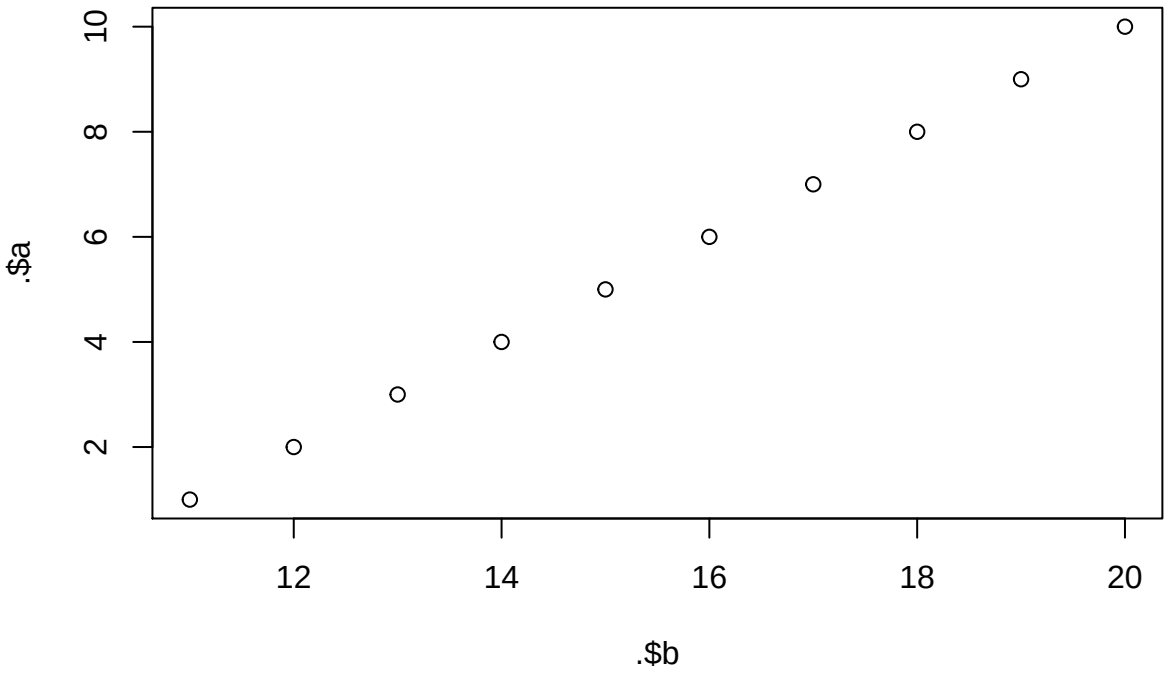
%T>%

T-pipa (izvorno *tee*) vraća izraz s lijeve strane umjesto izraza s desne strane. Zbog toga, možemo učiniti sljedeće:

```
1:10 %>% data.frame(a = ., b = 11:20) %T>% str(.) %$% plot(b, a)
## 'data.frame':    10 obs. of  2 variables:
##  $ a: int  1 2 3 4 5 6 7 8 9 10
##  $ b: int  11 12 13 14 15 16 17 18 19 20
```




```
1:10 %>% data.frame(a = ., b = 11:20) %T>%
{plot(.$b, .$a)} %$%
sum(a,b)
```




```
## [1] 210
```

Preostaje nam još jedna pipa...

%<>%

%<>% je *assignment* pipa. Ona istovremeno daje vrijednost s lijeve strane za argument i piše u nju. To nam omogućuje da neku varijablu provučemo kroz seriju transformacijskih koraka i da te transformacije odmah pohranimo.

```
str(za_graf)
## 'data.frame':    10 obs. of  2 variables:
##  $ x: int  1 2 3 4 5 6 7 8 9 10
##  $ y: int  11 12 13 14 15 16 17 18 19 20

za_graf$x %<>% magrittr::add(., 2) %>%
magrittr::multiply_by(., 2) %>% sqrt(.)

str(za_graf)
## 'data.frame':    10 obs. of  2 variables:
##  $ x: num  2.45 2.83 3.16 3.46 3.74 ...
##  $ y: int  11 12 13 14 15 16 17 18 19 20
```

Ovime završavamo upoznavanje s pipama. Nakratko se vraćamo natrag na primjer s funkcijom `describe`, nakon čega ponovno odlazimo u uzbudljivu digresiju.

dplyr::select i dplyr::filter

Već smo ranije vidjeli funkciju `select`, koja nam je omogućila da izaberemo 3 od 64 stupca iz `data.framea podaci`. Za odabiranje pojedinih **redova** koji zadovoljavaju određeni logički izraz možemo koristiti funkciju `filter`.

Sad ćemo prikazati deskriptivnu statistiku za pitanja koja tvore jednu od skala koja se nalazi u našim podacima - skalu internalizacije moralnog identiteta - samo na poduzorku žena. Sve varijable koje se odnose na tu skalu imaju ime oblika `moralIdentityInternalization<broj-pitanja>`. Zbog tog sustavnog imenovanja, ne moramo ispisivati imena (ili redne brojeve) svih varijabli koje želimo zahvatiti funkcijom `describe`, nego možemo pozvati funkciju `contains` unutar funkcije `select`. `contains` na omogućuje da odaberemo samo one varijable koje sadrže zadani string.

```
podaci %>%
dplyr::filter(., pi_gender == 'Female') %>%
dplyr::select(., dplyr::contains('Internal',
                                 # igore.case govori treba li
                                 # poštivati ili ignorirati
                                 # malo/veliko slovo
                                 ignore.case = T)) %T>% str(.) %>%
psych::describe(.)
## Classes 'tbl_df', 'tbl' and 'data.frame':    45 obs. of  5 variables:
##  $ moralIdentityInternalization01: int  4 5 5 4 4 5 4 6 6 5 ...
##  $ moralIdentityInternalization02: int  3 5 5 3 3 3 3 5 6 3 ...
##  $ moralIdentityInternalization03: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ moralIdentityInternalization04: int  2 3 1 2 4 2 2 1 1 2 ...
##  $ moralIdentityInternalization05: int  4 5 5 4 2 4 4 6 6 5 ...
##                                  vars  n mean   sd median trimmed  mad min
## moralIdentityInternalization01    1 45 4.71 0.66      5    4.68 0.00   3
## moralIdentityInternalization02    2 45 4.18 1.03      4    4.08 1.48   3
## moralIdentityInternalization03    3 45 1.00 0.00      1    1.00 0.00   1
## moralIdentityInternalization04    4 45 1.49 0.87      1    1.46 1.48   0
## moralIdentityInternalization05    5 45 4.49 0.92      4    4.51 1.48   2
##                                  max range  skew kurtosis   se
## moralIdentityInternalization01   6     3 -0.09    -0.26 0.10
## moralIdentityInternalization02   7     4  0.63    -0.28 0.15
## moralIdentityInternalization03   1     0   NaN      NaN 0.00
## moralIdentityInternalization04   4     4  0.34     0.20 0.13
## moralIdentityInternalization05   6     4 -0.22    -0.16 0.14

# base R rješenje za usporedbu
str(podaci[podaci$pi_gender == 'Female', qc(moralIdentityInternalization01,
                                            moralIdentityInternalization02,
                                            moralIdentityInternalization03,
                                            moralIdentityInternalization04,
                                            moralIdentityInternalization05)])
## Classes 'tbl_df', 'tbl' and 'data.frame':    45 obs. of  5 variables:
##  $ moralIdentityInternalization01: int  4 5 5 4 4 5 4 6 6 5 ...
##  $ moralIdentityInternalization02: int  3 5 5 3 3 3 3 5 6 3 ...
##  $ moralIdentityInternalization03: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ moralIdentityInternalization04: int  2 3 1 2 4 2 2 1 1 2 ...
##  $ moralIdentityInternalization05: int  4 5 5 4 2 4 4 6 6 5 ...
psych::describe(podaci[podaci$pi_gender == 'Female', qc(moralIdentityInternalization01,
                                            moralIdentityInternalization02,
                                            moralIdentityInternalization03,
                                            moralIdentityInternalization04,
                                            moralIdentityInternalization05)])
##                                  vars  n mean   sd median trimmed  mad min
## moralIdentityInternalization01    1 45 4.71 0.66      5    4.68 0.00   3
## moralIdentityInternalization02    2 45 4.18 1.03      4    4.08 1.48   3
```

```
## moralIdentityInternalization03    3 45 1.00 0.00      1    1.00 0.00   1
## moralIdentityInternalization04    4 45 1.49 0.87      1    1.46 1.48   0
## moralIdentityInternalization05    5 45 4.49 0.92      4    4.51 1.48   2
##                                 max range  skew kurtosis   se
## moralIdentityInternalization01   6     3 -0.09    -0.26 0.10
## moralIdentityInternalization02   7     4  0.63    -0.28 0.15
## moralIdentityInternalization03   1     0   NaN      NaN 0.00
## moralIdentityInternalization04   4     4  0.34     0.20 0.13
## moralIdentityInternalization05   6     4 -0.22    -0.16 0.14
```

`contains` je jedna od nekoliko pomoćnih funkcija koje su super za `select`. Druge su:

- `starts_with`, koja odabire varijable koje počinju s određenim stringom
- `ends_with`, isto, samo za kraj
- `one_of`, koju treba koristiti kad `selectu` dajemo `character` vektor; na primjer, ako imena varijabli koje želimo zahvatiti spremimo u varijablu
- `matches`, koji nam omogućava da odaberemo varijable čija imena odgovaraju nekom **regularnom izrazu**

Regularni izrazi

Regularni izrazi (eng. *regular expressions*, *regex* ili *regexp*) su stringovi koji označavaju neki uzorak za pretraživanje. Na primjer, sve ove izraze

```
string
striing
striiing
striiiiiiiiiiiiiiiing
```

možemo opisati stringom `stri*ng`. Znak `*` (asterisk) je **kvantifikator** koji označava *nula ili više ponavljanja prethodnog znaka*. To znači da bi taj regularni izraz pronašao i string `strng`. Uz razne kvantifikatore, postoje još i klase znakova te meta-znakovi koji nam omogućavaju lako pretraživanje stringova. Regexi su implementirani u base R-u (npr. funkcije `grep` i `grepl`) i u `tidyverseu` kroz paket `stringr`. Mi ćemo se baviti `stringrom`. Budući da postoje razne implementacije regularnih izraza, koje se razlikuju po kompleksnosti, bitno je znati da `stringr` koristi **Perl/PCRE** regularne izraze. U ovom dijelu ćemo pogledati osnove regularnih izraza, koje ćemo nadograđivati kroz ostatak radionice.

Kvantifikatori

*

Kao što je već rečeno, `*` označava **0 ili više** ponavljanja **znaka** koji mu prethodi. *Znak* se ovdje odnosi na doslovni znak, na klasu znakova ili na grupu znakova. S klasama i grupama ćemo se upoznati malo kasnije. Pogledat ćemo output funkcije `str_detect` koja kao input uzima string (ili više njih) i regularni izraz (`pattern`), a vraća `TRUE` ili `FALSE` ovisno o tome nalazi li se regularni izraz u stringu ili ne.

```
stringr::str_detect(string = qc(kobilaaaa, maajka, celer), pattern = 'a*') %>% print(.)
## [1] TRUE TRUE TRUE
```

+

`+` označava **jedno (1) ili više** ponavljanja prethodnog znaka/klase znakova/grupe znakova. Da vidimo što će nam vratiti funkcija `str_extract_all` koja prima iste argumente kao i `str_detect`, a vraća sve pronađene `patterne`.

```
stringr::str_extract_all(string = qc(kobilaaaa, maajka, celer), pattern = 'a+') %>% print(.)
## [[1]]
## [1] "aaaa"
##
## [[2]]
## [1] "aa" "a"
##
## [[3]]
## character(0)
```

Postoji i funkcija `str_extract` koja vraća **samo prvi** pronađeni uzorak.

```
stringr::str_extract(qc(kobilaaaa, maajka, celer), 'a+') %>% print(.)
## [1] "aaaa" "aa"   NA
```

Također, možemo vidjeti da `str_detect` više ne vraća `TRUE` za posljednju riječ.

```
stringr::str_detect(qc(kobilaaaa, maajka, celer), 'a+') %>% print(.)
## [1]  TRUE  TRUE FALSE
```

?

Upitnik označavao **0 ili jedno (1)** ponavljanje.

```
qc(kobilaaaa, maajka, celer) %>%
stringr::str_extract_all(., 'a?') %>%
print(.)
## [[1]]
##  [1] ""  ""  ""  ""  ""  "a" "a" "a" "a" ""
##
## [[2]]
## [1] ""  "a" "a" ""  ""  "a" ""
##
## [[3]]
## [1] "" "" "" "" "" ""
```

{n,m}

Ova sintaksa nam omogućava da specificiramo koliko ponavljanja želimo. Postoje tri valjane kombinacije:

- `{n,m}` znači od `n` do `m`
- `{n,}` znači `n` ili više
- `{n}` znači točno `n`

`{,m}` **nije valjan** regularni izraz! Također, bitno je da nema razmaka između `n` ili `m` i zareza. Vratit ćemo se na početni primjer.

```
qc(string, striing, striiing, striiiiiiiiiiiiiiiiiing) %>%
stringr::str_extract_all(., 'i{2,5}') %>% print(.)
## [[1]]
## character(0)
##
## [[2]]
## [1] "ii"
##
## [[3]]
```

```
## [1] "iii"
##
## [[4]]
## [1] "iiiiii" "iiiiii" "iiiiii" "ii"

qc(string, striing, striiing, striiiiiiiiiiiiiiiiing) %>%
stringr::str_extract_all(., 'i{3,}') %>% print(.)
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## [1] "iii"
##
## [[4]]
## [1] "iiiiiiiiiiiiiiiii"

qc(string, striing, striiing, striiiiiiiiiiiiiiiiing) %>%
stringr::str_extract_all(., 'i{17}') %>% print(.)
## [[1]]
## character(0)
##
## [[2]]
## character(0)
##
## [[3]]
## character(0)
##
## [[4]]
## [1] "iiiiiiiiiiiiiiiii"
```

Klase znakova

Pretraživanja koja smo dosad vidjeli su jednostavna i jako umjetna. U stvarnim primjenama uglavnom nećemo pokušavati uhvatiti jedno slovo, nego znakove određenog tipa (kao što su brojke) ili određene skupine znakova (npr. brojeve 1, 7 ili 5). U te svrhe, koristimo **klase znakova**.

NB: Klase znakova predstavljaju više mogućih znakova, ali **samo jedno mjesto**. Napravit ćemo mali `data.frame` koji se sastoji od dva stupca koja sadrže stringove.

```
# ne možemo koristiti qc za mjesta zbog razmaka
data.frame(mjesta = c('Slavonski Brod', 'BJELOVAR', 'Cista Provo', 'Banova Jaruga'),
           tablice = qc(SB1152KF, BJ302LD, CP999LO, BN2001KA)) -> registracije
```

Za početak, pokušat ćemo pronaći sva mjesta čija se imena sastoje od dvije riječi (to znači da ćemo isključiti BJELOVAR :(). Vidimo da sva mjesta koja se sastoje od dvije riječi imaju sljedeći uzorak: `[veliko slovo][nekoliko malih slova][razmak][veliko slovo][nekoliko malih slova]`. Koristeći regexe, možemo napraviti sljedeće:

```
print(registracije)
##           mjesta  tablice
## 1 Slavonski Brod SB1152KF
```

```
## 2       BJELOVAR  BJ302LD
## 3    Cista Provo  CP999LO
## 4  Banova Jaruga BN2001KA

registracije$mjesta %>%
stringr::str_detect(., '^[[:upper:]][[:lower:]]+\\s[[:upper:]][[:lower:]]+$')
## [1]  TRUE FALSE  TRUE  TRUE
```

`^` (eng. *caret*) je meta-znak koji označava **početak stringa**. `[[:upper:]]` i `[[:lower:]]` su klase koje označavaju velika odnosno mala slova. `\\s` označava razmak (ostavljanje praznog mjesta također funkcionira). Dakle, obrazac koji tražimo mora počinjati s velikim slovom kojem slijedi jedno ili više malih slova.

Drugi važan meta-znak je `$`, koji označava **kraj stringa**. NB: Ako želimo tražiti same meta-znakove (npr. u `$1551`), ispred njih moramo staviti `\\` (backslash x 2). Taj čin se zove *escaping*.

```
# qc ni ovdje ne funkcionira
c('$alaj', '€broj') %>%
stringr::str_detect(., '\\$')
## [1]  TRUE FALSE
```

Koristeći uglate zagrade, možemo sami definirati klasu znakova koja je prihvatljiva na nekom mjestu. Na primjer, možemo tražiti sva mjesta koja imaju dvije riječi i čija prva riječ počinje slovom B (velikim!) ili S (također!). Ovdje ćemo koristiti `str_subset`, koja vraća stringove koji sadrže zadani obrazac.

```
registracije$mjesta %>%
stringr::str_subset(., '^[SB][[:lower:]]+\\s[[:upper:]][[:lower:]]+')
## [1] "Slavonski Brod" "Banova Jaruga"
```

Možemo definirati i custom klasu znakova koji se **ne smiju** nalaziti na nekom mjestu. To radimo tako da na početak svoje klase stavimo znak `^` (`[^...]`).

NB: Ovdje treba obratiti pažnju na pozicioniranje znaka `-`. On mora dolaziti **odmah nakon otvarajuće** zagrade (ili iza `^` ako imamo negacijskju klasu) ili **neposredno prije** zatvarajuće zagrade. Pomoću znaka `-` možemo definirati raspone (npr. `0-9`), pa možemo dobiti error ili nešto neočekivano ako ga stavimo usred klase. Na primjer, možemo tražiti stringove koji ne počinju slovom S ili B:

```
registracije$mjesta %>%
stringr::str_subset(., '^[^SB].*')
## [1] "Cista Provo"
```

Točka je poseban znak u regularnim izrazima, a označava **bilo koji znak** (osim novog reda, što se u R-u označava s `\\n`). Budući da označava bilo što, `.` se zove *wildcard*. Klasa znakova ima razmjerno puno, pa ćemo spomenuti još jednu koja se često javlja. Pokušat ćemo izvući samo one registracijske oznake (`tablice`) koje imaju tri znamenke.

```
registracije$tablice %>%
stringr::str_subset(., '[[:upper:]]{2}\\d{3}[[:upper:]]')
## [1] "BJ302LD" "CP999LO"
```

`\\d`, dakle, označava znamenke. Zasad ćemo proći još samo kroz grupe znakova.

Grupe znakova

Znakove možemo grupirati koristeći obične zagrade (`(...)`). Grupe spajaju znakove u jednu cjelinu. To nam, primjerice, omogućuje da ponavljajuće uzorke lako kvantificiramo.

Na primjer, zamislimo da želimo izvući određene vrste smjehova iz nekih stringova.

```
qc(hehehe, hehahohohehe, hahahahihi) %>%
stringr::str_extract_all(., '(ha|he){2}') %>%
print(.)
## [[1]]
## [1] "hehe"
##
## [[2]]
## [1] "heha" "hehe"
##
## [[3]]
## [1] "haha"
```

Ovdje smo iskoristili i znak `|` (kod mene se nalazi na `AltGr-W` i zove se *pipe*), koji označava alternaciju, odnosno logičko ILI. Dakle, tražimo dva ponavljanja stringa `ha` ili `he`.

NB: Ne stavljati razmake oko alternatora jer će se to tumačiti kao razmak koji treba tražiti u stringu!

Vježbica

Radili smo longitudinalno istraživanje s dvije točke mjerenja. Svojim dragim sudionicima napisali smo jednostavnu formulu za stvaranje šifre: prva dva slova imena majke, posljednje dvije znamenke mobitela i prva dva slova imena rodnog grada.

Sve smo ih stavili u format pogodan za nekakvo analiziranje longitudinalnih podataka, zbog čega se šifre sudionika iz obje točke mjerenja nalaze u jednom stupcu. Ovo su šifre naša 4 sudionika:

```
sifre <- qc(BR83ZA, KA15ZA, RA75BJ, PE43SP,
            BR83ZG, ZA15KA, RA75BJ, PE43ST)
```

Koristeći moći opažanja, uočili smo da:

- su neki sudionici u drugoj točki mjerenja umjesto prva dva slova imena rodnog grada pisali registarsku oznaku rodnog grada
- je jedan sudionik zamijenio mjesto prvih slova imena majke i prvih slova imena rodnog grada. Pokušajte (i) izvući sve sudionike čiji je rodni grad Zagreb ili Split te (ii) izvući sve šifre sudionika koji je zamijenio redoslijed imena majke i slova rodnog grada. Napišite potpuni regularni izraz (dakle, nema švercanja s `.*`)!

```
sifre %>%
stringr::str_detect(., '[[:upper:]]{2}\\d{2}(ZG|ST)')
## [1] FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE

sifre %>%
stringr::str_subset(., '(ZA|KA)\\d{2}(ZA|KA)')
## [1] "KA15ZA" "ZA15KA"
```

Time završavamo digresivne tokove i bacamo se na borbu s podacima.

Nastavak pripreme podataka

Zasad smo pogledali strukturu podatka (`str`), kako izgledaju sirovi podaci (`head` i `tail`) te neke statističke sažetke (`describe` i `summary`, `skim`).

Sad ćemo se baciti na formatiranje sirovih podataka u nešto što nam je zgodnije za rad. Prvo ćemo se prisjetiti strukture podatka kojima baratamo.

```
str(podaci)
## Classes 'tbl_df', 'tbl' and 'data.frame':    100 obs. of  64 variables:
##  $ attitudesAndNorms01          : int  5 5 4 6 4 4 6 4 3 5 ...
##  $ attitudesAndNorms02          : int  5 4 6 2 1 4 0 4 7 7 ...
##  $ attitudesAndNorms03          : int  5 2 5 3 2 4 3 5 6 7 ...
##  $ attitudesAndNorms04          : int  5 1 5 2 3 3 3 7 5 6 ...
##  $ attitudesAndNorms05          : int  4 2 3 2 1 4 2 4 4 6 ...
##  $ attitudesAndNorms06          : int  3 2 2 3 2 3 3 3 3 4 ...
##  $ attitudesAndNorms07          : int  4 3 4 5 4 5 6 4 4 5 ...
##  $ attitudesAndNorms08          : int  6 7 5 6 5 5 7 5 3 5 ...
##  $ callToAction                 : int  7 6 7 1 8 7 11 8 3 7 ...
##  $ charitableBehavior01         : int  37 18 7 14 0 37 33 29 16 6 ...
##  $ charitableBehavior02         : int  4 3 3 5 0 2 4 3 2 3 ...
##  $ descriptiveSocialNorms01     : int  4 3 3 1 3 1 2 4 3 4 ...
##  $ descriptiveSocialNorms02     : int  3 1 3 1 1 1 2 3 3 5 ...
##  $ descriptiveSocialNorms03     : int  2 3 2 2 2 3 3 4 4 5 ...
##  $ descriptiveSocialNorms04     : int  2 1 5 3 4 2 2 2 2 4 ...
##  $ mf_AuthoritySubversion       : int  1 1 2 2 2 0 2 1 1 2 ...
##  $ mf_CareHarm                  : int  3 3 3 3 4 3 4 3 3 4 ...
##  $ mf_FairnessCheating          : int  3 3 4 3 2 4 4 5 3 4 ...
##  $ mf_LoyaltyBetrayal           : int  2 2 2 3 2 1 2 0 0 1 ...
##  $ mf_SanctityDegradation       : int  1 1 1 1 1 -1 1 -1 1 1 ...
##  $ moralFoundations01           : int  4 3 4 3 3 4 5 3 4 4 ...
##  $ moralFoundations02           : int  4 3 4 3 1 4 4 4 2 5 ...
##  $ moralFoundations03           : int  3 0 2 1 1 0 2 0 -1 0 ...
##  $ moralFoundations04           : int  1 0 2 2 2 0 2 0 1 2 ...
##  $ moralFoundations05           : int  2 2 1 3 3 -1 3 1 1 2 ...
##  $ moralFoundations06           : int  0 0 0 2 -1 1 0 0 -1 1 ...
##  $ moralFoundations07           : int  4 3 4 4 5 2 4 4 3 4 ...
##  $ moralFoundations08           : int  4 3 4 3 3 4 5 4 3 5 ...
##  $ moralFoundations09           : int  3 3 2 4 3 3 3 1 -1 1 ...
##  $ moralFoundations10           : int  0 -1 1 3 2 0 2 1 1 1 ...
##  $ moralFoundations11           : int  1 3 1 0 1 -1 2 0 3 3 ...
##  $ moralFoundations12           : int  6 5 4 5 4 4 5 4 3 5 ...
##  $ moralFoundations13           : int  3 5 4 5 5 3 4 5 4 5 ...
##  $ moralFoundations14           : int  4 2 1 1 3 3 3 1 3 2 ...
##  $ moralFoundations15           : int  3 2 2 1 2 3 3 2 5 3 ...
##  $ moralFoundations16           : int  3 1 1 2 -1 1 -2 2 0 1 ...
##  $ moralFoundations17           : int  2 5 3 4 4 4 3 4 3 3 ...
##  $ moralFoundations18           : int  2 3 3 4 5 2 4 5 4 4 ...
##  $ moralFoundations19           : int  0 2 4 2 2 2 4 4 4 2 ...
##  $ moralFoundations20           : int  0 1 0 4 1 3 3 3 2 2 ...
##  $ moralFoundations21           : int  0 1 1 1 3 3 1 2 1 -1 ...
##  $ moralFoundations22           : int  4 4 6 4 4 3 5 3 5 5 ...
##  $ moralFoundations23           : int  3 3 4 2 4 0 3 4 3 3 ...
##  $ moralFoundations24           : int  4 3 1 5 2 3 2 6 2 3 ...
##  $ moralFoundations25           : int  0 0 1 2 0 3 1 2 2 1 ...
##  $ moralFoundations26           : int  1 1 1 5 0 2 2 3 1 1 ...
##  $ moralFoundations27           : int  1 1 0 1 1 1 -1 2 0 1 ...
##  $ moralFoundations28           : int  0 -1 2 -1 -1 1 3 1 4 1 ...
##  $ moralFoundations29           : int  1 1 3 2 4 1 4 2 2 0 ...
##  $ moralFoundations30           : int  1 1 1 1 2 1 1 0 2 2 ...
##  $ moralFoundations31           : int  3 2 1 5 2 2 4 3 3 2 ...
```

```
##  $ moralFoundations32           : int  1 0 0 4 2 1 0 1 2 1 ...
##  $ moralIdentityInternalization01: int  5 4 6 6 4 4 5 3 6 5 ...
##  $ moralIdentityInternalization02: int  2 3 5 4 3 6 5 2 4 5 ...
##  $ moralIdentityInternalization03: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ moralIdentityInternalization04: int  2 3 1 3 2 1 3 3 3 1 ...
##  $ moralIdentityInternalization05: int  3 4 5 4 4 4 5 3 4 5 ...
##  $ pi_age                       : int  3 20 20 19 22 25 23 41 16 17 ...
##  $ pi_education                 : chr  "Some professional diploma, no degree" "Master's degree" "Hi
##  $ pi_gender                    : chr  "Male" "Male" "Male" "Male" ...
##  $ pi_ideology                  : chr  "Neither liberal or conservative" "Very liberal (left)" "Nei
##  $ pi_income                    : chr  "Somewhat below the average" "Somewhat above the average" "S
##  $ pi_nationality               : chr  "American" "USA" "Turkish" "United States of America" ...
##  $ pi_previousDonations         : chr  "Rarely" "Regularly" "Rarely" "Rarely" ...
```

Za početak, iskoristit ćemo moći opažanja i primijetiti da su varijable koje počinju s `pi` (osim `pi_age`) spremljene kao `character` vektori. Taj tip vrijednosti nije zgodan za većinu obrada koje bismo mogli htjeti raditi i razlog je zašto nam `summary` vraća nekoristan sažetak.

Baratanje kategoričkim varijablama

Stoga, pretvorit ćemo te varijable iz `charactera` u `factore`. Varijable možemo modificirati koristeći `mutate` obitelj funkcija. Ovdje ćemo iskoristiti `mutate_at`, koji nam omogućuje da specificiramo varijable na koje želimo primijeniti neku funkciju.

Uhvatit ćemo sve `pi` varijable osim `pi_age` te na njih primijeniti funkciju `as.factor`, koja će ih pretvoriti u `factore`. Budući da će `mutate_at` zadanu funkciju primijeniti na postojeće stupce, dobro je (a) uvjeriti se da biramo prave stupce i (b) uvjeriti se da radimo ono što želimo raditi prije nego što spremimo promjene. (a) ćemo riješiti koristeći `colnames` i `select`.

```
podaci %>%
dplyr::select(., dplyr::starts_with('pi'), -pi_age) %>%
colnames(.)
## [1] "pi_education"         "pi_gender"            "pi_ideology"
## [4] "pi_income"            "pi_nationality"       "pi_previousDonations"
```

Vidimo da ciljamo ispravne stupce. Sad možemo eksperimentirati s `mutate_at`.

```
podaci %>%
dplyr::mutate_at(.,
                 # varijable koje želimo zahvatiti treba omotati u
                 # funkciju vars; ona prima iste pomoćne funkcije kao
                 # i select
                 .vars = dplyr::vars(dplyr::starts_with('pi'), -pi_age),
                 .fun = as.factor) %>%
# ovaj dio je samo radi prikazivanja
dplyr::select(., starts_with('pi')) %>%
str(.)
## Classes 'tbl_df', 'tbl' and 'data.frame':    100 obs. of  7 variables:
##  $ pi_age              : int  3 20 20 19 22 25 23 41 16 17 ...
##  $ pi_education        : Factor w/ 6 levels "Elementary School",..: 5 3 2 3 3 3 5 3 6 5 ...
##  $ pi_gender           : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 1 2 2 1 ...
##  $ pi_ideology         : Factor w/ 7 levels "Extremely conservative (right)",..: 3 7 3 7 7 7 5 5 3 7
##  $ pi_income           : Factor w/ 5 levels "About the average",..: 5 4 4 4 4 4 4 4 4 5 ...
##  $ pi_nationality      : Factor w/ 16 levels "American","Asian american",..: 1 15 9 12 13 15 1 14 7 
##  $ pi_previousDonations: Factor w/ 4 levels "Never","Often",..: 3 4 3 3 4 4 4 2 2 3 ...
```

Zadovoljni smo outputom, pa možemo malko modificirati kod i spremiti promjene.

```
podaci %<>%
dplyr::mutate_at(.,
                 .vars = dplyr::vars(dplyr::starts_with('pi'), -pi_age),
                 .fun = as.factor)

str(podaci)
## Classes 'tbl_df', 'tbl' and 'data.frame':    100 obs. of  64 variables:
##  $ attitudesAndNorms01           : int  5 5 4 6 4 4 6 4 3 5 ...
##  $ attitudesAndNorms02           : int  5 4 6 2 1 4 0 4 7 7 ...
##  $ attitudesAndNorms03           : int  5 2 5 3 2 4 3 5 6 7 ...
##  $ attitudesAndNorms04           : int  5 1 5 2 3 3 3 7 5 6 ...
##  $ attitudesAndNorms05           : int  4 2 3 2 1 4 2 4 4 6 ...
##  $ attitudesAndNorms06           : int  3 2 2 3 2 3 3 3 3 4 ...
##  $ attitudesAndNorms07           : int  4 3 4 5 4 5 6 4 4 5 ...
##  $ attitudesAndNorms08           : int  6 7 5 6 5 5 7 5 3 5 ...
##  $ callToAction                  : int  7 6 7 1 8 7 11 8 3 7 ...
##  $ charitableBehavior01          : int  37 18 7 14 0 37 33 29 16 6 ...
##  $ charitableBehavior02          : int  4 3 3 5 0 2 4 3 2 3 ...
##  $ descriptiveSocialNorms01      : int  4 3 3 1 3 1 2 4 3 4 ...
##  $ descriptiveSocialNorms02      : int  3 1 3 1 1 1 2 3 3 5 ...
##  $ descriptiveSocialNorms03      : int  2 3 2 2 2 3 3 4 4 5 ...
##  $ descriptiveSocialNorms04      : int  2 1 5 3 4 2 2 2 2 4 ...
##  $ mf_AuthoritySubversion        : int  1 1 2 2 2 0 2 1 1 2 ...
##  $ mf_CareHarm                   : int  3 3 3 3 4 3 4 3 3 4 ...
##  $ mf_FairnessCheating           : int  3 3 4 3 2 4 4 5 3 4 ...
##  $ mf_LoyaltyBetrayal            : int  2 2 2 3 2 1 2 0 0 1 ...
##  $ mf_SanctityDegradation        : int  1 1 1 1 1 -1 1 -1 1 1 ...
##  $ moralFoundations01            : int  4 3 4 3 3 4 5 3 4 4 ...
##  $ moralFoundations02            : int  4 3 4 3 1 4 4 4 2 5 ...
##  $ moralFoundations03            : int  3 0 2 1 1 0 2 0 -1 0 ...
##  $ moralFoundations04            : int  1 0 2 2 2 0 2 0 1 2 ...
##  $ moralFoundations05            : int  2 2 1 3 3 -1 3 1 1 2 ...
##  $ moralFoundations06            : int  0 0 0 2 -1 1 0 0 -1 1 ...
##  $ moralFoundations07            : int  4 3 4 4 5 2 4 4 3 4 ...
##  $ moralFoundations08            : int  4 3 4 3 3 4 5 4 3 5 ...
##  $ moralFoundations09            : int  3 3 2 4 3 3 3 1 -1 1 ...
##  $ moralFoundations10            : int  0 -1 1 3 2 0 2 1 1 1 ...
##  $ moralFoundations11            : int  1 3 1 0 1 -1 2 0 3 3 ...
##  $ moralFoundations12            : int  6 5 4 5 4 4 5 4 3 5 ...
##  $ moralFoundations13            : int  3 5 4 5 5 3 4 5 4 5 ...
##  $ moralFoundations14            : int  4 2 1 1 3 3 3 1 3 2 ...
##  $ moralFoundations15            : int  3 2 2 1 2 3 3 2 5 3 ...
##  $ moralFoundations16            : int  3 1 1 2 -1 1 -2 2 0 1 ...
##  $ moralFoundations17            : int  2 5 3 4 4 4 3 4 3 3 ...
##  $ moralFoundations18            : int  2 3 3 4 5 2 4 5 4 4 ...
##  $ moralFoundations19            : int  0 2 4 2 2 2 4 4 4 2 ...
##  $ moralFoundations20            : int  0 1 0 4 1 3 3 3 2 2 ...
##  $ moralFoundations21            : int  0 1 1 1 3 3 1 2 1 -1 ...
##  $ moralFoundations22            : int  4 4 6 4 4 3 5 3 5 5 ...
##  $ moralFoundations23            : int  3 3 4 2 4 0 3 4 3 3 ...
##  $ moralFoundations24            : int  4 3 1 5 2 3 2 6 2 3 ...
```

```
##  $ moralFoundations25            : int  0 0 1 2 0 3 1 2 2 1 ...
##  $ moralFoundations26            : int  1 1 1 5 0 2 2 3 1 1 ...
##  $ moralFoundations27            : int  1 1 0 1 1 1 -1 2 0 1 ...
##  $ moralFoundations28            : int  0 -1 2 -1 -1 1 3 1 4 1 ...
##  $ moralFoundations29            : int  1 1 3 2 4 1 4 2 2 0 ...
##  $ moralFoundations30            : int  1 1 1 1 2 1 1 0 2 2 ...
##  $ moralFoundations31            : int  3 2 1 5 2 2 4 3 3 2 ...
##  $ moralFoundations32            : int  1 0 0 4 2 1 0 1 2 1 ...
##  $ moralIdentityInternalization01: int  5 4 6 6 4 4 5 3 6 5 ...
##  $ moralIdentityInternalization02: int  2 3 5 4 3 6 5 2 4 5 ...
##  $ moralIdentityInternalization03: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ moralIdentityInternalization04: int  2 3 1 3 2 1 3 3 3 1 ...
##  $ moralIdentityInternalization05: int  3 4 5 4 4 4 5 3 4 5 ...
##  $ pi_age                        : int  3 20 20 19 22 25 23 41 16 17 ...
##  $ pi_education                  : Factor w/ 6 levels "Elementary School",..: 5 3 2 3 3 3 5 3 6 5 ..
##  $ pi_gender                     : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 1 2 2 1 ...
##  $ pi_ideology                   : Factor w/ 7 levels "Extremely conservative (right)",..: 3 7 3 7 7
##  $ pi_income                     : Factor w/ 5 levels "About the average",..: 5 4 4 4 4 4 4 4 4 5 ..
##  $ pi_nationality                : Factor w/ 16 levels "American","Asian american",..: 1 15 9 12 13 
##  $ pi_previousDonations          : Factor w/ 4 levels "Never","Often",..: 3 4 3 3 4 4 4 2 2 3 ...
```

Ako sad pozovemo `summary`, dobit ćemo korisnije rezultate.

```
podaci %>%
dplyr::select(., dplyr::starts_with('pi_'), -pi_age) %>%
summary(.)
##                              pi_education  pi_gender
##  Elementary School                  : 1    Female:45
##  High school                        :23    Male  :55
##  Master's degree                    :24
##  PhD or higher                      : 1
##  Some professional diploma, no degree:19
##  The baccalaureate                  :32
##
##                          pi_ideology                      pi_income
##  Extremely conservative (right) : 1   About the average         :17
##  Extremely liberal (left)       :17   Much above the average    : 9
##  Neither liberal or conservative:12   Much below the average    : 8
##  Somewhat conservative (right)  :10   Somewhat above the average:47
##  Somewhat liberal (left)        :24   Somewhat below the average:19
##  Very conservative (right)      : 2
##  Very liberal (left)            :34
##                   pi_nationality pi_previousDonations
##  American                :24     Never    : 8
##  USA                     :24     Often    :31
##  Canadian                :13     Rarely   :40
##  British                 :11     Regularly:21
##  United States           : 9
##  united states of america: 3
##  (Other)                 :16
```

Gledajući output ove funkcije, primjećujemo da su pojedine vrijednosti prilično dugačke (npr. Some professional diploma, no degree). Koristeći `forcats` paket (dio `tidyversea`), vrlo lako možemo rekodirati te vrijednosti. Za početak, da bismo si uskratili nešto tipkanja, možemo pozvati funkciju `dput` kako bismo dobili reprezentaciju

razina faktora koju možemo kopipejstati.

```
podaci$pi_education %>% levels(.) %>% dput(.)
## c("Elementary School", "High school", "Master's degree", "PhD or higher",
## "Some professional diploma, no degree", "The baccalaureate")

podaci$pi_education %>%
head(., 10) %T>% print(.) %>%
forcats::fct_recode(., 'elem-sch' = "Elementary School", 'hi-sch' = "High school",
                    'masters' = "Master's degree", 'phd' = "PhD or higher",
                    'prof-dip' = "Some professional diploma, no degree",
                    'bac' = "The baccalaureate") %>% print(.)
##  [1] Some professional diploma, no degree
##  [2] Master's degree
##  [3] High school
##  [4] Master's degree
##  [5] Master's degree
##  [6] Master's degree
##  [7] Some professional diploma, no degree
##  [8] Master's degree
##  [9] The baccalaureate
## [10] Some professional diploma, no degree
## 6 Levels: Elementary School High school Master's degree ... The baccalaureate
##  [1] prof-dip masters  hi-sch   masters  masters  masters  prof-dip
##  [8] masters  bac      prof-dip
## Levels: elem-sch hi-sch masters phd prof-dip bac
```

Kratko pojašnjenje: uzimamo samo varijablu `pi_education` te prvih 10 unosa (`head`). Usput pozivamo `print` (s T-pipom!) kako bismo ispisali izvornih 10 vrijednosti. Varijablu s tih 10 vrijednosti šaljemo u `fct_recode`, gdje rekodiramo razine. Naposljetku, pozivamo `print` kako bismo ispisali nove vrijednosti (`print` ovdje nije potreban, tu je samo zato da bi se output izjednačio onom koji dobivamo nakon prvog poziva; to je specifičnost Jupyter Notebooka). Sad kad smo zadovoljni outputom, možemo maknuti nepotrebne dijelove i upisati promjenu.

```
podaci$pi_education %<>%
forcats::fct_recode(., 'elem-sch' = "Elementary School", 'hi-sch' = "High school",
                    'masters' = "Master's degree", 'phd' = "PhD or higher",
                    'prof-dip' = "Some professional diploma, no degree",
                    'bac' = "The baccalaureate")

levels(podaci$pi_education)
## [1] "elem-sch" "hi-sch"   "masters"  "phd"      "prof-dip" "bac"
```

Vježba

Pokušajte napraviti isto s varijablom `pi_income`.

Rekodirajte razine tako da `avg` označava `About the average`, a razine ispod i iznad toga označite dodavanjem odgovarajućeg broja minusa odnosno pluseva na kraj (npr. `avg-` ili `avg++`).

```
podaci$pi_income %>% levels(.) %>% dput(.)
## c("About the average", "Much above the average", "Much below the average",
## "Somewhat above the average", "Somewhat below the average")

podaci$pi_income %<>%
```

```
forcats::fct_recode(., 'avg' = "About the average",
                    'avg++' = "Much above the average",
                    'avg--' = "Much below the average",
                    'avg+' = "Somewhat above the average",
                    'avg-' = "Somewhat below the average")
```

Ovdje možemo primijetiti da je redoslijed razina podosta besmislen, tako da ćemo ih izvrtiti tako da idu od najniže do najviše. To ćemo učiniti pomoću funkcije `fct_relevel`.

```
podaci$pi_income %>%
forcats::fct_relevel(., 'avg--', 'avg-', 'avg', 'avg+', 'avg++') %>%
# još ćemo faktor pretvoriti u ordered
factor(., ordered = T) %>%
tail(., 10) %>% print(.)
##  [1] avg-- avg-  avg-  avg-  avg+  avg   avg++ avg+  avg-  avg+
## Levels: avg-- < avg- < avg < avg+ < avg++

podaci$pi_income %<>%
forcats::fct_relevel(., 'avg--', 'avg-', 'avg', 'avg+', 'avg++') %>%
factor(., ordered = T)

str(podaci$pi_income)
##  Ord.factor w/ 5 levels "avg--"<"avg-"<..: 2 4 4 4 4 4 4 4 4 2 ...
```

Nećemo prolaziti kroz rekodiranje svih faktora, ali hoćemo proći kroz rekodiranje nacionalnosti, zato jer nam to daje mogućnost da se igramo sa stringovima i regularnim izrazima.

Kodiranje nacionalnosti (pitanje otvorenog tipa)

Pitanje o nacionalnosti bilo je otvorenog tipa, tako da ista nacionalnost može biti reprezentirana na različite načine.

```
podaci$pi_nationality %>% head(.)
## [1] American                 USA
## [3] Turkish                  United States of America
## [5] US                       USA
## 16 Levels: American Asian american Australian British Canadian ... White
```

Već u prvih 6 unosa vidimo da se javljaju "US", "USA", "United States of America" te "American", što sve označava istu nacionalnost. Koristeći regularne izraze i funkciju `case_when`, lako možemo grupirati različite unose. Za početak, iskoristit ćemo funkciju `tolower` kako bismo sve stringove pretvorili u mala slova (tako da ne moramo paziti na to da su "american" i "American" različiti unosi) te funkciju `str_trim`, koja će ukloniti razmake s početka i kraja stringova (jer je moguće da je netko unio "American", a netko "American").

```
podaci$pi_nationality %<>% tolower(.) %>% stringr::str_trim(.)

head(podaci$pi_nationality)
## [1] "american"               "usa"
## [3] "turkish"                "united states of america"
## [5] "us"                     "usa"
```

Ok. Za početak, možemo pozvati `table` da dobijemo pregled frekvencija po faktorima, te `sort` kako bismo ih poredali od najučestalijih do najrjeđih.

```
table(podaci$pi_nationality) %>% sort(., decreasing = T)
##
```

```
##                     usa                american                  canadian
##                      26                      24                        13
##                 british           united states united states of america
##                      11                       9                         5
##               australian                  french               seychelles
##                       2                       2                         2
##                      us          asian american                     dutch
##                       2                       1                         1
##                 turkish                   white
##                       1                       1
```

Budući da ovdje imamo samo 100 sudionika i razmjerno malo različitih nacionalnosti, rekodiranje je lako. Za kodiranje nacionalnosti koristit ćemo funkciju `case_when`, koja nam omogućuje da specificiramo neki logički izraz (dakle, nešto što kao rezultat vraća `TRUE` ili `FALSE`) i akciju koju treba napraviti u `TRUE` slučaju. `case_when` za argumente prima logičke izraze i akcije odvojene tildom (`~`), pa pozivanje funkcije izgleda ovako:

```
case_when(logički-izraz ~ akcija-ako-TRUE,          logički-izraz-2 ~ akcija-ako-TRUE-2)
```

```
podaci$pi_nationality %>%
# case_when ovdje moramo obaviti u {} jer inače dobijemo error
{dplyr::case_when(stringr::str_detect(., 'usa?|american|united states.*|\\w+ americ') ~ 'american',
           str_detect(., 'dutch|french') ~ 'fr-nl',
           str_detect(., 'seychelles|turkish|white') ~ 'other',
           # akciju u svim nespecificiranim slučajevima određujemo
           # tako da stavimo TRUE ~ akcija. ovdje kao akciju stavljamo
           # točku, što znači da taj unos treba ostaviti onakvim
           # kakav je
           TRUE ~ .)} %>% table(.)
## .
## american  british canadian    fr-nl    other
##       69       11       13        3        4

podaci$pi_nationality %<>%
{dplyr::case_when(stringr::str_detect(., 'usa?|american|united states.*|\\w+ americ') ~ 'american',
           str_detect(., 'dutch|french') ~ 'fr-nl',
           str_detect(., 'seychelles|turkish|white') ~ 'other',
           TRUE ~ .)} %>%
as.factor(.)
```

Preimenovanje varijabli

Nekad su imena varijabli jako nezgrapna, neinformativna, mutava i slično. Budući da ćete se prije ili poslije susresti s takvim imenima, proći ćemo kroz nekoliko načina za mijenjanje imena varijabli. Ako želimo promijeniti imena manjeg broja varijabli, možemo koristiti funkciju `rename`. Na primjer, varijable `charitableBehavior01` i `charitableBehavior02` ne govore ništa o tome što su. Jedna je namjera doniranja novca, a druga namjera doniranja vremena. Stoga, preimenovat ćemo ih u `donationMoney` i `donationTime`.

```
podaci %>%
dplyr::select(10:11) %>%
colnames(.)
## [1] "charitableBehavior01" "charitableBehavior02"

podaci %<>%
```

```
dplyr::rename(., donationMoney = charitableBehavior01,
      donationTime = charitableBehavior02)

podaci %>%
dplyr::select(10:11) %>%
colnames(.)
## [1] "donationMoney" "donationTime"
```

Ako trebamo preimenovati veći broj varijabli i ako smo te sreće da njihova imena možemo uhvatiti regularnim izrazima, možemo koristiti `str_replace`. Na primjer, imamo 32 varijable koje se zovu `moralFoundationsXX` i koje predstavljaju pitanja na Moral Foundations Questionnaireu. MFQ se sastoji od 5 faktora (authority, care, loyalty, fairness, sanctity) - svaki faktor reprezentiran je sa 6 pitanja. Osim toga, ima i dvije kontrolne čestice. Preimenovat ćemo varijable tako da na kraj imena svake od njih dodamo oznaku faktora kojoj pripada. Za to ćemo koristiti funkciju `str_replace`, koja nam omogućuje da neki obrazac definiran regexom zamijenimo nekim drugim stringom.

```
qc(orahovica, orašar) %>%
stringr::str_replace(., 'ora(h|š)', 'bor')
## [1] "borovica" "borar"
```

Sad ćemo vidjeti kako ovu funkciju možemo koristiti za preimenovati varijable.

```
# dohvaćamo imena stupaca
colnames(podaci) %>%
# specificiramo stupce na kojima želimo izvršiti zamjenu
stringr::str_replace(., pattern = '(moralFoundations)(01|07|12|17|23|28)',
                     replacement = '\\1\\2_care') %>%
# ovo je samo radi prikazivanja svih MFQ pitanja
stringr::str_subset(., 'moralFoundations') %>% print(.)
##  [1] "moralFoundations01_care" "moralFoundations02"
##  [3] "moralFoundations03"      "moralFoundations04"
##  [5] "moralFoundations05"      "moralFoundations06"
##  [7] "moralFoundations07_care" "moralFoundations08"
##  [9] "moralFoundations09"      "moralFoundations10"
## [11] "moralFoundations11"      "moralFoundations12_care"
## [13] "moralFoundations13"      "moralFoundations14"
## [15] "moralFoundations15"      "moralFoundations16"
## [17] "moralFoundations17_care" "moralFoundations18"
## [19] "moralFoundations19"      "moralFoundations20"
## [21] "moralFoundations21"      "moralFoundations22"
## [23] "moralFoundations23_care" "moralFoundations24"
## [25] "moralFoundations25"      "moralFoundations26"
## [27] "moralFoundations27"      "moralFoundations28_care"
## [29] "moralFoundations29"      "moralFoundations30"
## [31] "moralFoundations31"      "moralFoundations32"
```

Vidimo da pitanja koja smo odredili sada imaju sufiks `_care`. U `replacement` argumentu smo iskoristili mogućnost referenciranja koju nam nudi grupiranje znakova u regularnim izrazima. Počevši s lijeva, svaku grupu definiranu pomoću `(...)` možemo dohvatiti pomoću `\\n`, gdje `n` označava redni broj grupe. Dakle, u gornjem primjeru se pri izvršavanju zamjene `\\1` širi u prvu pronađenu grupu (moralFoundations), a `\\2` u drugu pronađenu grupu (01, 07, 12, 17, 23 ili 28, ovisno o tome što je u pojedinom stringu pronađeno). Time dobivamo `moralFoundations01_care`, `moralFoundations07_care` itd. Kod ovakvog mijenjanja imena je zgodno to što nam se svaki put vraćaju imena svih stupaca - ako u imenu nekog stupca nije pronađen uzorak koji smo specificirali u `pattern`, ono ostaje netaknuto. Zbog toga, možemo napraviti lanac poziva `str_replace` pomoću pipa.

```r
colnames(podaci) %>%
    stringr::str_replace(., '(moralFoundations)(01|07|12|17|23|28)', '\\1\\2_care') %>%
    str_replace(., '(moralFoundations)(02|08|13|18|24|29)', '\\1\\2_fair') %>%
    str_replace(., '(moralFoundations)(03|09|14|19|25|30)', '\\1\\2_loyal') %>%
    str_replace(., '(moralFoundations)(04|10|15|20|26|31)', '\\1\\2_author') %>%
    str_replace(., '(moralFoundations)(05|11|16|21|27|32)', '\\1\\2_sanct') %>%
    str_replace(., '(moralFoundations)(06|22)', '\\1\\2_control') %>%
print(.)
##  [1] "attitudesAndNorms01"            "attitudesAndNorms02"
##  [3] "attitudesAndNorms03"            "attitudesAndNorms04"
##  [5] "attitudesAndNorms05"            "attitudesAndNorms06"
##  [7] "attitudesAndNorms07"            "attitudesAndNorms08"
##  [9] "callToAction"                   "donationMoney"
## [11] "donationTime"                   "descriptiveSocialNorms01"
## [13] "descriptiveSocialNorms02"       "descriptiveSocialNorms03"
## [15] "descriptiveSocialNorms04"       "mf_AuthoritySubversion"
## [17] "mf_CareHarm"                    "mf_FairnessCheating"
## [19] "mf_LoyaltyBetrayal"             "mf_SanctityDegradation"
## [21] "moralFoundations01_care"        "moralFoundations02_fair"
## [23] "moralFoundations03_loyal"       "moralFoundations04_author"
## [25] "moralFoundations05_sanct"       "moralFoundations06_control"
## [27] "moralFoundations07_care"        "moralFoundations08_fair"
## [29] "moralFoundations09_loyal"       "moralFoundations10_author"
## [31] "moralFoundations11_sanct"       "moralFoundations12_care"
## [33] "moralFoundations13_fair"        "moralFoundations14_loyal"
## [35] "moralFoundations15_author"      "moralFoundations16_sanct"
## [37] "moralFoundations17_care"        "moralFoundations18_fair"
## [39] "moralFoundations19_loyal"       "moralFoundations20_author"
## [41] "moralFoundations21_sanct"       "moralFoundations22_control"
## [43] "moralFoundations23_care"        "moralFoundations24_fair"
## [45] "moralFoundations25_loyal"       "moralFoundations26_author"
## [47] "moralFoundations27_sanct"       "moralFoundations28_care"
## [49] "moralFoundations29_fair"        "moralFoundations30_loyal"
## [51] "moralFoundations31_author"      "moralFoundations32_sanct"
## [53] "moralIdentityInternalization01" "moralIdentityInternalization02"
## [55] "moralIdentityInternalization03" "moralIdentityInternalization04"
## [57] "moralIdentityInternalization05" "pi_age"
## [59] "pi_education"                   "pi_gender"
## [61] "pi_ideology"                    "pi_income"
## [63] "pi_nationality"                 "pi_previousDonations"
```

Kad smo sigurni da dobivamo ono što očekujemo, samo promijenimo pipu %>% u %<>%.

```r
colnames(podaci) %<>%
    str_replace(., '(moralFoundations)(01|07|12|17|23|28)', '\\1\\2_care') %>%
    str_replace(., '(moralFoundations)(02|08|13|18|24|29)', '\\1\\2_fair') %>%
    str_replace(., '(moralFoundations)(03|09|14|19|25|30)', '\\1\\2_loyal') %>%
    str_replace(., '(moralFoundations)(04|10|15|20|26|31)', '\\1\\2_author') %>%
    str_replace(., '(moralFoundations)(05|11|16|21|27|32)', '\\1\\2_sanct') %>%
    str_replace(., '(moralFoundations)(06|22)', '\\1\\2_control')

colnames(podaci) %>% print(.)
##  [1] "attitudesAndNorms01"            "attitudesAndNorms02"
##  [3] "attitudesAndNorms03"            "attitudesAndNorms04"
```

```
##  [5] "attitudesAndNorms05"            "attitudesAndNorms06"
##  [7] "attitudesAndNorms07"            "attitudesAndNorms08"
##  [9] "callToAction"                   "donationMoney"
## [11] "donationTime"                   "descriptiveSocialNorms01"
## [13] "descriptiveSocialNorms02"       "descriptiveSocialNorms03"
## [15] "descriptiveSocialNorms04"       "mf_AuthoritySubversion"
## [17] "mf_CareHarm"                    "mf_FairnessCheating"
## [19] "mf_LoyaltyBetrayal"             "mf_SanctityDegradation"
## [21] "moralFoundations01_care"        "moralFoundations02_fair"
## [23] "moralFoundations03_loyal"       "moralFoundations04_author"
## [25] "moralFoundations05_sanct"       "moralFoundations06_control"
## [27] "moralFoundations07_care"        "moralFoundations08_fair"
## [29] "moralFoundations09_loyal"       "moralFoundations10_author"
## [31] "moralFoundations11_sanct"       "moralFoundations12_care"
## [33] "moralFoundations13_fair"        "moralFoundations14_loyal"
## [35] "moralFoundations15_author"      "moralFoundations16_sanct"
## [37] "moralFoundations17_care"        "moralFoundations18_fair"
## [39] "moralFoundations19_loyal"       "moralFoundations20_author"
## [41] "moralFoundations21_sanct"       "moralFoundations22_control"
## [43] "moralFoundations23_care"        "moralFoundations24_fair"
## [45] "moralFoundations25_loyal"       "moralFoundations26_author"
## [47] "moralFoundations27_sanct"       "moralFoundations28_care"
## [49] "moralFoundations29_fair"        "moralFoundations30_loyal"
## [51] "moralFoundations31_author"      "moralFoundations32_sanct"
## [53] "moralIdentityInternalization01" "moralIdentityInternalization02"
## [55] "moralIdentityInternalization03" "moralIdentityInternalization04"
## [57] "moralIdentityInternalization05" "pi_age"
## [59] "pi_education"                   "pi_gender"
## [61] "pi_ideology"                    "pi_income"
## [63] "pi_nationality"                 "pi_previousDonations"
```

Varijable u ovom setu zapravo su dosta dobro imenovane. Neke nisu dovoljno jasne, ali imenovanje je sustavno, što uvelike olakšava baratnje podacima. Nekad (kad radite s podacima sa Survey Monkeyja, recimo) vjerojatno nećete imati toliko jasne slučajeve. Na primjer, ime varijable moglo bi biti `1. Molimo Vas, odaberite vaš ekonomski status`. Takva imena su pakao. Kad bismo tako imenovanu varijablu ubacili u R, dobili bismo nešto ružno.

```
ruzno <- data.frame('1. Molimo Vas, odaberite vaš ekonomski status:' = 1:5)
print(ruzno)
##   X1..Molimo.Vas..odaberite.vaš.ekonomski.status.
## 1                                               1
## 2                                               2
## 3                                               3
## 4                                               4
## 5                                               5
```

Svaki razmak postao je točka, zarez i dvotočka također su postali točke, a imenu varijable dodan je prefiks `X` (jer ime varijable ne može započinjati brojem!). Možemo pozvati funkciju `clean_names` iz paketa `janitor`, koja će od ružnih imena napraviti nešto ljepša.

```
lijepo <- janitor::clean_names(ruzno)
print(lijepo)
##   x1_molimo_vas_odaberite_vas_ekonomski_status
## 1                                            1
## 2                                            2
```

```
## 3                                                 3
## 4                                                 4
## 5                                                 5
```

Ovisno o konkretnom imenu, ova će funkcija biti manje ili više korisna. Recimo, ako je potrebno u potpunosti preimenovati varijablu u nešto smisleno, nema druge nego ručno.

Ipak, isplati se pozvati `clean_names` jer može uvelike olakšati automatizirano preimenovanje. Dodat ćemo još 2 ružna stupca u `data.frame` `ruzno`.

```
ruzno %<>% data.frame(., '2. Koliko sam vina ja popio?' = 15:19,
                      '3. Je li vaše ludo srce biralo?' = F)
print(ruzno)
##   X1..Molimo.Vas..odaberite.vaš.ekonomski.status.
## 1                                               1
## 2                                               2
## 3                                               3
## 4                                               4
## 5                                               5
##   X2..Koliko.sam.vina.ja.popio. X3..Je.li.vaše.ludo.srce.biralo.
## 1                             15                           FALSE
## 2                             16                           FALSE
## 3                             17                           FALSE
## 4                             18                           FALSE
## 5                             19                           FALSE
```

Vidimo da su i upitnici pretvoreni u točke. Recimo da hoćemo svako ime svesti na format `[broj pitanja]_[prva riječ]`. Ako dopustimo R-u da obavi svoju masovnu konverziju, pa takva imena pretvaramo, mogli bismo imati problema (ili više nepotrebne patnje) sa specificiranjem obrasca koji želimo odbaciti. Ponovno ćemo pozvati `clean_names`:

```
lijepo <- janitor::clean_names(ruzno)
print(lijepo)
##   x1_molimo_vas_odaberite_vas_ekonomski_status x2_koliko_sam_vina_ja_popio
## 1                                            1                          15
## 2                                            2                          16
## 3                                            3                          17
## 4                                            4                          18
## 5                                            5                          19
##   x3_je_li_vase_ludo_srce_biralo
## 1                          FALSE
## 2                          FALSE
## 3                          FALSE
## 4                          FALSE
## 5                          FALSE
```

Ova imena su puno sustavnija, zbog čega je lakše napisati neki obrazac znakova koji želimo zadržati. Za primjer, svest ćemo imena varijabli na format `[broj pitanja]_[prva riječ]`.

```
colnames(lijepo) %<>%
stringr::str_replace(., '^x(\\d_[[:lower:]]+).*', '\\1')
print(lijepo)
##   1_molimo 2_koliko  3_je
## 1        1       15 FALSE
## 2        2       16 FALSE
## 3        3       17 FALSE
## 4        4       18 FALSE
```

```
## 5        5       19 FALSE
```

Obrnuto kodiranje varijabli

Neka od pitanja u ovom upitnik potrebno je obrnuto kodirati. To možemo učiniti pomoću funkcije `reverse.code` iz `psych` paketa. Ta funkcija ima dva obavezna argumenta: `keys`, koji je vektor brojki `1` i `-1`, te `items`, što su čestice koje treba rekodirati. Za primjer, rekodirat ćemo 3. i 4. pitanje skale `moralIdentityInternalization`.

```
podaci %>%
dplyr::select(contains('Internal')) %>%
head(.) %T>% print(.) %>%
{psych::reverse.code(keys = c(1, 1, -1, -1, 1),
                     items = .,
                     # zadajemo maksimum i minimum skale
                     # jer inače određuje prema vrijednostima
                     # koje se zapravo pojavljuju, a neke
                     # čestice imaju manji raspon od
                     # teoretski mogućeg
                     mini = 0, maxi = 7)} %T>%
str(.) %>% head(.)
## # A tibble: 6 x 5
##   moralIdentityIn~ moralIdentityIn~ moralIdentityIn~ moralIdentityIn~
##              <int>            <int>            <int>            <int>
## 1                5                2                1                2
## 2                4                3                1                3
## 3                6                5                1                1
## 4                6                4                1                3
## 5                4                3                1                2
## 6                4                6                1                1
## # ... with 1 more variable: moralIdentityInternalization05 <int>
##  num [1:6, 1:5] 5 4 6 6 4 4 2 3 5 4 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:5] "moralIdentityInternalization01" "moralIdentityInternalization02" "moralIdentityIn
##      moralIdentityInternalization01 moralIdentityInternalization02
## [1,]                              5                              2
## [2,]                              4                              3
## [3,]                              6                              5
## [4,]                              6                              4
## [5,]                              4                              3
## [6,]                              4                              6
##      moralIdentityInternalization03- moralIdentityInternalization04-
## [1,]                               6                               5
## [2,]                               6                               4
## [3,]                               6                               6
## [4,]                               6                               4
## [5,]                               6                               5
## [6,]                               6                               6
##      moralIdentityInternalization05
## [1,]                              3
## [2,]                              4
## [3,]                              5
```

```
## [4,]                                4
## [5,]                                4
## [6,]                                4
```

Sad kad smo se uvjerili da su varijable ispravno rekodirane, možemo skratiti postupak (recimo, tako da ciljamo samo one varijable koje zapravo treba rekodirati) i te rekodirane varijable dodati u `data.frame`.

```
podaci %<>%
# contains smo promijenili u matches
dplyr::select(matches('Internal.*(03|04)$')) %>%
# u keys ostavljamo samo onoliko -1 koliko
# imamo varijabli
{psych::reverse.code(keys = c(-1, -1),
                     items = .,
                     mini = 0, maxi = 7)} %>%
# reverse.code nam vraća matrix, pa ga pretvaramo
# u data.frame
as.data.frame(.) %$%
# otkrivamo imena varijabli kako bismo ih mogli
# koristiti direktno; tibble je dio tidyversea
tibble::add_column(podaci,
                   moralIdentityInternalization03_rec =
                   # ime varijable moramo staviti u `` (backticks)
                   # jer R inače baca error zbog - na kraju imena
                   # (taj - tumači kao sintaksu, a ne kao dio imena)
                   `moralIdentityInternalization03-`,
                   moralIdentityInternalization04_rec =
                   `moralIdentityInternalization04-`,
                   # pomoću .after definiramo iza kojeg stupca
                   # želimo dodati nove stupce; ovdje to radimo
                   # zato da bi mII varijable bile na okupu
                   .after = 'moralIdentityInternalization05')

colnames(podaci) %>% print(.)
##  [1] "attitudesAndNorms01"
##  [2] "attitudesAndNorms02"
##  [3] "attitudesAndNorms03"
##  [4] "attitudesAndNorms04"
##  [5] "attitudesAndNorms05"
##  [6] "attitudesAndNorms06"
##  [7] "attitudesAndNorms07"
##  [8] "attitudesAndNorms08"
##  [9] "callToAction"
## [10] "donationMoney"
## [11] "donationTime"
## [12] "descriptiveSocialNorms01"
## [13] "descriptiveSocialNorms02"
## [14] "descriptiveSocialNorms03"
## [15] "descriptiveSocialNorms04"
## [16] "mf_AuthoritySubversion"
## [17] "mf_CareHarm"
## [18] "mf_FairnessCheating"
## [19] "mf_LoyaltyBetrayal"
## [20] "mf_SanctityDegradation"
```

```
## [21] "moralFoundations01_care"
## [22] "moralFoundations02_fair"
## [23] "moralFoundations03_loyal"
## [24] "moralFoundations04_author"
## [25] "moralFoundations05_sanct"
## [26] "moralFoundations06_control"
## [27] "moralFoundations07_care"
## [28] "moralFoundations08_fair"
## [29] "moralFoundations09_loyal"
## [30] "moralFoundations10_author"
## [31] "moralFoundations11_sanct"
## [32] "moralFoundations12_care"
## [33] "moralFoundations13_fair"
## [34] "moralFoundations14_loyal"
## [35] "moralFoundations15_author"
## [36] "moralFoundations16_sanct"
## [37] "moralFoundations17_care"
## [38] "moralFoundations18_fair"
## [39] "moralFoundations19_loyal"
## [40] "moralFoundations20_author"
## [41] "moralFoundations21_sanct"
## [42] "moralFoundations22_control"
## [43] "moralFoundations23_care"
## [44] "moralFoundations24_fair"
## [45] "moralFoundations25_loyal"
## [46] "moralFoundations26_author"
## [47] "moralFoundations27_sanct"
## [48] "moralFoundations28_care"
## [49] "moralFoundations29_fair"
## [50] "moralFoundations30_loyal"
## [51] "moralFoundations31_author"
## [52] "moralFoundations32_sanct"
## [53] "moralIdentityInternalization01"
## [54] "moralIdentityInternalization02"
## [55] "moralIdentityInternalization03"
## [56] "moralIdentityInternalization04"
## [57] "moralIdentityInternalization05"
## [58] "moralIdentityInternalization03_rec"
## [59] "moralIdentityInternalization04_rec"
## [60] "pi_age"
## [61] "pi_education"
## [62] "pi_gender"
## [63] "pi_ideology"
## [64] "pi_income"
## [65] "pi_nationality"
## [66] "pi_previousDonations"
```

Brisanje stupaca

Ponekad se u podacima nađu varijable koje nam nisu potrebne, pa je zgodno znati kako ih možemo obrisati. Za potrebe ove demonstracije, obrisat ćemo dvije varijable - `mf_CareHarm` i `mf_FairnessCheating` - koje su ukupni rezultati na dvije subskale MFQ-a. Jedan način za brisanje je upisivanje posebne vrijednosti `NULL` u

stupac kojeg se želimo riješiti.

```
podaci$mf_CareHarm <- NULL

podaci %>% dplyr::select(., starts_with('mf_')) %>% str(.)
## Classes 'tbl_df', 'tbl' and 'data.frame':    100 obs. of  4 variables:
##  $ mf_AuthoritySubversion: int  1 1 2 2 2 0 2 1 1 2 ...
##  $ mf_FairnessCheating   : int  3 3 4 3 2 4 4 5 3 4 ...
##  $ mf_LoyaltyBetrayal    : int  2 2 2 3 2 1 2 0 0 1 ...
##  $ mf_SanctityDegradation: int  1 1 1 1 1 -1 1 -1 1 1 ...
```

Drugi je prepisivanje (u smislu *overwrite*) varijable koja drži `data.frame` `data.frameom` koji sadrži sve varijable osim te koju želimo ukloniti. To možemo učiniti pomoću funkcije `select` i negacijskog operatora `-`.

```
podaci %<>%
dplyr::select(-mf_FairnessCheating)

podaci %>% dplyr::select(., starts_with('mf_')) %>% str(.)
## Classes 'tbl_df', 'tbl' and 'data.frame':    100 obs. of  3 variables:
##  $ mf_AuthoritySubversion: int  1 1 2 2 2 0 2 1 1 2 ...
##  $ mf_LoyaltyBetrayal    : int  2 2 2 3 2 1 2 0 0 1 ...
##  $ mf_SanctityDegradation: int  1 1 1 1 1 -1 1 -1 1 1 ...
```

Stvaranje nove varijable pomoću `mutate`

Već smo vidjeli neke načine na koje možemo stvarati nove varijable. Sada ćemo pomoću funkcije `mutate` rekreirati dva stupca koja smo malo prije obrisali.

Kao rezultat na subskali uzet ćemo prosječnu vrijednost odabranih odgovora svakog sudionika.

```
podaci %>%
# koristimo rowMeans, koji računa aritmetičku sredinu svakog reda,
# kao što i samo ime kaže. funkciju primjenjujemo na varijable
# koje završavaju s 'care', što možemo napraviti jer smo bili
# mudri i smisleno i sustavno imenovali varijable
dplyr::mutate(.,
              mf_CareHarm = rowMeans(dplyr::select(.,
                                                   dplyr::ends_with('care'))),
              mf_FairnessCheating = rowMeans(dplyr::select(.,
                                                           dplyr::ends_with('fair')))) %>%
# kad koristimo select, redoslijed kojim unosimo varijable u funkciju
# određuje redoslijed varijabli nakon odabira stupaca. stoga, budući da
# mutate vraća data.frame, možemo iskoristiti select da nove varijable
# preselimo do njima srodnih. primijetit ćemo da u selectu možemo
# kombinirati numeričke indekse i imena varijabli; koristimo
# everything() za dodavanje svih preostalih varijabli
dplyr::select(., 1:mf_SanctityDegradation, mf_CareHarm, mf_FairnessCheating,
       dplyr::everything()) %>% str(.)
## Classes 'tbl_df', 'tbl' and 'data.frame':    100 obs. of  66 variables:
##  $ attitudesAndNorms01               : int  5 5 4 6 4 4 6 4 3 5 ...
##  $ attitudesAndNorms02               : int  5 4 6 2 1 4 0 4 7 7 ...
##  $ attitudesAndNorms03               : int  5 2 5 3 2 4 3 5 6 7 ...
##  $ attitudesAndNorms04               : int  5 1 5 2 3 3 3 7 5 6 ...
##  $ attitudesAndNorms05               : int  4 2 3 2 1 4 2 4 4 6 ...
##  $ attitudesAndNorms06               : int  3 2 2 3 2 3 3 3 3 4 ...
```

```
##  $ attitudesAndNorms07               : int  4 3 4 5 4 5 6 4 4 5 ...
##  $ attitudesAndNorms08               : int  6 7 5 6 5 5 7 5 3 5 ...
##  $ callToAction                      : int  7 6 7 1 8 7 11 8 3 7 ...
##  $ donationMoney                     : int  37 18 7 14 0 37 33 29 16 6 ...
##  $ donationTime                      : int  4 3 3 5 0 2 4 3 2 3 ...
##  $ descriptiveSocialNorms01          : int  4 3 3 1 3 1 2 4 3 4 ...
##  $ descriptiveSocialNorms02          : int  3 1 3 1 1 1 2 3 3 5 ...
##  $ descriptiveSocialNorms03          : int  2 3 2 2 2 3 3 4 4 5 ...
##  $ descriptiveSocialNorms04          : int  2 1 5 3 4 2 2 2 2 4 ...
##  $ mf_AuthoritySubversion            : int  1 1 2 2 2 0 2 1 1 2 ...
##  $ mf_LoyaltyBetrayal                : int  2 2 2 3 2 1 2 0 0 1 ...
##  $ mf_SanctityDegradation            : int  1 1 1 1 1 -1 1 -1 1 1 ...
##  $ mf_CareHarm                       : num  3.17 3 3.5 2.83 3.17 ...
##  $ mf_FairnessCheating               : num  3 3 3.17 3.67 3.33 ...
##  $ moralFoundations01_care           : int  4 3 4 3 3 4 5 3 4 4 ...
##  $ moralFoundations02_fair           : int  4 3 4 3 1 4 4 4 2 5 ...
##  $ moralFoundations03_loyal          : int  3 0 2 1 1 0 2 0 -1 0 ...
##  $ moralFoundations04_author         : int  1 0 2 2 2 0 2 0 1 2 ...
##  $ moralFoundations05_sanct          : int  2 2 1 3 3 -1 3 1 1 2 ...
##  $ moralFoundations06_control        : int  0 0 0 2 -1 1 0 0 -1 1 ...
##  $ moralFoundations07_care           : int  4 3 4 4 5 2 4 4 3 4 ...
##  $ moralFoundations08_fair           : int  4 3 4 3 3 4 5 4 3 5 ...
##  $ moralFoundations09_loyal          : int  3 3 2 4 3 3 3 1 -1 1 ...
##  $ moralFoundations10_author         : int  0 -1 1 3 2 0 2 1 1 1 ...
##  $ moralFoundations11_sanct          : int  1 3 1 0 1 -1 2 0 3 3 ...
##  $ moralFoundations12_care           : int  6 5 4 5 4 4 5 4 3 5 ...
##  $ moralFoundations13_fair           : int  3 5 4 5 5 3 4 5 4 5 ...
##  $ moralFoundations14_loyal          : int  4 2 1 1 3 3 3 1 3 2 ...
##  $ moralFoundations15_author         : int  3 2 2 1 2 3 3 2 5 3 ...
##  $ moralFoundations16_sanct          : int  3 1 1 2 -1 1 -2 2 0 1 ...
##  $ moralFoundations17_care           : int  2 5 3 4 4 4 3 4 3 3 ...
##  $ moralFoundations18_fair           : int  2 3 3 4 5 2 4 5 4 4 ...
##  $ moralFoundations19_loyal          : int  0 2 4 2 2 2 4 4 4 2 ...
##  $ moralFoundations20_author         : int  0 1 0 4 1 3 3 3 2 2 ...
##  $ moralFoundations21_sanct          : int  0 1 1 1 3 3 1 2 1 -1 ...
##  $ moralFoundations22_control        : int  4 4 6 4 4 3 5 3 5 5 ...
##  $ moralFoundations23_care           : int  3 3 4 2 4 0 3 4 3 3 ...
##  $ moralFoundations24_fair           : int  4 3 1 5 2 3 2 6 2 3 ...
##  $ moralFoundations25_loyal          : int  0 0 1 2 0 3 1 2 2 1 ...
##  $ moralFoundations26_author         : int  1 1 1 5 0 2 2 3 1 1 ...
##  $ moralFoundations27_sanct          : int  1 1 0 1 1 1 -1 2 0 1 ...
##  $ moralFoundations28_care           : int  0 -1 2 -1 -1 1 3 1 4 1 ...
##  $ moralFoundations29_fair           : int  1 1 3 2 4 1 4 2 2 0 ...
##  $ moralFoundations30_loyal          : int  1 1 1 1 2 1 1 0 2 2 ...
##  $ moralFoundations31_author         : int  3 2 1 5 2 2 4 3 3 2 ...
##  $ moralFoundations32_sanct          : int  1 0 0 4 2 1 0 1 2 1 ...
##  $ moralIdentityInternalization01    : int  5 4 6 6 4 4 5 3 6 5 ...
##  $ moralIdentityInternalization02    : int  2 3 5 4 3 6 5 2 4 5 ...
##  $ moralIdentityInternalization03    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ moralIdentityInternalization04    : int  2 3 1 3 2 1 3 3 3 1 ...
##  $ moralIdentityInternalization05    : int  3 4 5 4 4 4 5 3 4 5 ...
##  $ moralIdentityInternalization03_rec: num  6 6 6 6 6 6 6 6 6 6 ...
##  $ moralIdentityInternalization04_rec: num  5 4 6 4 5 6 4 4 4 6 ...
```

```
##  $ pi_age                          : int  3 20 20 19 22 25 23 41 16 17 ...
##  $ pi_education                    : Factor w/ 6 levels "elem-sch","hi-sch",..: 5 3 2 3 3 3 5 3 6 .
##  $ pi_gender                       : Factor w/ 2 levels "Female","Male": 2 2 2 2 1 2 1 2 2 1 ...
##  $ pi_ideology                     : Factor w/ 7 levels "Extremely conservative (right)",..: 3 7 3
##  $ pi_income                       : Ord.factor w/ 5 levels "avg--"<"avg-"<..: 2 4 4 4 4 4 4 4 4 2
##  $ pi_nationality                  : Factor w/ 5 levels "american","british",..: 1 1 5 1 1 1 1 1 4
##  $ pi_previousDonations            : Factor w/ 4 levels "Never","Often",..: 3 4 3 3 4 4 4 2 2 3 ..
```

Vidimo da dobivamo što smo i htjeli, pa spremamo promjene.

```
podaci %<>%
dplyr::mutate(.,
              mf_CareHarm = rowMeans(dplyr::select(.,
                                                   dplyr::ends_with('care'))),
              mf_FairnessCheating = rowMeans(dplyr::select(.,
                                                    dplyr::ends_with('fair')))) %>%
dplyr::select(., 1:mf_SanctityDegradation, mf_CareHarm, mf_FairnessCheating,
              dplyr::everything())
```

Long i wide formati podataka

Podaci kojima cijelo vrijeme baratamo nalaze se u **wide** formatu - svaki red predstavlja jedan *case* (u našem slučaju sudionika), a svaki stupac predstavlja jednu varijablu. Često, to je format s kojim želimo raditi. Ipak, ponekad nam je zgodno podatke prebaciti u **long** format, u kojem svaki *case* zauzima nekoliko redova. Takav format je potreban za, recimo, multilevel modeliranje u R-u. Za potrebe demonstracije prebacivanja iz jednog formata u drugi, napravit ćemo novi `data.frame`, koji sadrži podskup varijabli i *caseova* iz `data.framea` `podaci`.

```
podaci %>%
# slice nam omogućuje da biramo
# redove prema indeksu. uzet ćemo
# prvih 10 sudionika
dplyr::slice(., 1:10) %>%
dplyr::select(pi_gender, starts_with('descriptive')) %>%
# dodajemo eksplicitni indeks za svakog sudionika
tibble::add_column(., sub_index = 1:nrow(.)) ->
podaci_wide

podaci_wide
## # A tibble: 10 x 6
##    pi_gender descriptiveSoci~ descriptiveSoci~ descriptiveSoci~
##    <fct>                <int>            <int>            <int>
##  1 Male                     4                3                2
##  2 Male                     3                1                3
##  3 Male                     3                3                2
##  4 Male                     1                1                2
##  5 Female                   3                1                2
##  6 Male                     1                1                3
##  7 Female                   2                2                3
##  8 Male                     4                3                4
##  9 Male                     3                3                4
## 10 Female                   4                5                5
## # ... with 2 more variables: descriptiveSocialNorms04 <int>,
```

```
## #   sub_index <int>
```

`podaci_wide`, dakle, sadrži podskup `podataka`, u wide formatu. Sad ćemo taj `data.frame` prebaciti u long format, koristeći funkciju `gather` (kao, bacamo sve na hrpu) iz `tidyr` paketa. `gatheru` moramo dati neku tablicu s podacima (dakle, recimo, `data.frame`), odrediti ime varijable koja će služiti kao `key`, ime varijable koja će služiti kao `value`, te stupce koje želimo svesti na `key` - `value` format.

```
podaci_wide %>%
tidyr::gather(., key = 'pitanje', value = 'odgovor',
              descriptiveSocialNorms01:descriptiveSocialNorms04) ->
podaci_long

podaci_long
## # A tibble: 40 x 4
##    pi_gender sub_index pitanje                  odgovor
##    <fct>         <int> <chr>                      <int>
##  1 Male              1 descriptiveSocialNorms01       4
##  2 Male              2 descriptiveSocialNorms01       3
##  3 Male              3 descriptiveSocialNorms01       3
##  4 Male              4 descriptiveSocialNorms01       1
##  5 Female            5 descriptiveSocialNorms01       3
##  6 Male              6 descriptiveSocialNorms01       1
##  7 Female            7 descriptiveSocialNorms01       2
##  8 Male              8 descriptiveSocialNorms01       4
##  9 Male              9 descriptiveSocialNorms01       3
## 10 Female           10 descriptiveSocialNorms01       4
## # ... with 30 more rows
```

Za prebacivanje natrag u wide format, koristimo `spread` (kao, bacanje đubreta po livadi). Ovoj funkciji trebamo dati podatke (recimo, `data.frame`), `key` koji želimo "rastaviti" i `value`, što su vrijednosti koje trebamo potpisati pod stupce nastale rastavljanjem `key`.

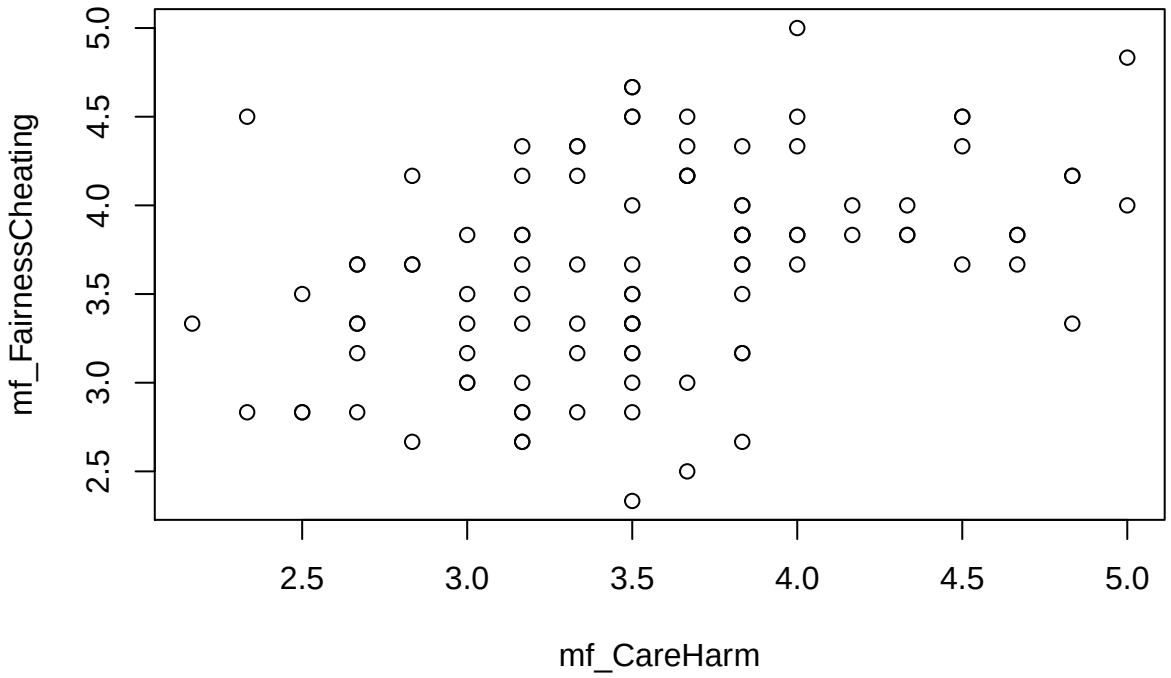
`spread` uzima jedinstvene vrijednosti iz varijable navedene kao `key` i širi ih u nove varijable, koje potom puni vrijednostima zadanima pod `value`.

```
podaci_long %>%
tidyr::spread(., key = pitanje, value = odgovor) %>%
dplyr::arrange(., sub_index)
## # A tibble: 10 x 6
##    pi_gender sub_index descriptiveSoci~ descriptiveSoci~ descriptiveSoci~
##    <fct>         <int>            <int>            <int>            <int>
##  1 Male              1                4                3                2
##  2 Male              2                3                1                3
##  3 Male              3                3                3                2
##  4 Male              4                1                1                2
##  5 Female            5                3                1                2
##  6 Male              6                1                1                3
##  7 Female            7                2                2                3
##  8 Male              8                4                3                4
##  9 Male              9                3                3                4
## 10 Female           10                4                5                5
## # ... with 1 more variable: descriptiveSocialNorms04 <int>
```

Motivacijski primjeri - vizualizacija podataka

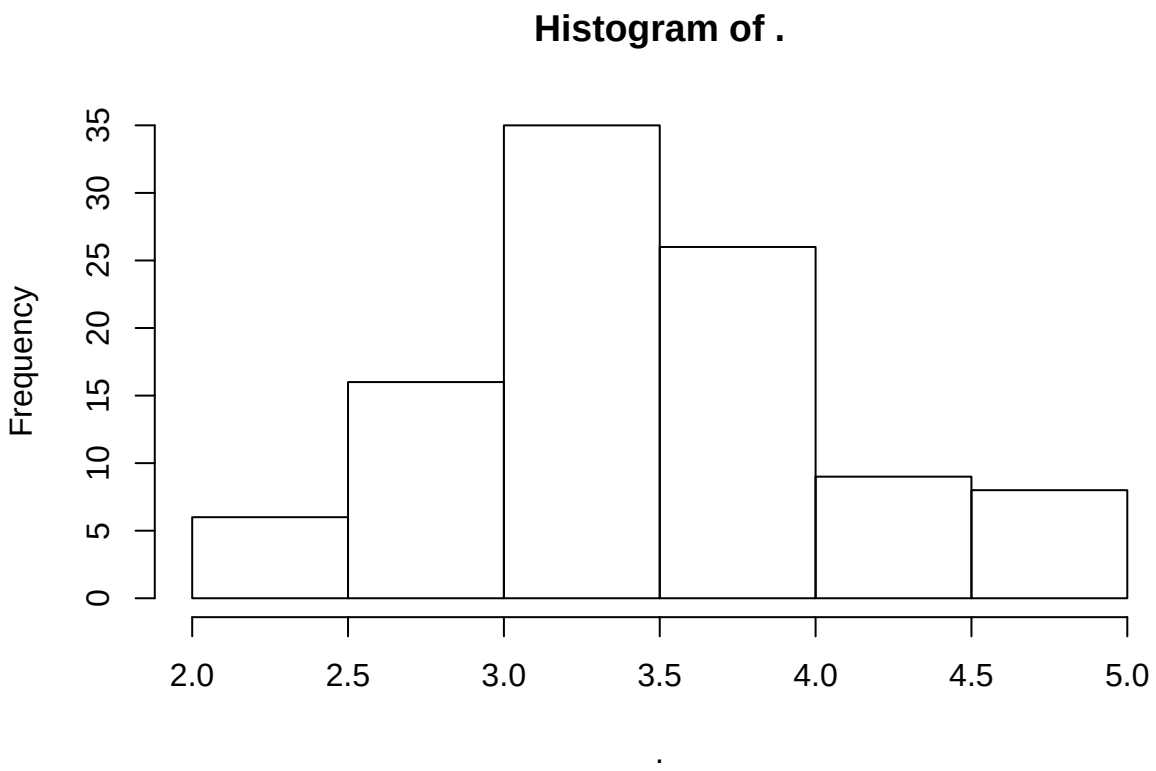
U ovom dugoočekivanom, posljednjem dijelu proći ćemo kroz par motivacijskih primjera koji pokazuju razne zgodnosti koje nam R nudi. Za početak, pogledat ćemo osnove vizualizacije podataka. Kao što smo vidjeli u dijelu o pipama, podatke možemo vizualizirati koristeći generičku funkciju `plot`. Za dobiti, na primjer, dijagram raspršenja, dovoljno je u `plot` proslijediti dvije numeričke varijable.

```
podaci %>%
dplyr::select(., mf_CareHarm, mf_FairnessCheating) %>%
plot(.)
```



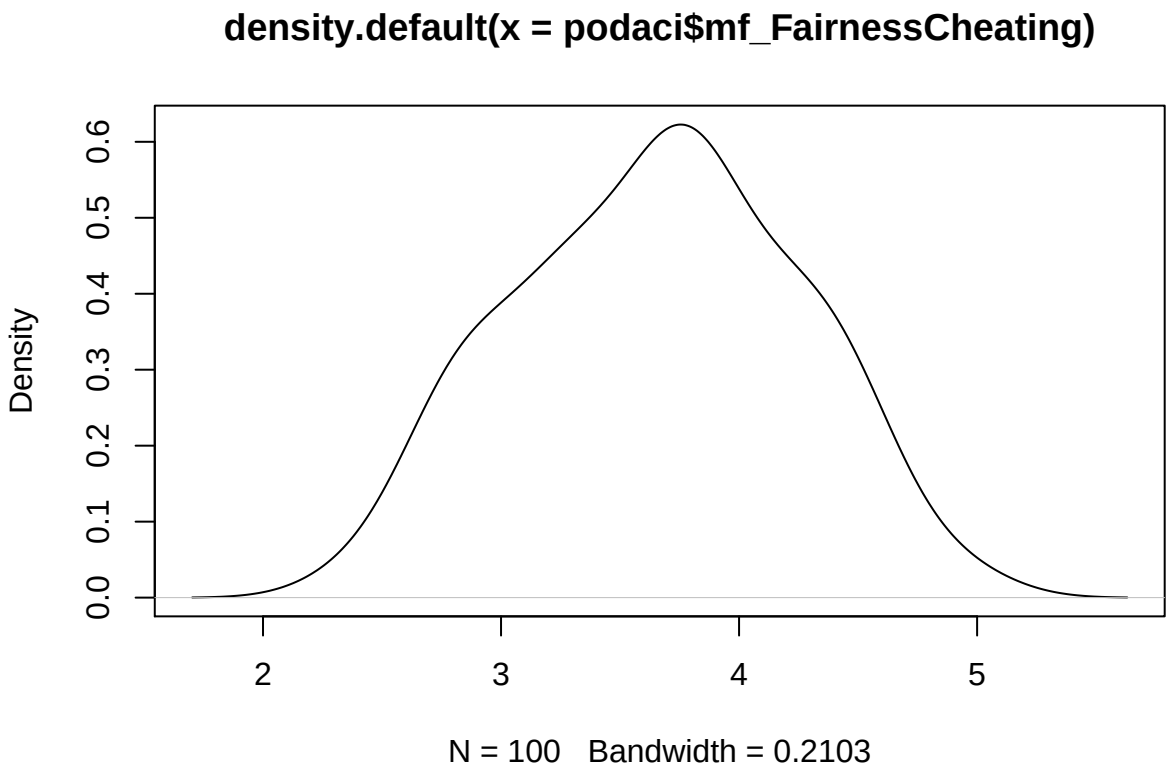
Histogram možemo dobiti pomoću funkcije `hist`.

```
podaci$mf_CareHarm %>% hist(.)
```



A možemo dobiti i graf gustoće distribucije tako da varijablu prvo bacimo u funkciju `density`, a potom u `plot`.

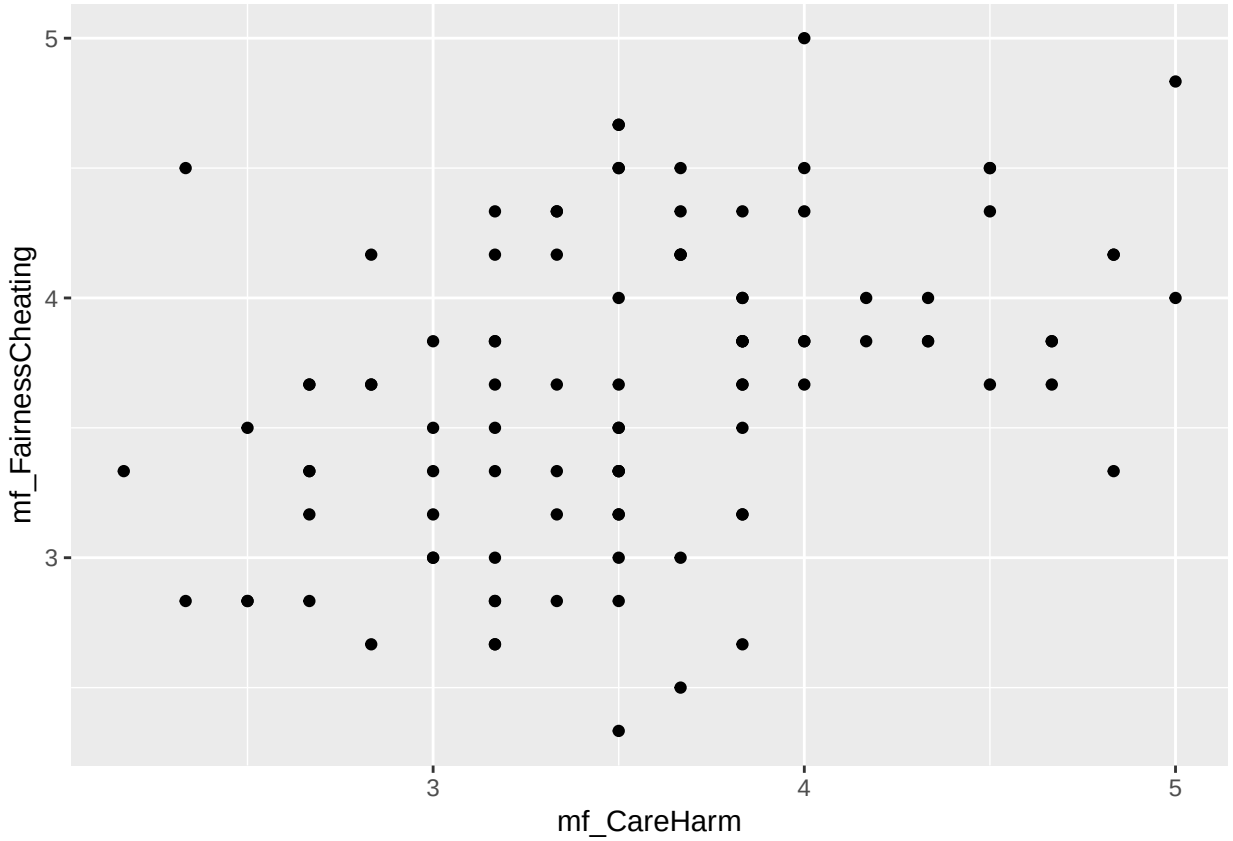
```
plot(density(podaci$mf_FairnessCheating))
```



Moje poznavanje base grafike staje otprilike ovdje jer za vizualizacije koristim paket `ggplot`, koji je nekad

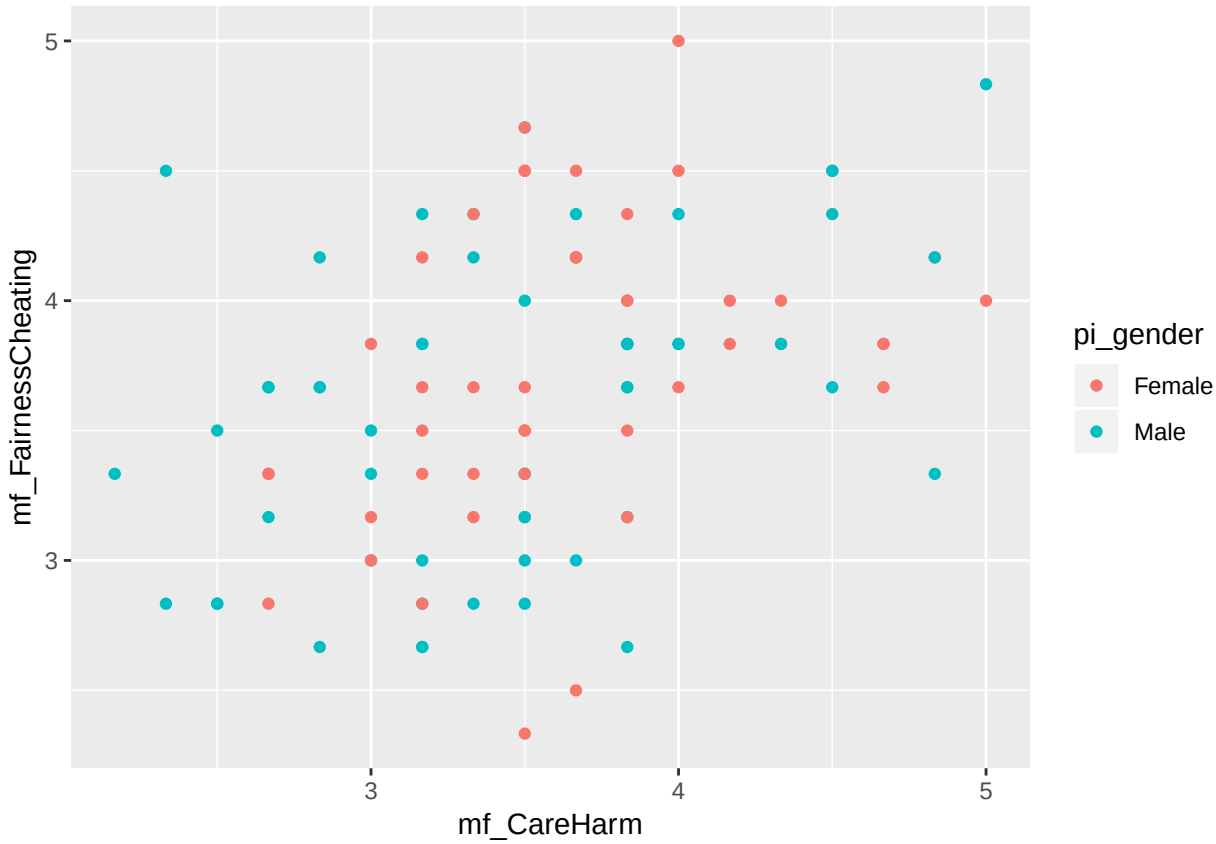
nešto zahtjevniji, ali je i dosta moćniji. `ggplot` dolazi s funkcijom `qplot` (*quick plot*), koja služi za brzinsko crtanje. Dijagram raspršenja, recimo, možemo dobiti isto kao i s base `plotom`, ali ovaj je nešto ljepši.

```
ggplot2::qplot(data = podaci, x = mf_CareHarm, y = mf_FairnessCheating)
```



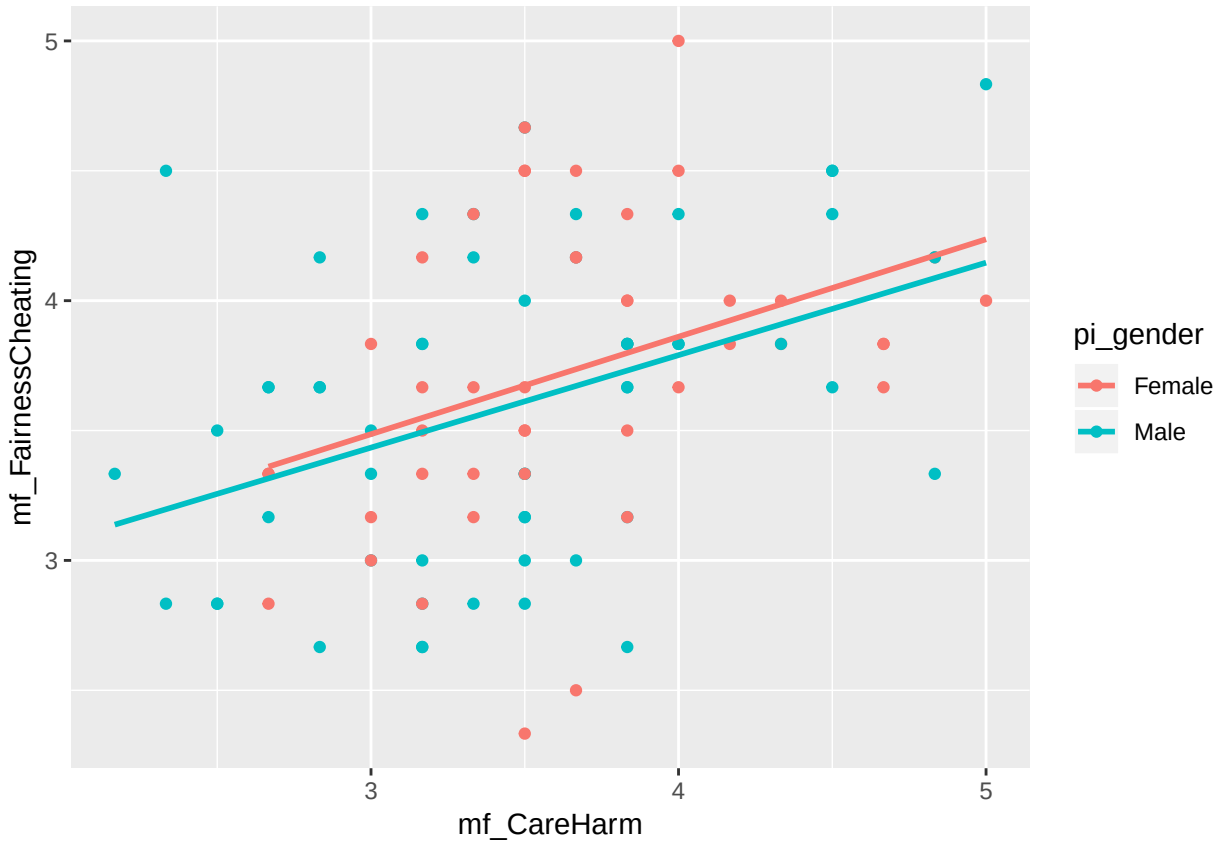
A lako možemo promijeniti boju točaka na temelju, recimo, spola.

```
ggplot2::qplot(data = podaci, x = mf_CareHarm, y = mf_FairnessCheating,
               color = pi_gender)
```



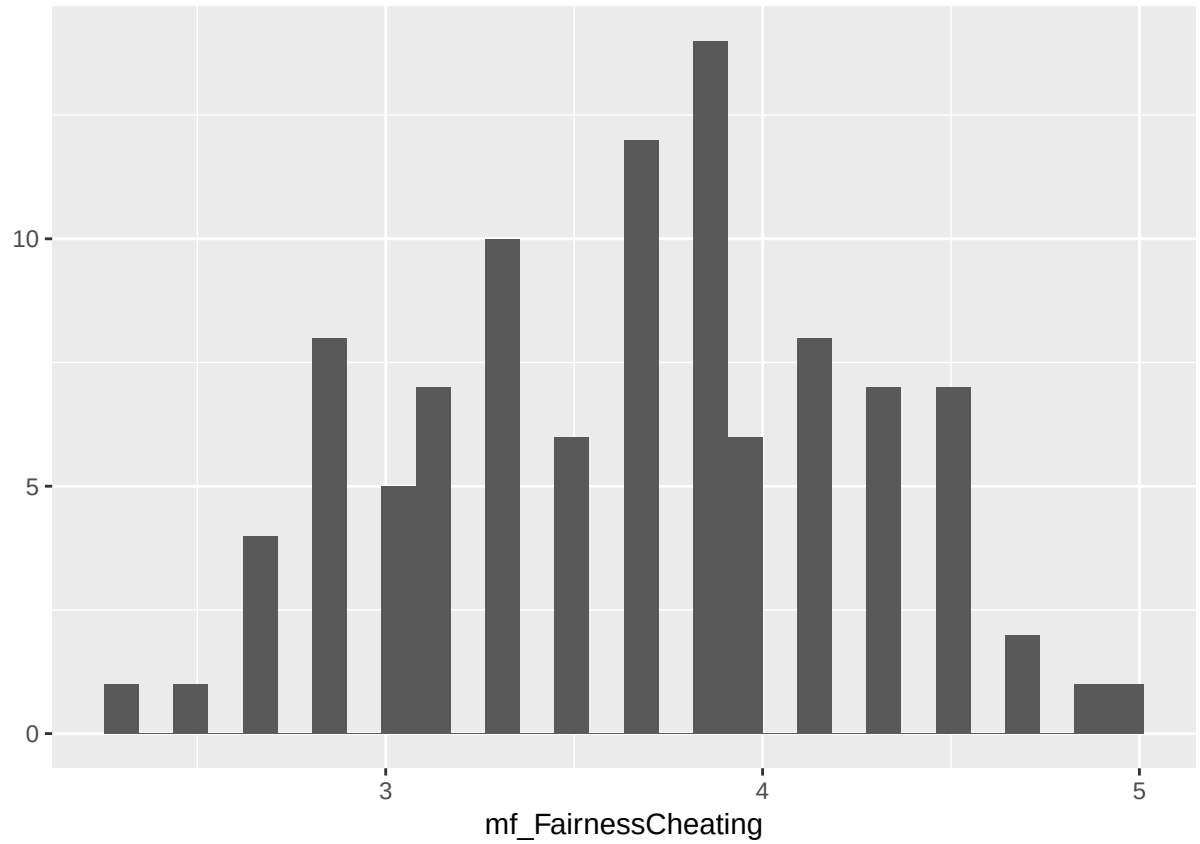
Fora kod `ggplota` je u tome da se graf izgrađuje sloj po sloj. `qplot` nešto skraćuje taj proces, ali i dalje ostavlja mogućnost dodavanja slojeva pomoću operatora `+`. Kad bismo, recimo, grafu još htjeli dodati regresijske pravce za svaku skupinu, na kraj bismo dodali:

```
ggplot2::qplot(data = podaci, x = mf_CareHarm, y = mf_FairnessCheating,
               color = pi_gender) + geom_smooth(method = 'lm', se = F)
```

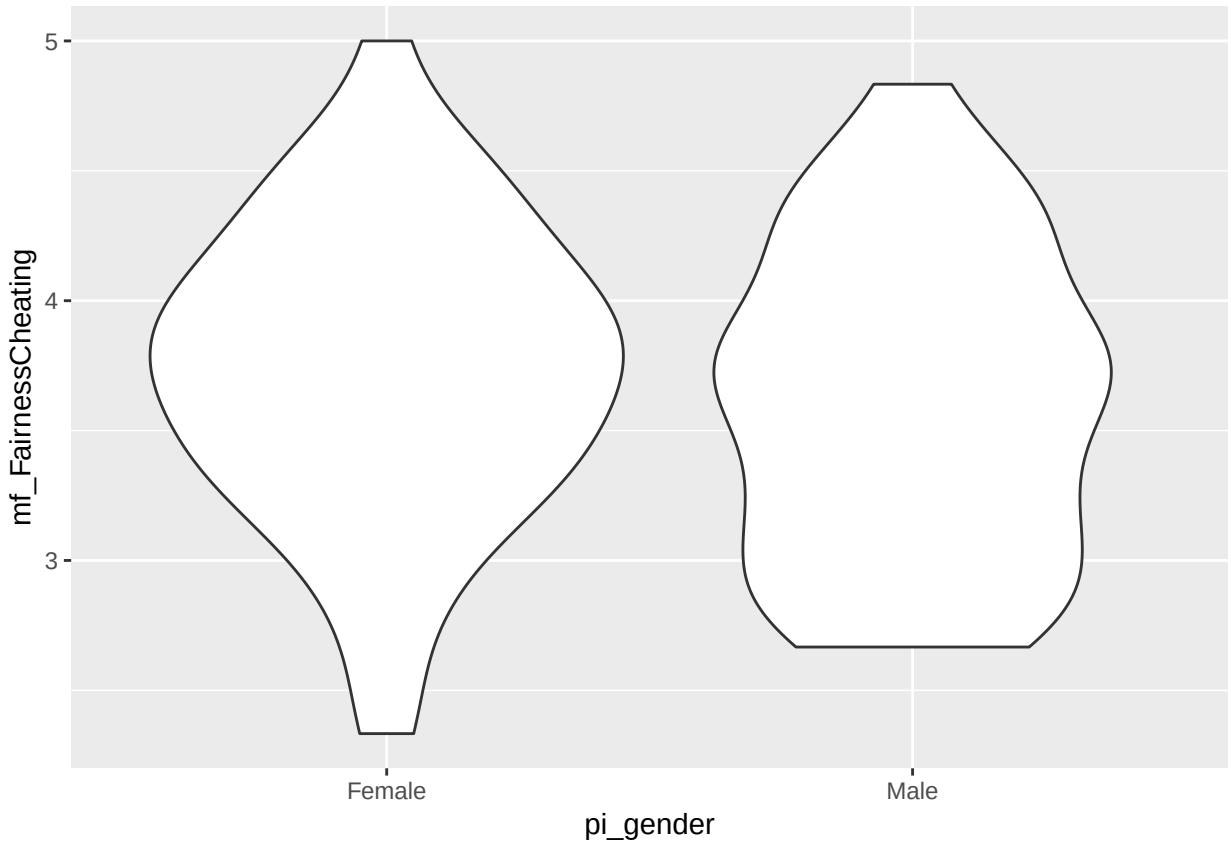


Evo i histograma (koji je meni, ovako po difoltu, ružniji od base R-ovog):

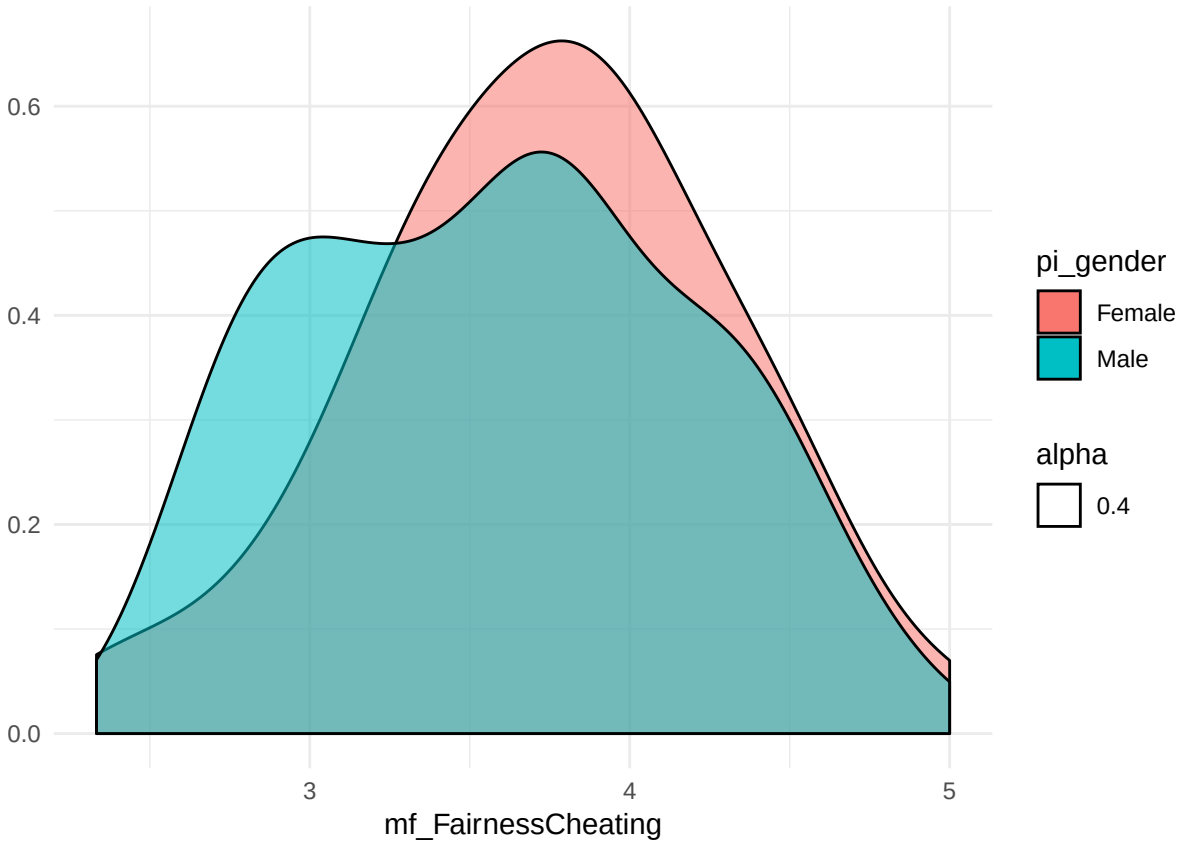
```
ggplot2::qplot(data = podaci, x = mf_FairnessCheating,
               geom = 'histogram')
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot2::qplot(data = podaci, x = pi_gender,
               y = mf_FairnessCheating, geom = 'violin')
```

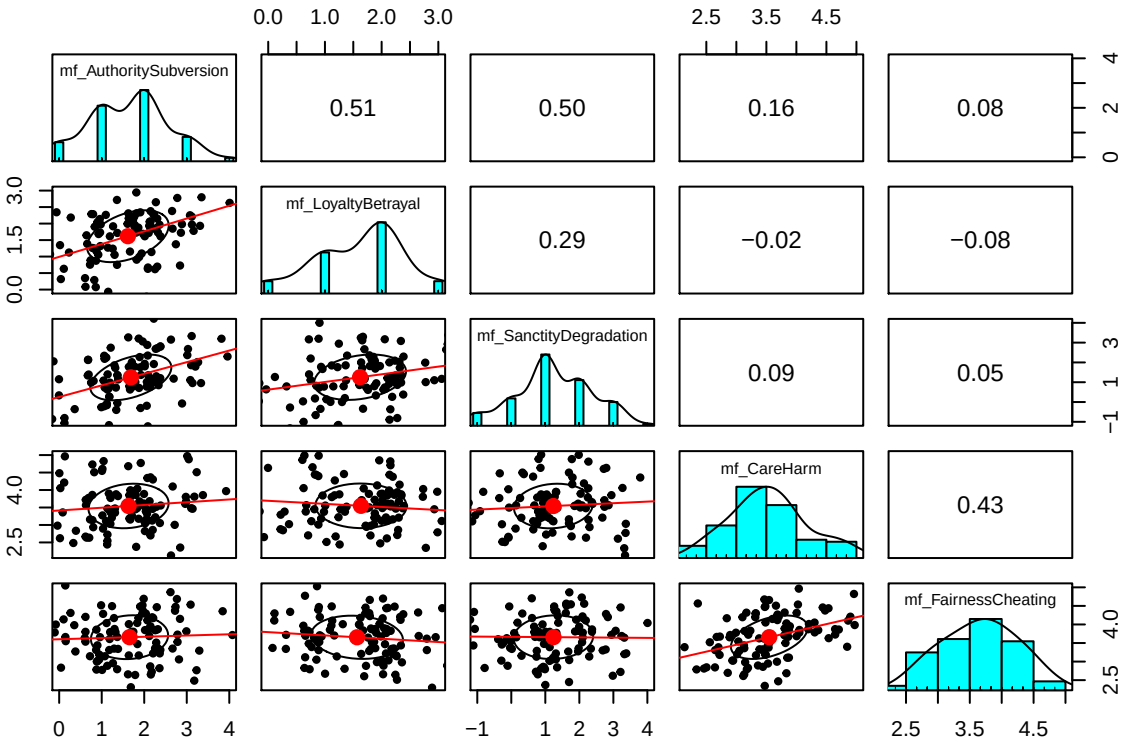



```
ggplot2::qplot(data = podaci, x = mf_FairnessCheating,
               geom = 'density', fill = pi_gender,
               # određuje razinu transparentnosti
               alpha = .4) + theme_minimal()
```



S `qplotom` (a pogotovo s `ggplotom`) se može puno igrati, tako da neću pretjerano nastavljati ovaj niz. Igranje prepuštam čitatelju. Za kraj ovog dijela, pogledat ćemo jednu funkciju iz `psych` paketa koja daje dosta opširan vizualni sažetak podatka.

```
podaci %>%
dplyr::select(., starts_with('mf_')) %>%
psych::pairs.panels(., lm = T, method = 'spearman',
                    # ovo je veličina slova, ne znam
                    # zašto se zove cex...
                    cex = .5,
                    # malo razbaca točke koje se nalaze
                    # na istoj poziciji radi dobivanja
                    # boljeg dojma o broju jedinki
                    jiggle = T)
```



Motivacijski primjeri - missing data

Za početak, ubacit ćemo neke missing vrijednosti (`NA`) u naš set podataka.

```
set.seed(151059)

podaci %>%
dplyr::select(., 1:moralIdentityInternalization04_rec) %>%
apply(., MARGIN = c(1,2), FUN = function(x) {
    if(runif(1) < .1) x <- NA
    else return(x)
}) %>%
# primijetiti da u ovom pozivu selecta nema točke!
# to je zato što select pozivamo na tablici 'podaci',
# a ne na tablici koju guramo kroz pipeline
cbind(., dplyr::select(podaci, starts_with('pi_'))) ->
podaci_na

head(podaci_na)
##   attitudesAndNorms01 attitudesAndNorms02 attitudesAndNorms03
## 1                   5                   5                   5
## 2                   5                   4                   2
## 3                   4                   6                   5
## 4                   6                   2                   3
## 5                  NA                   1                  NA
## 6                   4                   4                   4
##   attitudesAndNorms04 attitudesAndNorms05 attitudesAndNorms06
## 1                   5                   4                   3
## 2                   1                  NA                   2
```

```
## 3                    5                   NA                   NA
## 4                    2                    2                    3
## 5                    3                    1                    2
## 6                    3                    4                    3
##   attitudesAndNorms07 attitudesAndNorms08 callToAction donationMoney
## 1                   4                   NA            7           37
## 2                   3                    7            6           18
## 3                   4                   NA            7           NA
## 4                   5                    6            1           14
## 5                  NA                    5            8            0
## 6                   5                    5            7           37
##   donationTime descriptiveSocialNorms01 descriptiveSocialNorms02
## 1            4                       NA                       3
## 2            3                        3                       1
## 3            3                        3                       3
## 4            5                        1                       1
## 5            0                        3                       1
## 6            2                        1                       1
##   descriptiveSocialNorms03 descriptiveSocialNorms04 mf_AuthoritySubversion
## 1                        2                        2                      1
## 2                       NA                        1                      1
## 3                        2                       NA                      2
## 4                        2                        3                      2
## 5                       NA                        4                      2
## 6                        3                        2                      0
##   mf_LoyaltyBetrayal mf_SanctityDegradation mf_CareHarm
## 1                  2                     1    3.166667
## 2                  2                     1    3.000000
## 3                 NA                     1    3.500000
## 4                  3                     1    2.833333
## 5                  2                     1    3.166667
## 6                  1                    -1    2.500000
##   mf_FairnessCheating moralFoundations01_care moralFoundations02_fair
## 1            3.000000                       4                      4
## 2            3.000000                       3                      3
## 3            3.166667                       4                     NA
## 4            3.666667                      NA                      3
## 5            3.333333                       3                      1
## 6            2.833333                       4                      4
##   moralFoundations03_loyal moralFoundations04_author
## 1                        3                       1
## 2                       NA                       0
## 3                        2                       2
## 4                        1                       2
## 5                        1                      NA
## 6                        0                       0
##   moralFoundations05_sanct moralFoundations06_control
## 1                        2                        0
## 2                        2                        0
## 3                        1                        0
## 4                        3                        2
## 5                       NA                       -1
## 6                       -1                        1
```

```
##   moralFoundations07_care moralFoundations08_fair moralFoundations09_loyal
## 1                       4                       4                       3
## 2                       3                      NA                      NA
## 3                       4                       4                       2
## 4                      NA                       3                       4
## 5                       5                       3                      NA
## 6                       2                       4                       3
##   moralFoundations10_author moralFoundations11_sanct
## 1                         0                       1
## 2                        -1                      NA
## 3                         1                       1
## 4                         3                       0
## 5                         2                       1
## 6                         0                      -1
##   moralFoundations12_care moralFoundations13_fair moralFoundations14_loyal
## 1                       6                       3                       4
## 2                       5                       5                       2
## 3                       4                       4                       1
## 4                       5                      NA                       1
## 5                       4                       5                       3
## 6                       4                      NA                       3
##   moralFoundations15_author moralFoundations16_sanct
## 1                        NA                       3
## 2                         2                       1
## 3                         2                       1
## 4                         1                       2
## 5                         2                      -1
## 6                         3                       1
##   moralFoundations17_care moralFoundations18_fair moralFoundations19_loyal
## 1                       2                       2                       0
## 2                       5                       3                       2
## 3                       3                       3                       4
## 4                       4                       4                       2
## 5                       4                       5                       2
## 6                      NA                       2                       2
##   moralFoundations20_author moralFoundations21_sanct
## 1                         0                       0
## 2                        NA                       1
## 3                         0                       1
## 4                         4                       1
## 5                         1                       3
## 6                        NA                       3
##   moralFoundations22_control moralFoundations23_care
## 1                          4                       3
## 2                          4                       3
## 3                          6                       4
## 4                          4                       2
## 5                          4                       4
## 6                          3                       0
##   moralFoundations24_fair moralFoundations25_loyal
## 1                      NA                       0
## 2                       3                       0
## 3                       1                       1
```

```
## 4                        5                        2
## 5                        2                        0
## 6                        3                        3
##   moralFoundations26_author moralFoundations27_sanct
## 1                         1                        1
## 2                         1                        1
## 3                         1                        0
## 4                        NA                        1
## 5                         0                        1
## 6                         2                        1
##   moralFoundations28_care moralFoundations29_fair moralFoundations30_loyal
## 1                       0                       1                       1
## 2                      -1                       1                       1
## 3                       2                       3                       1
## 4                      -1                       2                       1
## 5                      -1                       4                       2
## 6                       1                       1                       1
##   moralFoundations31_author moralFoundations32_sanct
## 1                         3                        1
## 2                         2                        0
## 3                         1                        0
## 4                         5                        4
## 5                         2                        2
## 6                         2                       NA
##   moralIdentityInternalization01 moralIdentityInternalization02
## 1                              5                              2
## 2                              4                              3
## 3                              6                              5
## 4                              6                              4
## 5                              4                              3
## 6                              4                              6
##   moralIdentityInternalization03 moralIdentityInternalization04
## 1                              1                              2
## 2                              1                              3
## 3                              1                              1
## 4                              1                              3
## 5                              1                              2
## 6                             NA                              1
##   moralIdentityInternalization05 moralIdentityInternalization03_rec
## 1                              3                                  6
## 2                              4                                  6
## 3                              5                                  6
## 4                              4                                  6
## 5                              4                                  6
## 6                              4                                 NA
##   moralIdentityInternalization04_rec pi_age pi_education pi_gender
## 1                                  5      3     prof-dip      Male
## 2                                  4     20      masters      Male
## 3                                  6     20       hi-sch      Male
## 4                                  4     19      masters      Male
## 5                                  5     22      masters    Female
## 6                                  6     25      masters      Male
##                     pi_ideology pi_income pi_nationality
```

```
## 1 Neither liberal or conservative      avg-       american
## 2             Very liberal (left)      avg+       american
## 3 Neither liberal or conservative      avg+          other
## 4             Very liberal (left)      avg+       american
## 5             Very liberal (left)      avg+       american
## 6             Very liberal (left)      avg+       american
##   pi_previousDonations
## 1                Rarely
## 2             Regularly
## 3                Rarely
## 4                Rarely
## 5             Regularly
## 6             Regularly
```

Prethodni blok koda donosi neke novosti.

1) `set.seed` je funkcija kojom možemo *random number generator* R-a postaviti na neku vrijednost (*seed*). Po mojim saznanjima, to se uglavnom radi zato da bi se osigurala reproducibilnost stohastičkih procesa. Zbog toga bi obrazac `NA` vrijednosti koje vi dobivate trebao biti jednak onom koji ja dobivam.

2) `apply` je funkcija koja služi kao skraćenica za `for` petlju (koju nismo obradili, heh). `apply` prima (i) set podataka, (ii) funkciju `FUN` te (iii) `MARGIN`, koji određuje na što će se `FUN` primijenjivati, a može biti `1` (redovi), `2` (stupci) ili `c(1, 2)` (svaki pojedini element)

Sintaksa za `for` petlju je, inače:

```
for(i in 1:10) {
    print(i)
 }
```

tj.

```
for(nešto preko čega možemo iterirati, tj. prolaziti) {
naredbe
}
```

3) kao `FUN` smo dali anonimnu funkciju koju smo *ad hoc* definirali koristeći naredbu `function(...)`. Definirali smo funkciju koja prima samo jedan element (`x`), a koji predstavlja jedan podatak dohvaćen iz tablice koju smo proslijedili u `apply`. Da smo vrijednost `MARGIN` stavili na `1`, `x` bi bio red podataka, a da smo je stavili na `2`, stupac podataka.

4) vidjeli smo i R-ovu

```
if(logički uvjet) naredba
else druga naredba
```

sintaksu. Za `if ... else` je bitno znati da **ne može raditi s vektorima**, već samo s pojedinim elementima. `case_when` i `ifelse` (base R) su vektorizirane verzije `if ... else` naredbi.

Dosad, dakle, imamo kod koji je odabrao sve redove i samo neke stupce tablice `podaci`, te ih proslijedio u funkciju `apply`. `apply` potom uzima svaki pojedini element (odnosno, svaki pojedini podatak, odnosno svaku pojedinu vrijednost) i daje ga u funkciju koju smo sami definirali (koristeći `function`). Unutar funkcije, taj je element dostupan kao `x`.

Unutar funkcije, imamo `if ... else` izraz u kojem:

5) koristimo `runif(1)` (**random uniform**) kako bismo dobili **jednu** vrijednost u rasponu od 0 do 1, te provjeravamo je li ta vrijednost manja od `.1`. Ako jest, `if` uvjet se evaluira kao `TRUE` i u `x` upisujemo vrijednost `NA`, kojom R predstavlja vrijednosti koje nedostaju. S obzirom na `runif`, vjerojatnost upisivanja vrijednosti `NA` je 10%. Ako `runif` vrati vrijednost veću od `.1`, `if` uvjet se evaluira kao `FALSE`

te prelazimo na `else` dio. Naredba pod `else` je `return(x)`, što znači da u tom slučaju funkcija treba vratiti vrijednost `x`.

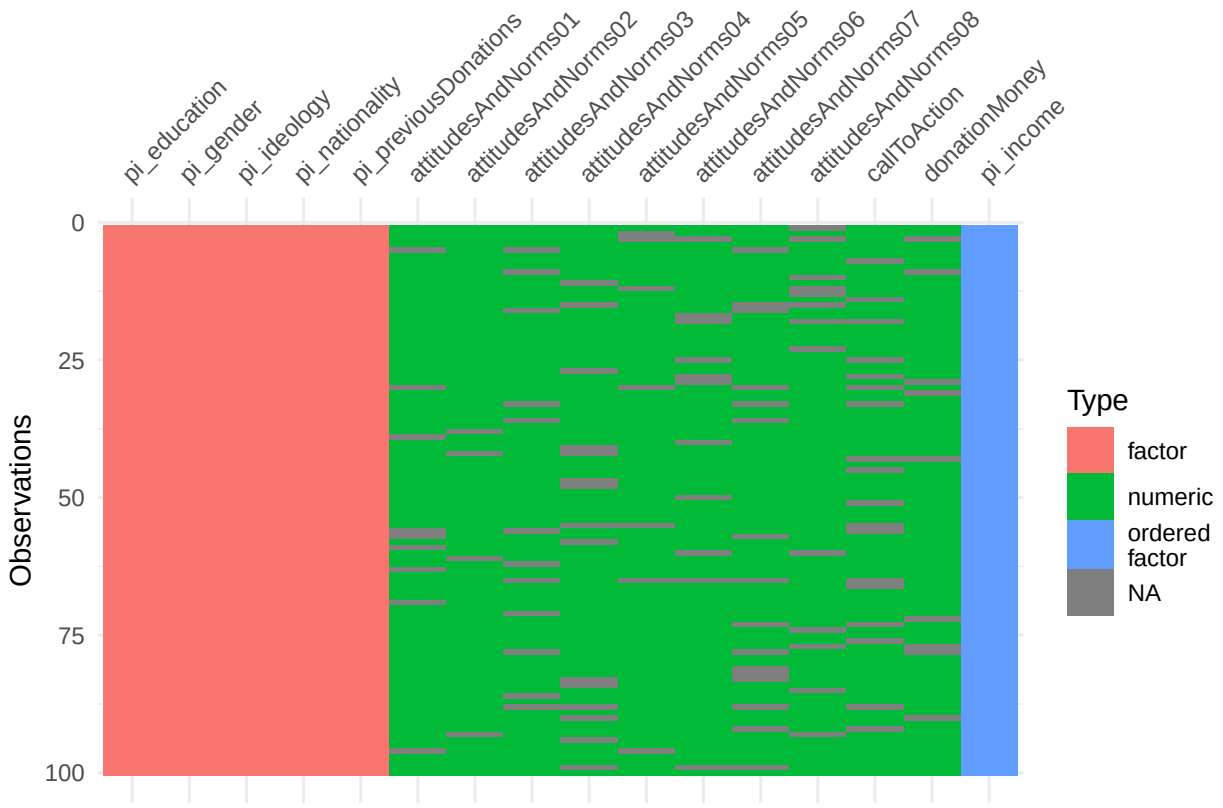
6) Budući da je `select` vratio tablicu koja ne sadrži `pi_` varijable, a koje želimo imati, koristimo `cbind` (*column bind*) kako bismo dodali i te varijable. Kao prvi argument, u `cbind` stavljamo `.`, dakle modificiranu tablicu koju smo provukli kroz `apply` (i kroz cijeli *pipeline*), a kao drugi argument dajemo tablicu s odabranim stupcima iz tablice `podaci`.

Na kraju, to spremamo kao `podaci_na`. Sad kad imamo set podataka koji sadrži, missing vrijednosti, bacit ćemo se na motiviranje. Za početak, pogledat ćemo funkciju koja nam omogućava da dobijemo brzinski pregled svojih podataka. `create_report` stvorit će interaktivni `.html` file koji sadrži hrpu deskriptivne statistike i neke zgodne grafove. Output se može jako prilagođavati dodavanjem argumenata, ali funkcija može biti zgodno-korisna i bez njih.

```
podaci %>%
dplyr::select(., contains('foundations')) %>%
DataExplorer::create_report(.)
```

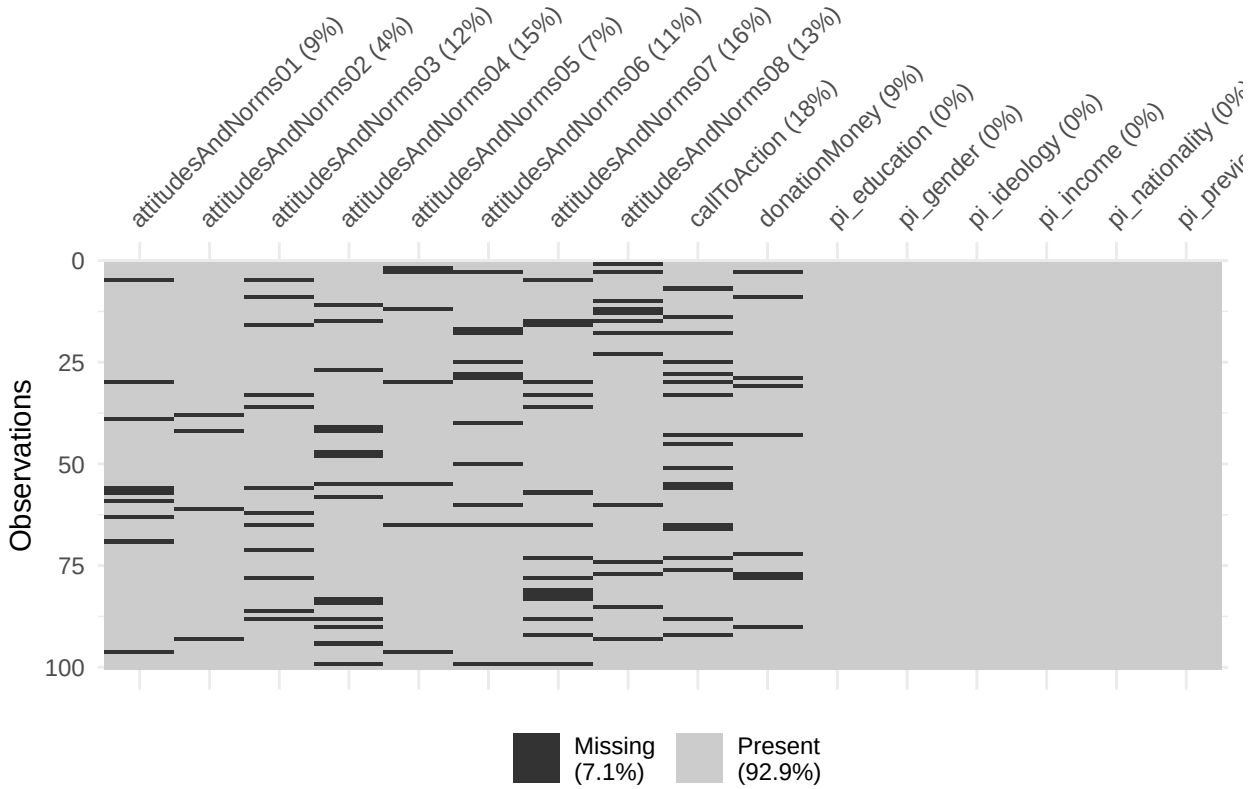
Sad ćemo pogledati neke funkcije koje nam olakšavaju pregledavanje obrazaca podataka koji nedostaju. Opći pregled stanja s podacima koji nedostaju možemo dobiti pomoću funkcije `vis_dat`. Radi preglednosti, ograničit ćemo se na prvih 10 i posljednjih 5 varijabli.

```
podaci_na %>%
# kako bismo odabrali posljednjih 5 stupaca,
# koristimo ncol za dobivanje broja stupaca te
# od vrijednosti koju dobijemo oduzimamo 5, a
# raspon proteženo do ncol. ovdje ncol vraća
# 66, pa efektivno imamo 61:65. oduzimanje od
# prvog poziva ncol mora biti u zagradama, inače
# error!
dplyr::select(., 1:10, (ncol(.)-5):ncol(.)) %>%
visdat::vis_dat(.)
```



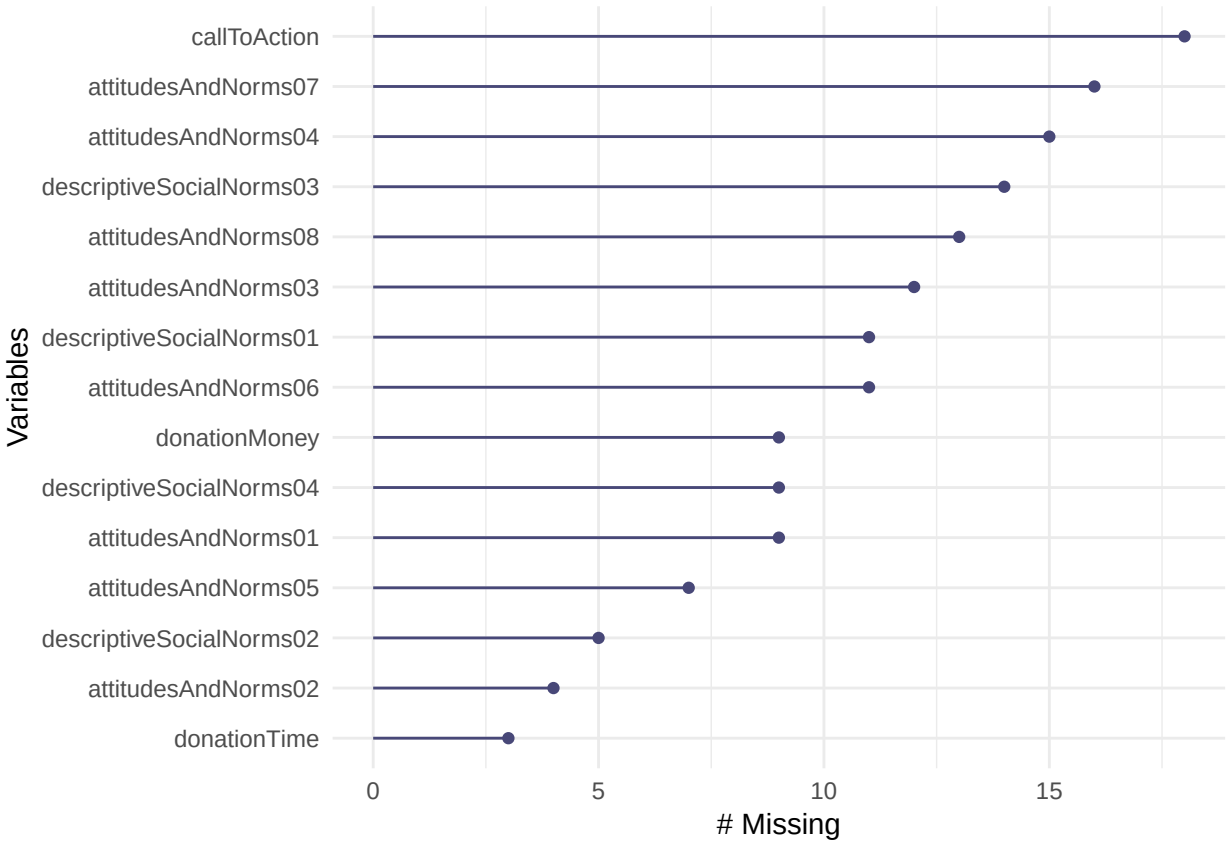
Ovaj graf prikazuje svakog pojedinog sudionika na y-osi, te svaki pojedinu varijablu na x-osi. Različite boje označavaju tip varijable (`factor`, `numeric`...) te `NA`, odnosno missing. Dakle, gdje god je nešto sivo, tamo nedostaje vrijednost. Funkcija `vis_miss` crta sličan graf, samo što ne označava tipove varijabli i govori nam koliki je postotak varijabli missing.

```
podaci_na %>%
dplyr::select(., 1:10, (ncol(.)-5):ncol(.)) %>%
visdat::vis_miss(.)
```



Još jedan grubi prikaz:

```
podaci_na %>%
dplyr::select(1:15) %>%
naniar::gg_miss_var(.)
```



Pomoću `n_miss` možemo dobiti broj vrijednosti koje nedostaju. Komplementarna funkcija je `n_complete`, koja, jel...

```
naniar::n_miss(podaci_na)
## [1] 594
naniar::n_complete(podaci_na)
## [1] 6006

podaci_na %>%
{naniar::n_miss(.) + naniar::n_complete(.) ==
    nrow(.) * ncol(.)}
## [1] TRUE
```

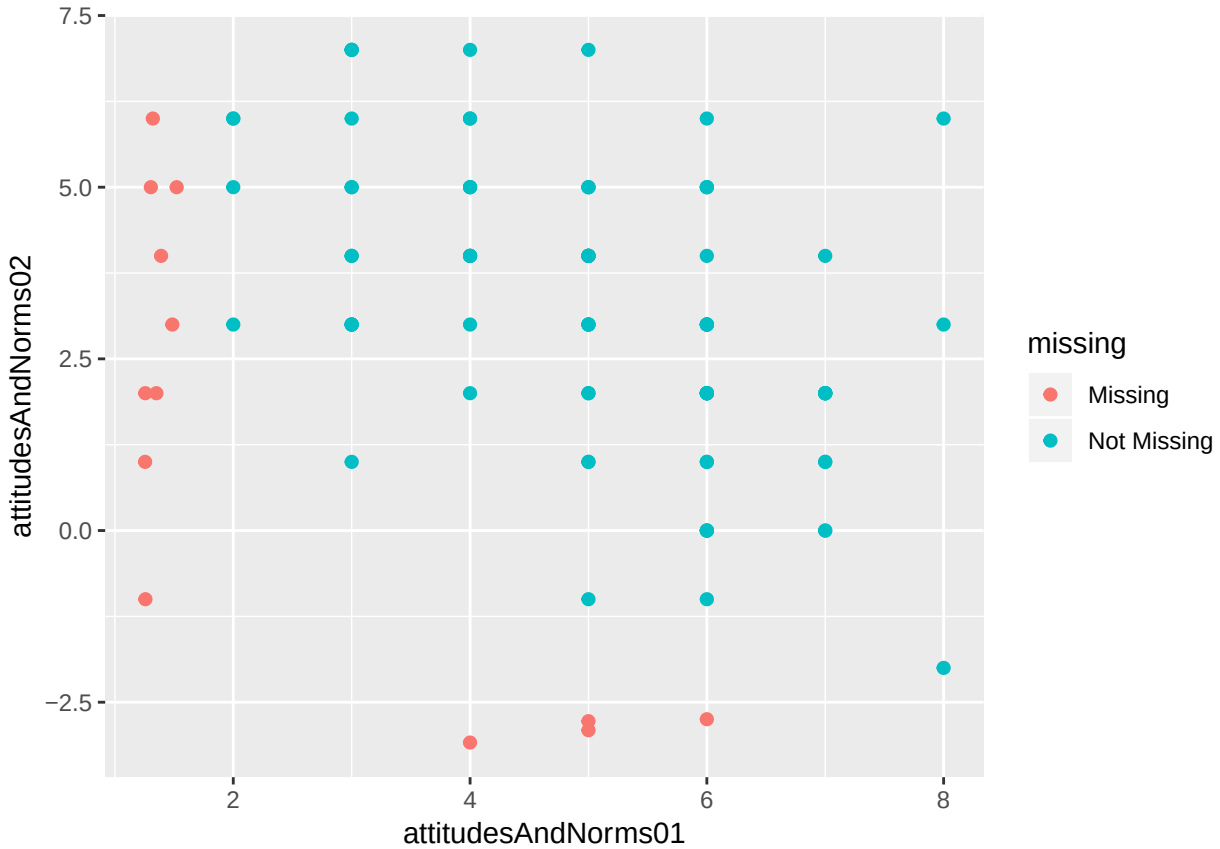
Pomoću `miss_case_summary` možemo dobiti informaciju o tome koliko svaki pojedini sudionik ima missinga.

```
podaci_na %>%
# gledamo samo prvih 20
slice(1:20) %>%
naniar::miss_case_summary(.)
## # A tibble: 20 x 3
##     case n_miss pct_miss
##    <int>  <int>    <dbl>
##  1    13     12    18.2
##  2     2      7    10.6
##  3     3      7    10.6
##  4     5      7    10.6
##  5    12      7    10.6
##  6    16      7    10.6
```

```
##  7    18      7    10.6
##  8     6      6     9.09
##  9     9      6     9.09
## 10     8      5     7.58
## 11    14      5     7.58
## 12    17      5     7.58
## 13    20      5     7.58
## 14     1      4     6.06
## 15     4      4     6.06
## 16    10      4     6.06
## 17    15      4     6.06
## 18    19      4     6.06
## 19    11      3     4.55
## 20     7      2     3.03
```

Ove funkcije su zgodne za opći pregled. Ako želimo pobliže ispitati obrasce nedostajućih podataka, trebamo ući dublje u odnose među pojedinim varijablama. Prva funkcija koja nam ovdje uskače upomoć dolazi iz `naniara`.

```
ggplot2::ggplot(podaci_na, aes(x = attitudesAndNorms01,
                      y = attitudesAndNorms02)) +
       naniar::geom_miss_point()
```



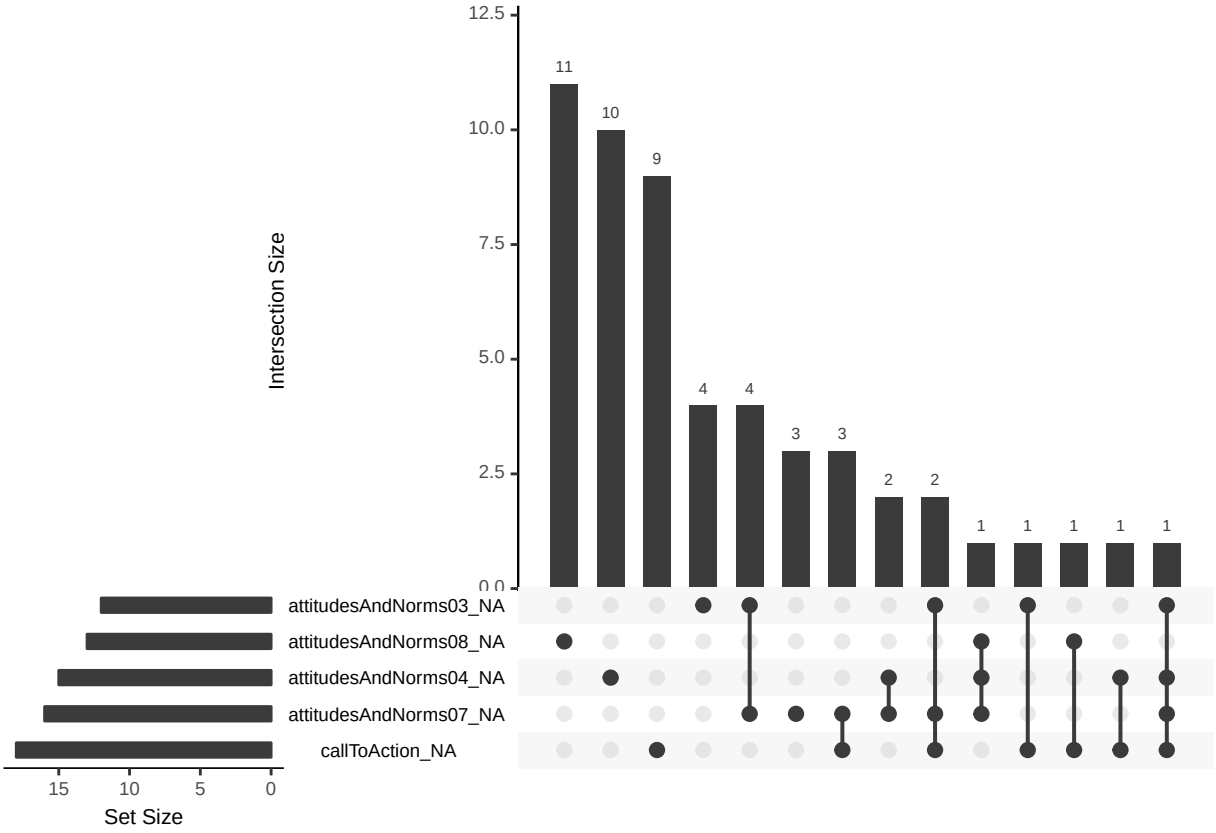
Točke označene kao `Missing` imaju vrijednost na osi (tj. varijabli) kojoj su priklonjene (tj. s kojom su paralelne), ali nemaju na varijabli na koju su okomite.

`Missing` točke nalaze se ispod minimuma koji vrijednosti dosežu na skali na kojoj nemaju rezultat. Da bude jasnije:

```
podaci_na %>%
dplyr::select(., attitudesAndNorms01, attitudesAndNorms02) %>%
summary(.)
##  attitudesAndNorms01 attitudesAndNorms02
##  Min.   :2.000       Min.   :-2.000
##  1st Qu.:4.000       1st Qu.: 2.000
##  Median :5.000       Median : 3.000
##  Mean   :5.055       Mean   : 3.198
##  3rd Qu.:6.000       3rd Qu.: 5.000
##  Max.   :8.000       Max.   : 7.000
##  NA's   :9           NA's   :4
```

Vidimo da je minimum na `attitudesAndNorms01` 2, a na `attitudesAndNorms02` -2. Missing vrijednosti na grafu se nalaze ispod tih vrijednosti. `gg_miss_upset` daje nam prikaz obrasca povezanosti missing vrijednosti kroz varijable.

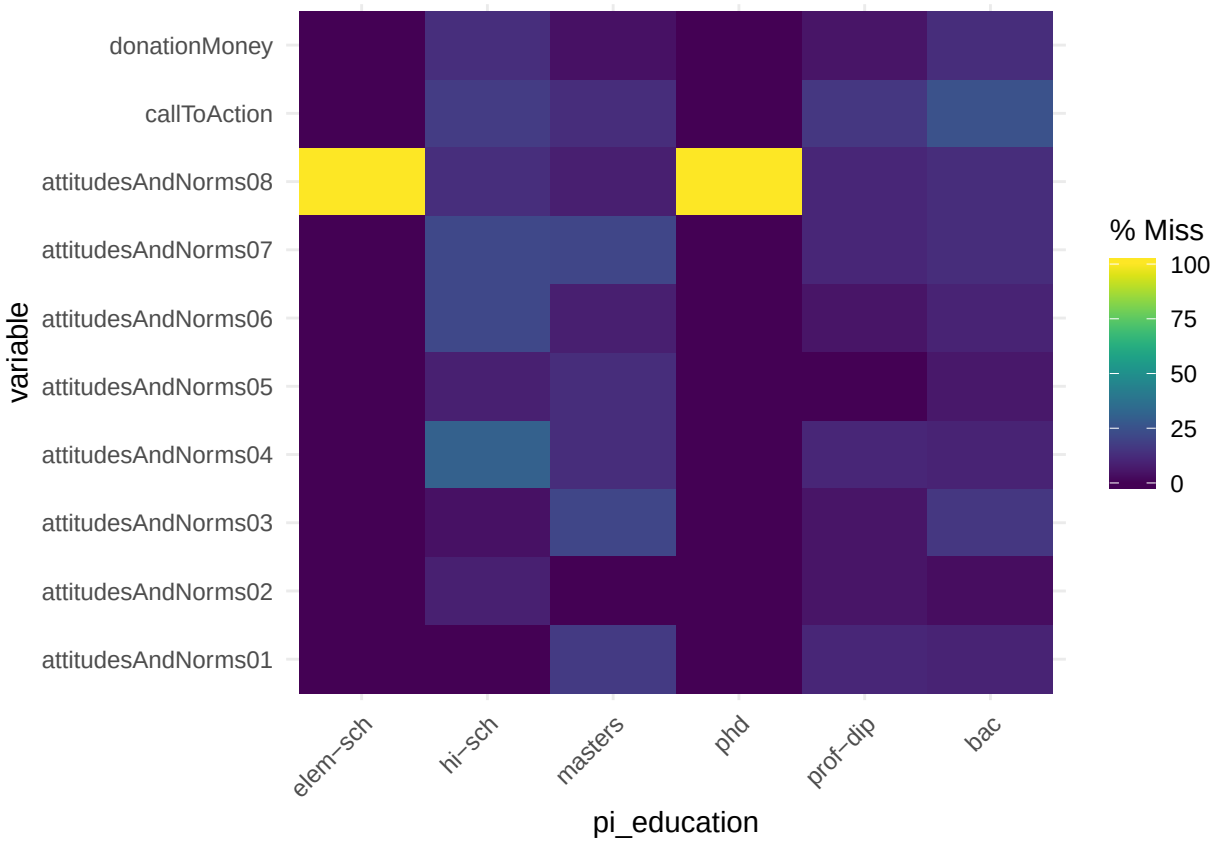
```
podaci_na %>%
dplyr::select(., 1:10) %>%
naniar::gg_miss_upset(., nsets = 5, nintersect = 18)
```



Pomoću `nsets = 5` ograničili smo se na 5 najkritičnijih varijabli. Vidimo da broj missing vrijednosti (na grafu označeno kao `Set Size`) pada od `callToAction_NA` prema `moralFoundatoins08_NA`. `nintersect` određuje koliko će *križanja* varijabli biti prikazano. Ova vrijednost trebala bi biti barem `nsets` + 1 da bi imala smisla. Okomiti stupci pokazuju nam koliko je missing vrijednosti u pojedinom križanju (uključujući i "križanja" jedne varijable). Uzmimo `callToAction_NA`, koji ima 18 vrijednosti koje nedostaju, odnosno nema 18 vrijednosti. Kad zbrojimo sve okomite stupce u kojima ta varijabla ima točku, doći ćemo do broja 18. Nekad je zgodno vidjeti razlikuju li se obrasci nedostajanja ovisno o nekoj kategoričkoj varijabli. U tu svrhu, možemo koristiti `gg_miss_fct`. Funkcija prima dva argumenta, neku tablicu s podacima i kategoričku varijablu na temelju

koje treba prikazati obrazac vrijednosti koje nedostaju.

```
podaci_na %>%
dplyr::select(., 1:10, pi_education) %>%
naniar::gg_miss_fct(., fct = pi_education)
```



Ovdje, recimo, možemo vidjeti da nitko ili gotovo nitko tko je završio osnovnu školu ili doktorat nije odgovorio na osmo pitanje u `attitudesAndNorms08`, što nije pretjerano zabrinjavajuće jer su podaci simulirani, ali bi se u stvarnoj situaciji čovjek možda htio zapitati.

Prtljanje po podacima iz SPSS-a za opće dobro

Kao što je na početku najavljeno, proći ćemo kroz R-ovsko prtljanje po korumpiranim podacima iz SPSS-a.

```
head(podaci_spss)
## # A tibble: 6 x 65
##   attitudesAndNor~ attitudesAndNor~ attitudesAndNor~ attitudesAndNor~
##              <dbl>            <dbl>            <dbl>            <dbl>
## 1                5                5                5                5
## 2                5                4                2                1
## 3                4                6                5                5
## 4                6                2                3                2
## 5                4                1                2                3
## 6                4                4                4                3
## # ... with 61 more variables: attitudesAndNorms05 <dbl>,
## #   attitudesAndNorms06 <dbl>, attitudesAndNorms07 <dbl>,
## #   attitudesAndNorms08 <dbl>, callToAction <dbl>,
```

```
## #   charitableBehavior01 <dbl>, charitableBehavior02 <dbl>,
## #   descriptiveSocialNorms01 <dbl>, descriptiveSocialNorms02 <dbl>,
## #   descriptiveSocialNorms03 <dbl>, descriptiveSocialNorms04 <dbl>,
## #   mf_AuthoritySubversion <dbl>, mf_CareHarm <dbl>,
## #   mf_FairnessCheating <dbl>, mf_LoyaltyBetrayal <dbl>,
## #   mf_SanctityDegradation <dbl>, moralFoundations01 <dbl>,
## #   moralFoundations02 <dbl>, moralFoundations03 <dbl>,
## #   moralFoundations04 <dbl>, moralFoundations05 <dbl>,
## #   moralFoundations06 <dbl>, moralFoundations07 <dbl>,
## #   moralFoundations08 <dbl>, moralFoundations09 <dbl>,
## #   moralFoundations10 <dbl>, moralFoundations11 <dbl>,
## #   moralFoundations12 <dbl>, moralFoundations13 <dbl>,
## #   moralFoundations14 <dbl>, moralFoundations15 <dbl>,
## #   moralFoundations16 <dbl>, moralFoundations17 <dbl>,
## #   moralFoundations18 <dbl>, moralFoundations19 <dbl>,
## #   moralFoundations20 <dbl>, moralFoundations21 <dbl>,
## #   moralFoundations22 <dbl>, moralFoundations23 <dbl>,
## #   moralFoundations24 <dbl>, moralFoundations25 <dbl>,
## #   moralFoundations26 <dbl>, moralFoundations27 <dbl>,
## #   moralFoundations28 <dbl>, moralFoundations29 <dbl>,
## #   moralFoundations30 <dbl>, moralFoundations31 <dbl>,
## #   moralFoundations32 <dbl>, moralIdentityInternalization01 <dbl>,
## #   moralIdentityInternalization02 <dbl>,
## #   moralIdentityInternalization03 <dbl>,
## #   moralIdentityInternalization04 <dbl>,
## #   moralIdentityInternalization05 <dbl>, pi_age <dbl>,
## #   pi_education <chr>, pi_gender <chr>, pi_ideology <chr>,
## #   pi_income <chr>, pi_nationality <chr>, pi_previousDonations <chr>,
## #   V65 <chr>
```

Prvo ćemo korumpirane redove izvući u novi `data.frame` - `podaci_spss_korumpirani`. To možemo napraviti tako da tražimo sve redove čiji unos pod `pi_gender` ne sadrži string *degree*. Dakle, koristit ćemo funkciju `filter`, u kojoj ćemo pozvati funkciju `str_detect`. Ona vraća vektor logičkih vrijednosti (`TRUE, FALSE, TRUE, ...`) koje `filter` može iskoristiti za, je li, filtriranje.

```
podaci_spss %>%
dplyr::filter(., str_detect(.$pi_gender, 'degree')) ->
podaci_spss_korumpirani
```

Zatim, filtriramo originalni `data.frame`, tako da u njemu ostaju samo redovi koji nisu korumpirani, odnosno oni koji nemaju string *degree*. Možemo samo kopipejstati raniji kod i pred poziv `str_detect` staviti uskličnik, koji je simbol za negaciju (pretvorit će vektor iz `TRUE, FALSE, TRUE, ...` u `FALSE, TRUE, FALSE, ...`). Također, mičemo upisivanje u novu varijablu, i pipu mijenjamo u assignment pipu.

```
podaci_spss %<>%
dplyr::filter(., !str_detect(.$pi_gender, 'degree'))
```

Sad trebamo popraviti unose u `podaci_spss_korumpirano` i popravljene unose prilijepiti na `podaci_spss`.

```
head(podaci_spss_korumpirani)
## # A tibble: 6 x 65
##   attitudesAndNor~ attitudesAndNor~ attitudesAndNor~ attitudesAndNor~
##              <dbl>            <dbl>            <dbl>            <dbl>
## 1                5                5                5                5
## 2                6                0                3                3
## 3                5                7                7                6
```

```
## 4                2                3                3                2
## 5                6                0                3                1
## 6                6                5                3                2
## # ... with 61 more variables: attitudesAndNorms05 <dbl>,
## #   attitudesAndNorms06 <dbl>, attitudesAndNorms07 <dbl>,
## #   attitudesAndNorms08 <dbl>, callToAction <dbl>,
## #   charitableBehavior01 <dbl>, charitableBehavior02 <dbl>,
## #   descriptiveSocialNorms01 <dbl>, descriptiveSocialNorms02 <dbl>,
## #   descriptiveSocialNorms03 <dbl>, descriptiveSocialNorms04 <dbl>,
## #   mf_AuthoritySubversion <dbl>, mf_CareHarm <dbl>,
## #   mf_FairnessCheating <dbl>, mf_LoyaltyBetrayal <dbl>,
## #   mf_SanctityDegradation <dbl>, moralFoundations01 <dbl>,
## #   moralFoundations02 <dbl>, moralFoundations03 <dbl>,
## #   moralFoundations04 <dbl>, moralFoundations05 <dbl>,
## #   moralFoundations06 <dbl>, moralFoundations07 <dbl>,
## #   moralFoundations08 <dbl>, moralFoundations09 <dbl>,
## #   moralFoundations10 <dbl>, moralFoundations11 <dbl>,
## #   moralFoundations12 <dbl>, moralFoundations13 <dbl>,
## #   moralFoundations14 <dbl>, moralFoundations15 <dbl>,
## #   moralFoundations16 <dbl>, moralFoundations17 <dbl>,
## #   moralFoundations18 <dbl>, moralFoundations19 <dbl>,
## #   moralFoundations20 <dbl>, moralFoundations21 <dbl>,
## #   moralFoundations22 <dbl>, moralFoundations23 <dbl>,
## #   moralFoundations24 <dbl>, moralFoundations25 <dbl>,
## #   moralFoundations26 <dbl>, moralFoundations27 <dbl>,
## #   moralFoundations28 <dbl>, moralFoundations29 <dbl>,
## #   moralFoundations30 <dbl>, moralFoundations31 <dbl>,
## #   moralFoundations32 <dbl>, moralIdentityInternalization01 <dbl>,
## #   moralIdentityInternalization02 <dbl>,
## #   moralIdentityInternalization03 <dbl>,
## #   moralIdentityInternalization04 <dbl>,
## #   moralIdentityInternalization05 <dbl>, pi_age <dbl>,
## #   pi_education <chr>, pi_gender <chr>, pi_ideology <chr>,
## #   pi_income <chr>, pi_nationality <chr>, pi_previousDonations <chr>,
## #   V65 <chr>
```

Sredit ćemo si `dput` output da si uštedimo neko tipkanje u kasnijim koracima.

```
podaci_spss_korumpirani %>%
dplyr::select(., pi_gender:V65) %>%
colnames(.) %>%
dput(.)
## c("pi_gender", "pi_ideology", "pi_income", "pi_nationality",
## "pi_previousDonations", "V65")
```

Za sređivanje stupaca možemo iskoristiti malo zgodnog prtljanja s `unite` i `separate` funkcijama. `unite` radi točno ono što nam treba - uzima n stupaca i spaja ih u jedan novi (onaj definiran pod arugmentom `col`). Vrijednosti iz različitih stupaca odvaja stringom navedenim pod `sep`. Po difoltu, stupci koje spajamo se brišu te ostaje samo novi stupac. To ponašanje možemo mijenjati putem argumenta `remove`. U ovom slučaju, to je okej, pa ćemo ostaviti difolt argument.

Želimo spojiti unose pod `pi_education` i `pi_gender` u `pi_education`. Kao `sep` ćemo koristiti prazan string. Dakle, radimo sljedeće:

```
podaci_spss_korumpirani %>%
tidyr::unite(.,
             # novi stupac
             col = pi_education,
             # stupci koje spajamo
             pi_education, pi_gender,
             # separator
             sep = '') %>%
# ovaj dio je samo za fokusiranje outputa
dplyr::select(., pi_education:ncol(.)) %>%
head(.)
## # A tibble: 6 x 6
##   pi_education pi_ideology pi_income pi_nationality pi_previousDona~ V65
##   <chr>        <chr>       <chr>     <chr>          <chr>            <chr>
## 1 "\"Some pro~ Male        Neither ~ Somewhat belo~ American         Rare~
## 2 "\"Some pro~ Female      Somewhat~ Somewhat abov~ American         Regu~
## 3 "\"Some pro~ Female      Very lib~ Somewhat belo~ Canadian         Rare~
## 4 "\"Some pro~ Female      Very lib~ Somewhat abov~ American         Often
## 5 "\"Some pro~ Female      Very lib~ Somewhat abov~ American         Often
## 6 "\"Some pro~ Female      Somewhat~ Much below th~ British          Rare~
```

Vidimo da su vrijednosti pod `pi_education` točne, ali sad imena varijabli ne odgovaraju njihovom sadržaju. Na primjer, spol se nalazi pod `pi_ideology`. Kako ne bismo morali mijenjati ime svake pojedine varijable, iskoristiti ćemo moći koje nam nudi funkcija `separate`. Ona uzima jedan stupac i razdvaja ga na n stupaca na temelju separatora `sep`. Novi stupci dobivaju imena definirana pod `into`. Stoga, možemo prvo uzeti sve preostale stupce - od `pi_ideology` do `V65` - i spojiti ih u jedan stupac - `tmp` (kao, *temporary*). Kao `sep` ćemo korsititi `@@`, budući da se taj string vrlo vjerojato neće naći nigdje u vrijednostima varijabli. Mogli bismo uzeti bilo koji drugi simbol za koji smo sigurni da se ne pojavljuje.

```
podaci_spss_korumpirani %>%
tidyr::unite(., col = pi_education,
             pi_education:pi_gender,
             sep = '') %>%
tidyr::unite(.,col = tmp,
             pi_ideology:V65,
             sep = '@@') %>%
dplyr::select(., pi_education:ncol(.)) %>%
head(.)
## # A tibble: 6 x 2
##   pi_education                  tmp
##   <chr>                         <chr>
## 1 "\"Some professional diploma~ Male@@Neither liberal or conservative@@So~
## 2 "\"Some professional diploma~ Female@@Somewhat liberal (left)@@Somewhat~
## 3 "\"Some professional diploma~ Female@@Very liberal (left)@@Somewhat bel~
## 4 "\"Some professional diploma~ Female@@Very liberal (left)@@Somewhat abo~
## 5 "\"Some professional diploma~ Female@@Very liberal (left)@@Somewhat abo~
## 6 "\"Some professional diploma~ Female@@Somewhat liberal (left)@@Much bel~
```

Dobili smo novi stupac `tmp` koji sadrži ružne stringove. Sad ćemo iskoristiti `separate` kako bismo vrijednosti u tom stupcu podijelili po separatoru `@@`. U argument `into` ćemo kopipejstati output funkcije `dput` koji smo ranije priredili, pri čemu ćemo obrisati posljednji unos (`V65`) jer to ne želimo gledati u konačnoj tablici.

```
podaci_spss_korumpirani %>%
tidyr::unite(., col = pi_education,
             pi_education:pi_gender,
```

```
            sep = '') %>%
tidyr::unite(.,col = tmp,
            pi_ideology:V65,
            sep = '@@') %>%
tidyr::separate(., col = tmp,
               # ovo je output funkcije dput koju smo
               # pozvali ranije, bez posljednjeg unosa,
               # V65
               into = c("pi_gender", "pi_ideology",
                        "pi_income", "pi_nationality",
                        "pi_previousDonations"),
               sep = '@@') %>%
dplyr::select(., pi_education:ncol(.)) %>%
head(.)
## # A tibble: 6 x 6
##   pi_education pi_gender pi_ideology pi_income pi_nationality
##   <chr>        <chr>     <chr>       <chr>     <chr>
## 1 "\"Some pro~ Male      Neither li~ Somewhat~ American
## 2 "\"Some pro~ Female    Somewhat l~ Somewhat~ American
## 3 "\"Some pro~ Female    Very liber~ Somewhat~ Canadian
## 4 "\"Some pro~ Female    Very liber~ Somewhat~ American
## 5 "\"Some pro~ Female    Very liber~ Somewhat~ American
## 6 "\"Some pro~ Female    Somewhat l~ Much bel~ British
## # ... with 1 more variable: pi_previousDonations <chr>
```

Sad imamo ispravno posložene stupce. Možemo maknuti nepotrebne navodnike iz unosa pod `pi_education` koristeći `mutate_at` da na taj stupac primijenimo funkciju `str_replace_all`.

```
podaci_spss_korumpirani %>%
tidyr::unite(., col = pi_education,
            pi_education:pi_gender,
            sep = '') %>%
tidyr::unite(.,col = tmp,
            pi_ideology:V65,
            sep = '@@') %>%
tidyr::separate(., col = tmp,
               into = c("pi_gender", "pi_ideology",
                        "pi_income", "pi_nationality",
                        "pi_previousDonations"),
               sep = '@@') %>%
dplyr::mutate_at(., .vars = vars(pi_education),
                 .f = stringr::str_replace_all,
                 pattern = '"', replacement = '') %>%
dplyr::select(., pi_education:ncol(.)) %>%
head(.)
## # A tibble: 6 x 6
##   pi_education pi_gender pi_ideology pi_income pi_nationality
##   <chr>        <chr>     <chr>       <chr>     <chr>
## 1 Some profes~ Male      Neither li~ Somewhat~ American
## 2 Some profes~ Female    Somewhat l~ Somewhat~ American
## 3 Some profes~ Female    Very liber~ Somewhat~ Canadian
## 4 Some profes~ Female    Very liber~ Somewhat~ American
## 5 Some profes~ Female    Very liber~ Somewhat~ American
## 6 Some profes~ Female    Somewhat l~ Much bel~ British
```

```
## # ... with 1 more variable: pi_previousDonations <chr>
```

Sad kad smo zadovoljni outputom našeg pipelinea, spremit ćemo promjene.

NB: Moramo maknuti zadnje dvije linije (koje smo koristili za fokusiranje outputa) jer će inače `podaci_spss_korumpirano` sadržavati samo ovo što vidimo gore.

```
podaci_spss_korumpirani %<>%
tidyr::unite(., col = pi_education,
             pi_education:pi_gender,
             sep = '') %>%
tidyr::unite(.,col = tmp,
             pi_ideology:V65,
             sep = '@@') %>%
tidyr::separate(., col = tmp,
                into = c("pi_gender", "pi_ideology",
                         "pi_income", "pi_nationality",
                         "pi_previousDonations"),
                sep = '@@') %>%
dplyr::mutate_at(., .vars = vars(pi_education),
                 .f = stringr::str_replace_all,
                 pattern = '"', replacement = '')

head(podaci_spss_korumpirani)
## # A tibble: 6 x 64
##   attitudesAndNor~ attitudesAndNor~ attitudesAndNor~ attitudesAndNor~
##              <dbl>            <dbl>            <dbl>            <dbl>
## 1                5                5                5                5
## 2                6                0                3                3
## 3                5                7                7                6
## 4                2                3                3                2
## 5                6                0                3                1
## 6                6                5                3                2
## # ... with 60 more variables: attitudesAndNorms05 <dbl>,
## #   attitudesAndNorms06 <dbl>, attitudesAndNorms07 <dbl>,
## #   attitudesAndNorms08 <dbl>, callToAction <dbl>,
## #   charitableBehavior01 <dbl>, charitableBehavior02 <dbl>,
## #   descriptiveSocialNorms01 <dbl>, descriptiveSocialNorms02 <dbl>,
## #   descriptiveSocialNorms03 <dbl>, descriptiveSocialNorms04 <dbl>,
## #   mf_AuthoritySubversion <dbl>, mf_CareHarm <dbl>,
## #   mf_FairnessCheating <dbl>, mf_LoyaltyBetrayal <dbl>,
## #   mf_SanctityDegradation <dbl>, moralFoundations01 <dbl>,
## #   moralFoundations02 <dbl>, moralFoundations03 <dbl>,
## #   moralFoundations04 <dbl>, moralFoundations05 <dbl>,
## #   moralFoundations06 <dbl>, moralFoundations07 <dbl>,
## #   moralFoundations08 <dbl>, moralFoundations09 <dbl>,
## #   moralFoundations10 <dbl>, moralFoundations11 <dbl>,
## #   moralFoundations12 <dbl>, moralFoundations13 <dbl>,
## #   moralFoundations14 <dbl>, moralFoundations15 <dbl>,
## #   moralFoundations16 <dbl>, moralFoundations17 <dbl>,
## #   moralFoundations18 <dbl>, moralFoundations19 <dbl>,
## #   moralFoundations20 <dbl>, moralFoundations21 <dbl>,
## #   moralFoundations22 <dbl>, moralFoundations23 <dbl>,
## #   moralFoundations24 <dbl>, moralFoundations25 <dbl>,
## #   moralFoundations26 <dbl>, moralFoundations27 <dbl>,
```

```
## #   moralFoundations28 <dbl>, moralFoundations29 <dbl>,
## #   moralFoundations30 <dbl>, moralFoundations31 <dbl>,
## #   moralFoundations32 <dbl>, moralIdentityInternalization01 <dbl>,
## #   moralIdentityInternalization02 <dbl>,
## #   moralIdentityInternalization03 <dbl>,
## #   moralIdentityInternalization04 <dbl>,
## #   moralIdentityInternalization05 <dbl>, pi_age <dbl>,
## #   pi_education <chr>, pi_gender <chr>, pi_ideology <chr>,
## #   pi_income <chr>, pi_nationality <chr>, pi_previousDonations <chr>
```

Za kraj, trebamo ovu tablicu pripojiti tablici `podaci_spss`. To možemo učiniti pomoću funkcije `rbind` (*rows bind*).

```
dim(podaci_spss)
## [1] 81 65
dim(podaci_spss_korumpirani)
## [1] 19 64
```

Vidimo da ove dvije tablice imaju različit broj stupaca. To je zato jer `podaci_spss` i dalje imaju varijablu `V65`, koja je u toj tablici prazna. Obrisat ćemo je.

```
podaci_spss$V65 <- NULL

dim(podaci_spss)
## [1] 81 64
dim(podaci_spss_korumpirani)
## [1] 19 64
```

Sad možemo spojiti te dvije tablice.

```
podaci_spss %<>%
rbind(., podaci_spss_korumpirani)
```

Reference i dodatna literatura

Grolemund, G. i Wickham, H. *R for data science*. O'Reilly Media, Inc.

Michael Crawley (2012). *The R Book*.

Pipe

- https://cran.r-project.org/web/packages/magrittr/vignettes/magrittr.html
- http://r4ds.had.co.nz/pipes.html

Regularni izrazi

- jako dobar šalabahter
- još jedan
- stranica koja omogućuje isprobavanje različitih uzoraka na tekstu
- uvod u `stringr`

Data wrangling (dplyr i srodno):

- prvi od četiri dijela (linkovi na druge na dnu stranice) blogova o formatiranju podataka

Korisni savjeti za organizaciju podataka u tablicama

- Broman, K. W., & Woo, K. H. (2018). Data organization in spreadsheets. *The American Statistician, 72*(1), 2–10.

`naniar`:

- intro
- galerija vizualizacija

Šalabahteri (obavezno skinuti!)

- obavezno!

Pretvaranje `.sav` fileova u `.csv`

- https://pspp.benpfaff.org/

Epilog

```
sessionInfo()
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Manjaro Linux
##
## Matrix products: default
## BLAS: /usr/lib/libblas.so.3.8.0
## LAPACK: /usr/lib/liblapack.so.3.8.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=hr_HR.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=hr_HR.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=hr_HR.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=hr_HR.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2.2   here_0.1         wrapr_1.7.0      readxl_1.1.0
##  [5] haven_1.1.2      conflicted_1.0.1 magrittr_1.5     forcats_0.3.0
##  [9] stringr_1.3.1    dplyr_0.7.7      purrr_0.2.5      readr_1.1.1
## [13] tidyr_0.8.1      tibble_1.4.2     ggplot2_3.0.0    tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.19      lubridate_1.7.4   lattice_0.20-35
##  [4] assertthat_0.2.0  rprojroot_1.3-2   digest_0.6.18
##  [7] psych_1.8.10      utf8_1.1.4        R6_2.3.0
## [10] cellranger_1.1.0  plyr_1.8.4        backports_1.1.2
## [13] visdat_0.5.1      evaluate_0.12     httr_1.3.1
## [16] pillar_1.3.0      rlang_0.3.0       lazyeval_0.2.1
## [19] rstudioapi_0.8    rmarkdown_1.10    labeling_0.3
## [22] foreign_0.8-71    munsell_0.5.0     broom_0.5.0
## [25] compiler_3.5.1    modelr_0.1.2      janitor_1.1.1
## [28] pkgconfig_2.0.2   mnormt_1.5-5      htmltools_0.3.6
## [31] tidyselect_0.2.5  gridExtra_2.3     fansi_0.4.0
## [34] viridisLite_0.3.0 crayon_1.3.4      withr_2.1.2
## [37] grid_3.5.1        nlme_3.1-137      jsonlite_1.5
```

```
## [40] gtable_0.2.0      scales_1.0.0      cli_1.0.1
## [43] stringi_1.2.4     viridis_0.5.1     snakecase_0.9.2
## [46] xml2_1.2.0        naniar_0.4.1      tools_3.5.1
## [49] glue_1.3.0        hms_0.4.2         parallel_3.5.1
## [52] yaml_2.2.0        colorspace_1.3-2  UpSetR_1.3.3
## [55] rvest_0.3.2       memoise_1.1.0     knitr_1.20
## [58] bindr_0.1.1
```