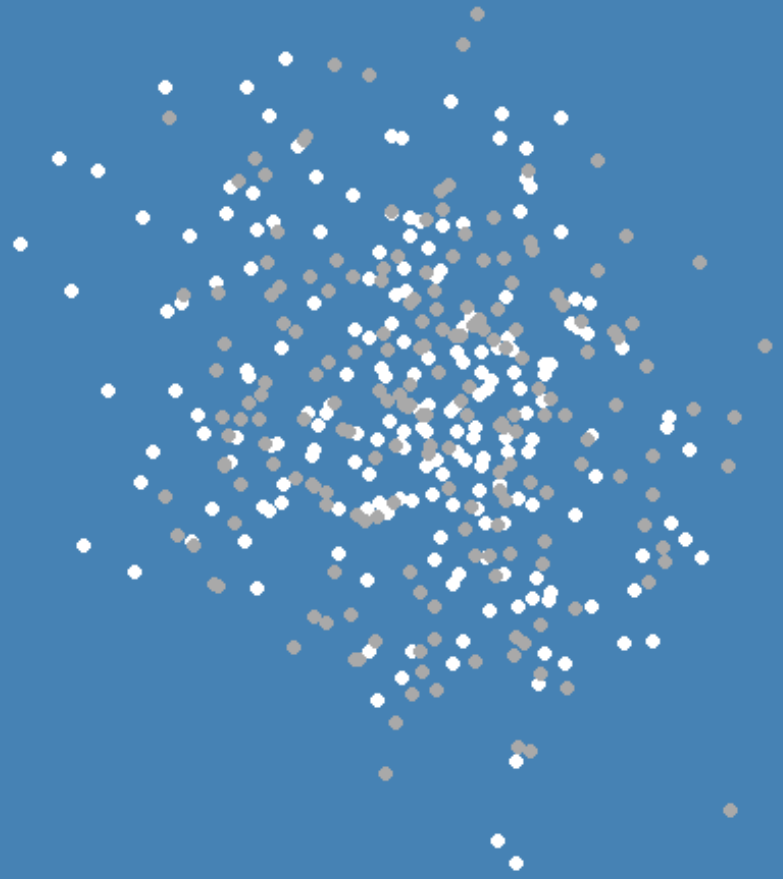
Introduction to R

2.4 Subsetting Rows and Variables

select(), filter()

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
```

Data Import

```
library(haven)
ess10 <- haven::read_dta("./dat/ESS10.dta")
colnames(ess10)[1:50]
```

```
##  [1] "name"     "essround" "edition"  "proddate" "idno"     "cntry"
##  [7] "dweight"  "pweight"  "nwspol"   "netusoft" "netustm"  "ppltrst"
## [13] "pplfair"  "pplhlp"   "polintr"  "psppsgva" "actrolga" "psppipla"
## [19] "cptppola" "trstprl"  "trstlgl"  "trstplc"  "trstplt"  "trstprt"
## [25] "trstep"   "trstun"   "trstsci"  "vote"     "prtvtebg" "prtvtbhr"
## [31] "prtvtecz" "prtvthee" "prtvtefi" "prtvtefr" "prtvtghu" "prtvclt1"
## [37] "prtvclt2" "prtvclt3" "prtvtfsi" "prtvtesk" "contplt"  "donprty"
## [43] "badge"    "sgnptit"  "pbldmna"  "bctprd"   "pstplonl" "volunfp"
## [49] "clsprty"  "prtclebg"
```

Data Import

```
print(ess10[1:15, 1:10])
```

```
## # A tibble: 15 × 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…
## 11 ESS10e01_2       10 1.2     28.06.… 10064 BG      1.00    0.218 300    5 [Eve…
## 12 ESS10e01_2       10 1.2     28.06.… 10068 BG      1.03    0.218   0    1 [Nev…
## 13 ESS10e01_2       10 1.2     28.06.… 10071 BG      0.931   0.218  30    5 [Eve…
## 14 ESS10e01_2       10 1.2     28.06.… 10077 BG      0.991   0.218  30    1 [Nev…
## 15 ESS10e01_2       10 1.2     28.06.… 10078 BG      0.990   0.218  60    1 [Nev…
## # … with abbreviated variable names ¹essround, ²proddate, ³netusoft
```

Reducing your dataset to hand-picked variables

Selecting Variables

Let's say we only want to work with a reduced version of the data set - with those variables that are relevant for our statistical analysis.

- There are some straightforward ways using base R
- The respective function of the tidyverse is select()

base R Option 1

```
# subset country and trust variables
ess10 <- ess10[, c("cntry", "trstprl", "trstlgl", "trstplc", "trstplt", "trstprt",
                   "trstep", "trstun", "trstsci")]
```

Selecting Variables

Let's say we only want to work with a reduced version of the data set - with those variables that are relevant for our statistical analysis.

- There are some straightforward ways using base R
- The respective function of the tidyverse is select()

base R Option 1

```
# subset country and trust variables
ess10 <- ess10[, c("cntry", "trstprl", "trstlgl", "trstplc", "trstplt", "trstprt",
                   "trstep", "trstun", "trstsci")]
ess10 <- ess10[, c(6, 20:27)] # equivalent
```

Selecting Variables

Let's say we only want to work with a reduced version of the data set - with those variables that are relevant for our statistical analysis.

- There are some straightforward ways using base R
- The respective function of the tidyverse is select()

base R Option 1

```
# subset country and trust variables
ess10 <- ess10[, c("cntry", "trstprl", "trstlgl", "trstplc", "trstplt", "trstprt",
                   "trstep", "trstun", "trstsci")]
ess10 <- ess10[, c(6, 20:27)] # equivalent
ess10 <- ess10[, c(6, which(substr(colnames(ess10), 1, 4) == "trst"))] # equivalent
```

base R Option 2

```
# subset country and trust variables
ess10 <- subset(x = ess10,
                subset = TRUE,
                select = c(cntry, trstprl, trstlgl, trstplc, trstplt, trstprt,
                           trstep, trstun, trstsci))
```

Selecting Variables

In the tidyverse, we can select individual variables (columns) using the verb select().

```
# subset country and trust variables
ess10 <- ess10 %>%
  select(cntry, trstprl, trstlgl, trstplc, trstplt, trstprt, trstep, trstun, trstsci)
```

Selecting Variables

In the tidyverse, we can select individual variables (columns) using the verb select().

```
# subset country and trust variables
ess10 <- ess10 %>%
  select(cntry, trstprl, trstlgl, trstplc, trstplt, trstprt, trstep, trstun, trstsci)
```

```
# inspect first rows
head(ess10)
```

```
## # A tibble: 6 × 9
##   cntry     trstprl     trstlgl trstplc trstplt trstprt trstep  trstun  trstsci
##   <chr+lbl> <dbl+lbl>   <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+lb>
## 1 BG        3 [3]       2 [2]   3 [3]   3 [3]   3 [3]   4 [4]   4 [4]    6 [6]
## 2 BG        5 [5]       8 [8]   9 [9]   6 [6]   7 [7]   8 [8]   8 [8]   10 [Com…
## 3 BG        3 [3]       3 [3]   3 [3]   3 [3]   2 [2]   6 [6]   5 [5]    6 [6]
## 4 BG        2 [2]       2 [2]   3 [3]   0 [No … 0 [No … 3 [3]   3 [3]    3 [3]
## 5 BG        0 [No trus… 0 [No … 0 [No … 0 [No … 0 [No … 0 [No … 0 [No …  3 [3]
## 6 BG        0 [No trus… 0 [No … 0 [No … 0 [No … 0 [No … 5 [5]   3 [3]    5 [5]
```

Selecting Variables

The verb select() also works well if you specify a numeric range of consecutive columns.

```
# subset country and trust variables
ess10 <- ess10 %>%
  select(6, 20:27)
```

```
# inspect first rows
head(ess10)
```

```
## # A tibble: 6 × 9
##   cntry     trstprl     trstlgl trstplc trstplt trstprt trstep  trstun  trstsci
##   <chr+lbl> <dbl+lbl>   <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+l> <dbl+lb>
## 1 BG        3 [3]       2 [2]   3 [3]   3 [3]   3 [3]   4 [4]   4 [4]    6 [6]
## 2 BG        5 [5]       8 [8]   9 [9]   6 [6]   7 [7]   8 [8]   8 [8]   10 [Com…
## 3 BG        3 [3]       3 [3]   3 [3]   3 [3]   2 [2]   6 [6]   5 [5]    6 [6]
## 4 BG        2 [2]       2 [2]   3 [3]   0 [No … 0 [No … 3 [3]   3 [3]    3 [3]
## 5 BG        0 [No trus… 0 [No … 0 [No … 0 [No … 0 [No … 0 [No … 0 [No …  3 [3]
## 6 BG        0 [No trus… 0 [No … 0 [No … 0 [No … 0 [No … 5 [5]   3 [3]    5 [5]
```

Selecting and Renaming in One Step

Selecting and renaming variables can be conveniently performed in one step.

```
# subset country and trust variables
ess10 <- ess10 %>%
  select(country = cntry,
         trust_parliament = trstprl,
         trust_legalSys = trstlgl,
         trust_police = trstplc,
         trust_politicians = trstplt,
         trust_parties = trstprt,
         trust_EP = trstep,
         trust_UN = trstun,
         trust_scientists = trstsci)
```

Excluding Variables

If we want to use all variables apart from a small number of columns, it might make more sense to specifically exclude those that won't be used.

Let's say we want to exclude the variables dweight and pweight.

Here are a variety of options for you:

```
# base R
ess10 <- subset(x = ess10,
                subset = TRUE,
                select = - c(dweight, pweight))

# tidyverse
ess10 <- ess10 %>%
  select(-c(dweight, pweight))

# tidyverse, helper functions
ess10 <- ess10 %>%
  select(-c(ends_with("weight")))
```

Note: Other helper functions that are available in the tidyverse are

- starts_with("xyz")
- contains("xyz")

Reducing your dataset to specific rows

Filtering Rows

If we want to subset our dataset to a particular set of observations (rows).

- dplyr holds the verb filter() ready for us
- In base R, this would be called selection with conditions

Comparison operators

Operator	Definition
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	equal to
!=	not equal to

Logical Operators

Operator	Definition
&	and
|	or
!	not
xor	either or, not both
%in%	included in

Filtering Rows in base R

Let's say we want to subset our ESS dataset to respondents who live in Hungary.

Option 1: Selection with conditions

```
ess10 <- ess10[ess10$cntry == "HU",]
dim(ess10)
```

```
## [1] 1849  513
```

Option 2: subset() function

```
ess10 <- subset(ess10,
                cntry == "HU")
dim(ess10)
```

```
## [1] 1849  513
```

Filtering Rows in base R

Let's say we want to subset our ESS dataset to

- respondents who live in Hungary
- who have not voted in the last national elections

First, let's inspect the variable vote:

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

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

We are interested in Category 2 "No".

Filtering Rows in base R

Let's say we want to subset our ESS dataset to

- respondents who live in Hungary
- who have not voted in the last national elections

Option 1: Selection with conditions

```
ess10 <- ess10[which(ess10$cntry == "HU" & ess10$vote == 2),]
dim(ess10)
```

```
## [1] 422 513
```

Option 2: subset() function

```
ess10 <- subset(ess10,
                cntry == "HU" & vote == 2)
dim(ess10)
```

```
## [1] 422 513
```

The Data Wrangling Pipeline (I/III)

```
library(tidyverse)
ess10 <- haven::read_dta("./dat/ESS10.dta")
```

The 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
         )
```

The 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
         ) %>%
  filter(country == "FR") # subset cases (only include France)
```

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.