FROM RAW DATA TO CLEAN DATA IN R

Essential Data Science Operations for Importing, Inspecting, Transforming, and Visualizing Datasets

◈ GOLDEN RULE: Keep raw data unchanged; save cleaning decisions in a script so the analysis is reproducible.

① SETUP | ② IMPORT & CHECK | ③ SUBSET & STRUCTURE | ④ CLEAN & TRANSFORM | ⑤ AUDIT & PLOT

1 SETUP

INSTALL PACKAGES

R is your car's engine; RStudio (Posit) is the interactive dashboard.
Installing (Once per PC): Downloads packages from CRAN into your hard drive library.
Loading (Every Session): Places tools on your active workbench script.

```
install.packages(c("tidyverse","readxl","janitor","lubridate",
"stringr", "skimr"))
```

Alternative way: Menu: Tools > Install Packages > type package name > Install

LOAD

Load packages at the start of each R session so their functions are available.

```
library(tidyverse)
library(readxl)
library(janitor)
library(lubridate)
library(stringr)
library(skimr)
```

→ Work with data + plots
→ Read Excel files
→ Clean column names
→ Handle dates
→ Clean text
→ Summarise data quickly

2 IMPORT & CHECK

IMPORT

Importing means reading an external file, such as a CSV, Excel or text file, into R and storing it as a data frame for analysis

```
df <- read_csv("file.csv")

df <-read_excel("file.xlsx")

df <- read_delim("file.txt",
  delim = "\t")
```

→ Read CSV table

→ Read Excel workbook

→ Read text table (use tab separator)

INSPECT

Inspection tells you what the data looks like before you change it.

```
head(df)
View(df)
glimpse(df)
str(df)
summary(df)
skim(df)
```

→ First rows
→ Spreadsheet view
→ Columns and types
→ Detailed structure
→ Basic summaries
→ Full overview

TYPE CHECKING

Confirm each variable has the right type for analysis.

```
class(df$variable)
as.numeric(x)
as.character(x)
as.factor(x)
as.logical(x)
```

→ Check current type
→ Make numbers
→ Make text
→ Make groups
→ Make TRUE/FALSE

3 SUBSET & STRUCTURE

ASSIGNMENT & SUBSETTING

Subsetting keeps the rows and columns needed for a focused analysis.

```
scores <- c(5, 8, 9)
df <- select(name, age, score)
df <- filter(score >= "Urban")
df <- slice(1:10)
df <- arrange(desc(score))
df <- distinct()
```

→ Create an object
→ Keep columns
→ Keep matching rows
→ Keep rows
→ Sort rows
→ Remove duplicates

REORGANISING COLUMNS

Reorganisation makes variable names and column order clear and consistent.

```
clean_names(df)
rename(df, score = total_score)
mutate(df, pass = score >= 50)
relocate(df, id, date)
select(df, id, date, site, score)
```

→ Standardise names
→ Rename a column
→ Create a column
→ Move columns
→ Set final order

4 CLEAN & TRANSFORM

MISSING DATA

Missing-data checks avoid biased summaries and model errors.

```
is.na(df$score)
df <-
fill(df$score)
sum(is.na(score))
replace_na(list
(score = 0))
```

→ Find missing values
→ Fill down rows
→ Count missing values
→ Fill values

STRINGS

String cleaning fixes spacing, case and inconsistent labels.

```
str_trim(site)
str_to_title(site)
str_to_lower(site)
```

→ Remove spaces
→ Convert to title case
→ Lower case

DATES

Date cleaning turns text dates into usable time variables.

```
ymd(date)
mdy(date)
dmy(date)
year(date)
month(date)
wday(date)
```

→ Year-month-day
→ Month-day-year
→ Day-month-year
→ Extract year
→ Extract month
→ Weekday

PIPE WORKFLOW

A pipe keeps cleaning steps readable from top to bottom.

```
clean_df <- raw_df |>
  mutate(score =
as.numeric(score)) |>
  filter(!is.na(score))
|>select(id, date,
score, site)
```

→ Start clean object
→ Fix numbers
→ Remove missing
→ Pick data
→ Make group
→ Make number

5 AUDIT & PLOT

AUDIT DATA

Auditing catches problems before analysis

```
summary(clean_df)
sum(duplicated())
count(df, site)
count(df, group %in%
c("A","B"))
summarise(mean(score),
  na.rm = TRUE, n =
n())
```

→ Basic summary
→ Count duplicates
→ Rows per group
→ Group data
→ Mean + count
→ Ignore NAs + count

VISUALIZATION

Plots show patterns, outliers and relationships.

```
ggplot(clean_df, aes(age,
score)) +
  geom_point() +
  geom_smooth(method =
"loess") +
  labs(title = "Score vs
Age",
       x = "Age", y =
"Score") +
  theme_minimal()
```

→ Choose data/aes
→ Colour by group
→ Draw points
→ Add labels
→ Clean layout

PLOT GUIDE

Choose plots by variable type

```
geom_bar()
geom_boxplot()
geom_point()
geom_line()
facets
coord.
```

MISTAKES

Small errors can change results.

```
=
==
na.rm = TRUE
library(...)
set.seed(123)
raw_df
```

→ Names arguments
→ Tests equality
→ Ignore missing
→ Load tools
→ Fix dates
→ Keep original

MINI PIPELINE

```
raw_df <- read_csv("survey_data.csv") |>
clean_df <- raw_df |> clean_names() |> mutate(date = dmy(date), score = as.numeric(score)) |>
filter(!is.na(score)) |>
distinct()
```

→ Read raw data
→ Clean names, fix types, dates, remove missing
→ And duplicates

☼ GENERAL TIPS: Name files clearly • Check columns before cleaning • Count missing values • Remove duplicates carefully • Check strange values • Save your results

Spolufinancováno Evropskou unií
Ministerstvo školství, mládeže a tělovýchovy
Open Science
MENDELU