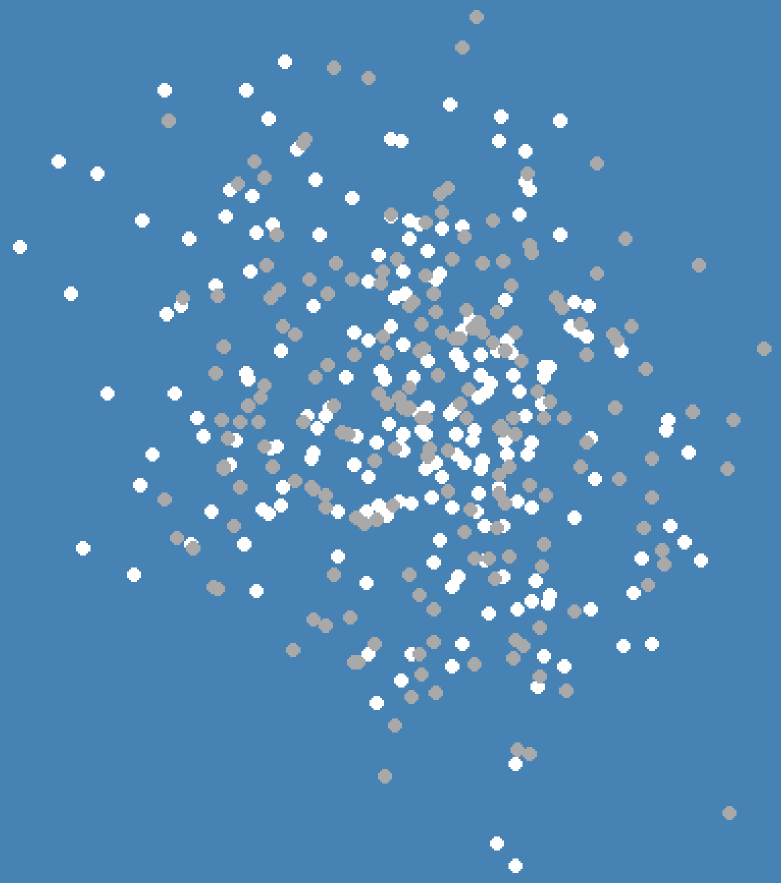
Introduction to R

2.6 Transforming Variables

summarize(), group_by()

Lion Behrens, M.Sc.

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



Data Import

```
library(haven)
ess10 <- haven::read_dta("./dat/ESS10.dta")
dim(ess10) # check dimensionality of data frame
```

```
## [1] 18060   513
```

```
print(ess10[1:10, 1:10])
```

```
## # A tibble: 10 × 10
##    name       essro…¹ edition prodd…²  idno cntry dweight pweight nwspol netus…³
##    <chr>        <dbl> <chr>   <chr>   <dbl> <chr>   <dbl>   <dbl> <dbl+> <dbl+l>
##  1 ESS10e01_2      10 1.2     28.06.… 10002 BG      1.03    0.218  80    1 [Nev…
##  2 ESS10e01_2      10 1.2     28.06.… 10006 BG      0.879   0.218  63    5 [Eve…
##  3 ESS10e01_2      10 1.2     28.06.… 10009 BG      1.01    0.218 390    5 [Eve…
##  4 ESS10e01_2      10 1.2     28.06.… 10024 BG      0.955   0.218  60    5 [Eve…
##  5 ESS10e01_2      10 1.2     28.06.… 10027 BG      0.841   0.218 120    5 [Eve…
##  6 ESS10e01_2      10 1.2     28.06.… 10048 BG      0.946   0.218  60    5 [Eve…
##  7 ESS10e01_2      10 1.2     28.06.… 10053 BG      1.01    0.218  30    5 [Eve…
##  8 ESS10e01_2      10 1.2     28.06.… 10055 BG      1.03    0.218  70    5 [Eve…
##  9 ESS10e01_2      10 1.2     28.06.… 10059 BG      0.991   0.218  60    1 [Nev…
## 10 ESS10e01_2      10 1.2     28.06.… 10061 BG      1.05    0.218  60    1 [Nev…
## # … with abbreviated variable names ¹essround, ²proddate, ³netusoft
```

Using summarize() for summary statistics

dplyr::summarize() vs. dplyr::mutate()

Other than mutate() which…

- generates new variables as transformations of existing variables
- keeps the data structure untouched

… summarize() changes the structure of your data frame.

Computations using summarize()…

- collapse rows to summary statistics
- automatically remove all variables that are irrelevant for the computations

dplyr::mutate()

What are the dimensions of our data frame?

```
dim(ess10)
```

```
## [1] 18060   513
```

Let's build an additive index for trust.

```
ess10 <- ess10 %>%
  mutate(trust_index = trstprl + trstlgl + trstplc + trstplt + trstprt + trstep + trstun)
```

```
table(ess10$trust_index)
```

```
## 
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19 
## 370  51  89  94 114 149 129 155 159 131 205 189 174 226 238 256 212 255 255 272 
##  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 
## 317 294 299 299 276 332 328 323 323 376 388 353 359 377 375 511 417 394 382 379 
##  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59 
## 350 371 333 311 332 307 298 257 253 307 262 219 196 183 195 217 192 132 123  77 
##  60  61  62  63  64  65  66  67  68  69  70 
##  86  48  36  49  27  12  23  13  19   5  51 
```

dplyr::mutate()

What are the dimensions of our data frame?

```r
dim(ess10)
```

```
## [1] 18060   513
```

Let's build an additive index for trust.

```r
ess10 <- ess10 %>%
  mutate(trust_index = trstprl + trstlgl + trstplc + trstplt + trstprt + trstep + trstun)
```

Did the dimensions change?

```r
dim(ess10)
```

```
## [1] 18060   514
```

dplyr::summarize()

Let's try out summarize().

```
new_df <- ess10 %>%
  summarize(tindex_mean = mean(trust_index))
```

```
print(new_df)
```

```
## # A tibble: 1 × 1
##   tindex_mean
##         <dbl>
## 1          NA
```

dplyr::summarize()

Let's try out summarize().

```
new_df <- ess10 %>%
  summarize(tindex_mean = mean(trust_index, na.rm = T))
```

```
print(new_df)
```

```
## # A tibble: 1 × 1
##   tindex_mean
##         <dbl>
## 1        31.8
```

```
dim(new_df)
```

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

dplyr::summarize()

Let's calculate some more summary statistics.

```
new_df <- ess10 %>%
  summarize(tindex_mean = mean(trust_index, na.rm = T),
            tindex_median = median(trust_index, na.rm = T),
            tindex_min = min(trust_index, na.rm = T),
            tindex_max = max(trust_index, na.rm = T),
            tindex_sd = sd(trust_index, na.rm = T)
            )
```

```
print(new_df)
```

```
## # A tibble: 1 × 5
##   tindex_mean tindex_median tindex_min tindex_max tindex_sd
##         <dbl>         <dbl>      <dbl>      <dbl>     <dbl>
## 1        31.8            33          0         70      15.4
```

```
dim(new_df)
```

```
## [1] 1 5
```

Combine dplyr::summarize() with dplyr::group_by()

summarize() can be combined very conveniently with group_by().

Let's calculate summary statistics for different groups!

```
new_df <- ess10 %>%
  summarize(tindex_mean = mean(trust_index, na.rm = T),
            tindex_median = median(trust_index, na.rm = T),
            tindex_min = min(trust_index, na.rm = T),
            tindex_max = max(trust_index, na.rm = T),
            tindex_sd = sd(trust_index, na.rm = T)
            )
```

```
table(ess10$vote)
```

```
## 
##     1     2     3 
## 12037  4684  1155
```

```
library(sjlabelled)
get_labels(ess10$vote)
```

```
## [1] "Yes"                  "No"                   "Not eligible to vote"
## [4] "Refusal"              "Don't know"           "No answer"
```

Combine dplyr::summarize() with dplyr::group_by()

summarize() can be combined very conveniently with group_by().

Let's calculate summary statistics for different groups!

```
new_df <- ess10 %>%
  group_by(vote) %>%
  summarize(tindex_mean = mean(trust_index, na.rm = T),
            tindex_median = median(trust_index, na.rm = T),
            tindex_min = min(trust_index, na.rm = T),
            tindex_max = max(trust_index, na.rm = T),
            tindex_sd = sd(trust_index, na.rm = T)
            )
```

Combine dplyr::summarize() with dplyr::group_by()

summarize() can be combined very conveniently with group_by().

Let's calculate summary statistics for different groups!

```
new_df <- ess10 %>%
  group_by(vote) %>%
  summarize(tindex_mean = mean(trust_index, na.rm = T),
            tindex_median = median(trust_index, na.rm = T),
            tindex_min = min(trust_index, na.rm = T),
            tindex_max = max(trust_index, na.rm = T),
            tindex_sd = sd(trust_index, na.rm = T)
            )
```

Combine dplyr::summarize() with dplyr::group_by()

summarize() can be combined very conveniently with group_by().

Let's calculate summary statistics for different groups!

```
new_df <- ess10 %>%
  group_by(vote) %>%
  summarize(tindex_mean = mean(trust_index, na.rm = T),
            tindex_median = median(trust_index, na.rm = T),
            tindex_min = min(trust_index, na.rm = T),
            tindex_max = max(trust_index, na.rm = T),
            tindex_sd = sd(trust_index, na.rm = T)
            )
```

```
print(new_df)
```

```
## # A tibble: 4 × 6
##   vote                         tindex_mean tindex_median tinde…¹ tinde…² tinde…³
##   <dbl+lbl>                          <dbl>         <dbl>   <dbl>   <dbl>   <dbl>
## 1      1 [Yes]                        33.2            34       0      70    15.2
## 2      2 [No]                         27.1            27       0      70    15.0
## 3      3 [Not eligible to vote]       36.3            38       0      70    15.0
## 4 NA(a)                               28.0            29       0      60    13.2
## # … with abbreviated variable names ¹tindex_min, ²tindex_max, ³tindex_sd
```

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.