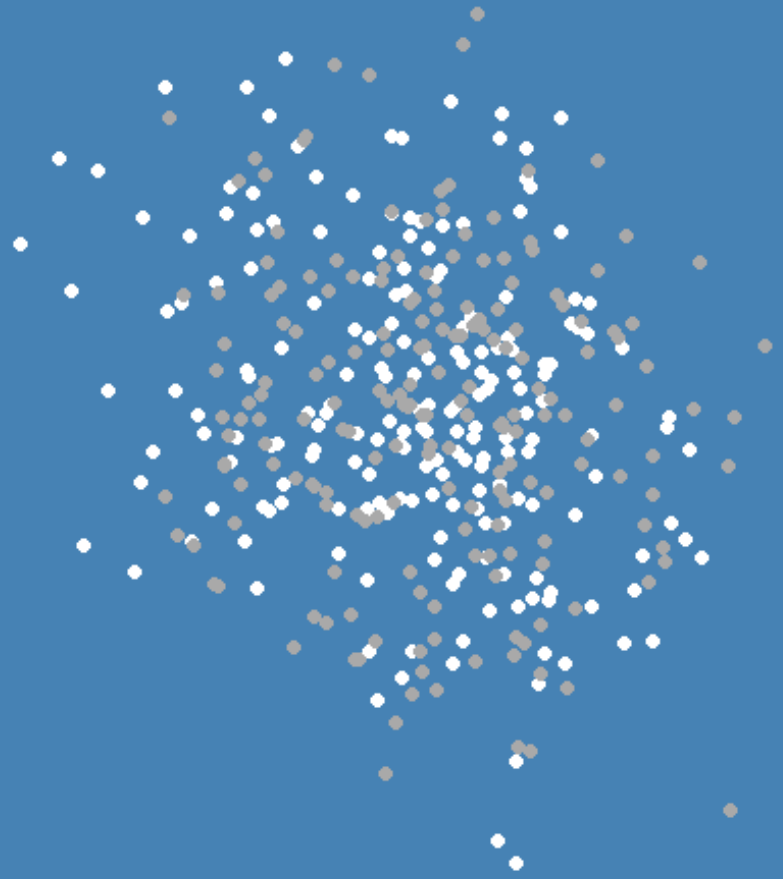
Introduction to R

2.8 Missing Values

Recoding Values to Missings, Computations Based on Vectors With Missing Values, Listwise Deletion

Lion Behrens, M.Sc.



University of Mannheim
Chair of Social Data Science and Methodology
Chair of Quantitative Methods in the Social Sciences

What Are the Issues With Missing Values?

When performing statistical analyses, our data often do not come in a perfectly clean format.

Rather, they often include missing values.

```
library(haven)
ess10 <- haven::read_dta("./dat/ESS10.dta")
table(ess10$stfgov, useNA = "always")
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10 <NA>
## 2171 1104 1819 2024 1747 2695 1786 1965 1418  536  375  420
```

Getting an Overview of the Extent of Missings: skimr

```
library(skimr)
skim_tee(ess10$stfgov)
```

```
## ── Data Summary ────────────────────────
##                            Values
## Name                       data
## Number of rows             18060
## Number of columns          1
## _______________________
## Column type frequency:
##   numeric                  1
## _______________________
## Group variables            None
##
## ── Variable type: numeric ─────────────────────────────────────────────────────
##   skim_variable n_missing complete_rate mean   sd p0 p25 p50 p75 p100 hist
## 1 data                420         0.977 4.29 2.71  0   2   4   6   10 ▇▆▇▆▂
```

Recoding Values to Missings

Recoding Values to Missings

Sometimes, supposedly valid entries in our variables are actually missings.

```
print(ess10$stfgov[1:10])
```

```
## <labelled<double>[10]>: How satisfied with the national government
##  [1] 4 3 2 2 9 0 3 1 2 0
##
## Labels:
##  value                   label
##      0 Extremely dissatisfied
##      1                      1
##      2                      2
##      3                      3
##      4                      4
##      5                      5
##      6                      6
##      7                      7
##      8                      8
##      9                      9
##     10    Extremely satisfied
##     77                Refusal
##     88             Don't know
##     99              No answer
```

Recoding Values to Missings

How did recoding work in dplyr?

```
ess10 <- ess10 %>%
  mutate(lr_binary = as.numeric(lrscale)) %>%
  mutate(lr_binary = recode(lr_binary,
                            `0` = 0,
                            `1` = 0,
                            `2` = 0,
                            `3` = 0,
                            `4` = 0,
                            `5` = 1,
                            `6` = 2,
                            `7` = 2,
                            `8` = 2,
                            `9` = 2,
                            `10` = 2
                            )
  )
```

Recoding Values to Missings: dplyr

How do we set supposedly valid values to missings?

```
ess10 <- ess10 %>%
  mutate(stfgov_adap = na_if(stfgov, 10))
```

```
table(ess10$stfgov, useNA = "always")
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10 <NA>
## 2171 1104 1819 2024 1747 2695 1786 1965 1418  536  375  420
```

```
table(ess10$stfgov_adap, useNA = "always")
```

```
##
##    0    1    2    3    4    5    6    7    8    9 <NA>
## 2171 1104 1819 2024 1747 2695 1786 1965 1418  536  795
```

Recoding Values to Missings: base R

How did recoding work in base R?

```
ess10$voted <- NA
ess10$voted[ess10$vote == 1] <- "Yes"
ess10$voted[ess10$vote == 2] <- "No"
ess10$voted[ess10$vote == 3] <- "Not eligible"
ess10$voted <- as.factor(ess10$voted)
```

Recoding Values to Missings: base R

How do we set supposedly valid values to missings?

```
ess10$stfgov_adap2 <- ess10$stfgov
ess10$stfgov_adap2[ess10$stfgov == 10] <- NA
```

```
table(ess10$stfgov_adap, useNA = "always")
```

```
##
##    0    1    2    3    4    5    6    7    8    9 <NA>
## 2171 1104 1819 2024 1747 2695 1786 1965 1418  536  795
```

```
table(ess10$stfgov_adap2, useNA = "always")
```

```
##
##    0    1    2    3    4    5    6    7    8    9 <NA>
## 2171 1104 1819 2024 1747 2695 1786 1965 1418  536  795
```

Computations Based on Vectors With Missing Values

Computations Based on Vectors With Missing Values

Many of R's built in functions won't work if vectors/variables include missing values.

```
mean(ess10$stfgov)
```

```
## [1] NA
```

```
sd(ess10$stfgov)
```

```
## [1] NA
```

```
mean(ess10$stfgov, na.rm = T)
```

```
## [1] 4.289456
```

```
sd(ess10$stfgov, na.rm = T)
```

```
## [1] 2.710664
```

Excluding Observations With Missing Values

Listwise Deletion: dplyr

Let's say we want to delete all observations (rows) from our dataset that include missing values on specific variables.

In dplyr, this can be accomplished by drop_na().

```
dim(ess10)
```

```
## [1] 18060   515
```

```
ess10 <- ess10 %>%
  drop_na(gndr, stfgov, ppltrst)
```

```
dim(ess10)
```

```
## [1] 17580   515
```

Listwise Deletion: base R

How to do this in base R?

```
ess10 <- ess10[complete.cases(ess10[,c("gndr", "stfgov", "ppltrst")]),]
```

```
dim(ess10)
```

```
## [1] 17580   515
```

Listwise Deletion: base R

We can even drop all cases that have a missing value *somewhere*...

```
ess10 <- na.omit(ess10)
```

```
dim(ess10)
```

```
## [1]   0 515
```

Data Wrangling Pipeline (I/III)

```
library(tidyverse)
ess10 <- haven::read_dta("./dat/ESS10.dta")
ess10 <- ess10 %>% # subset variables
  select(country = cntry, # sociodemographics
         gender = gndr,
         education_years = eduyrs,
         trust_social = ppltrst, # multidimensional trust
         trust_parliament = trstprl,
         trust_legalSys = trstlgl,
         trust_police = trstplc,
         trust_politicians = trstplt,
         trust_parties = trstprt,
         trust_EP = trstep,
         trust_UN = trstun,
         left_right = lrscale, # attitudes
         life_satisfaction = stflife,
         pol_interest = polintr,
         voted = vote, # turnout
         party_choice = prtvtefr # party choice
         ) %>%
  mutate_at(c("country", "gender", "voted", "party_choice"), as.character) %>% # change types
  mutate_at("pol_interest", as.numeric) %>% # change types
  filter(country == "FR") # subset cases (only include France)
```

Data Wrangling Pipeline (II/III)

```
ess10 <- ess10 %>%
  mutate(gender = recode_factor(gender,
                                `1` = "Male",
                                `2` = "Female"),
         voted = recode_factor(voted,
                               `1` = "Yes",
                               `2` = "No",
                               `3` = "Not eligible"),
         party_choice = recode_factor(party_choice,
                                      `1`  = "Lutte Ouvriére",
                                      `2`  = "Nouv. Parti Anti-Capitaliste",
                                      `3`  = "Parti Communiste Français",
                                      `4`  = "La France Insoumise",
                                      `5`  = "Parti Socialiste",
                                      `6`  = "Europe Ecologie Les Verts",
                                      `7`  = "La République en Marche",
                                      `8`  = "Mouvement Démocrate",
                                      `9`  = "Les Républicains",
                                      `10` = "Debout la France",
                                      `11` = "Front National",
                                      `12` = "Other",
                                      `13` = "Blank",
                                      `14` = "Null")
         )
```

Data Wrangling Pipeline (III/III)

```
ess10 <- ess10 %>%
  mutate(education_years = na_if(education_years, 114), # set 114 to missing
         pol_interest = (pol_interest * -1) + 5, # invert scale
         life_satisfaction = life_satisfaction + 1 # change scale to [1, 11]
         ) %>%
  drop_na(trust_politicians, gender, education_years,
          life_satisfaction, pol_interest) # list-wise deletion of missings
```

Data Wrangling Pipeline (III/III)

```
ess10 <- ess10 %>%
  mutate(education_years = na_if(education_years, 114), # set 114 to missing
         pol_interest = (pol_interest * -1) + 5, # invert scale
         life_satisfaction = life_satisfaction + 1 # change scale to [1, 11]
         ) %>%
  drop_na(trust_politicians, gender, education_years,
          life_satisfaction, pol_interest)
```

References

Parts of this course are inspired by the following resources:

- Wickham, Hadley and Garrett Grolemund, 2017. *R for Data Science - Import, Tidy, Transform, Visualize, and Model Data*. O'Reilly.
- Bahnsen, Oke and Guido Ropers, 2022. *Introduction to R for Quantitative Social Science*. Course held as part of the GESIS Workshop Series.
- Breuer, Johannes and Stefan Jünger, 2021. *Introduction to R for Data Analysis*. Course held as part of the GESIS Summer School in Survey Methodology.
- Teaching material developed by Verena Kunz, David Weyrauch, Oliver Rittmann and Viktoriia Semenova.