university of groningen

bcn

Best Practices in Writing Reproducible Code in R (Part 1)

Lan Zhou
Cognitive Neuroscience Center
2025/06/03
l.zhou01@umcg.nl

umcg

Agenda today

Part 1 (Lan)

- Introduction to RStudio (10 min)
- Writing Maintainable R Codes in R Notebook (10 min)
- Data Workflow: Load, Check, Clean, Save (30 min)
- Q&A and Resources (10 min)

Part 2 (Diego)

- Basic statistics (30 min)
    - Visualization (30 min)
- Q&A and Resources (10 min)

NOTE: You can always use different packages and different functions to complete the same tasks in R.

We just share our own experiences and preferred ways. It does not mean our codes are the only correct or best approach.

What is R, RStudio and Posit?

R: Programming language for statistics & data

Rstudio: An Integrated Development Environment (IDE) for R (and Python)

Posit: a company that created and maintains Rstudio, support for other languages (like Python) and tools beyond R, such as:

- Shiny (interactive web apps)
- Quarto (scientific publishing)

RStudio Interface Tour

- Script Editor: R script, R notebook
- Console
- Environment / History
- Files / Plots / Packages / Help / Viewer

First Script

- Basic syntax:
  - x = 1
  - y = 2
  - z = x + y
  - print(z)
- Running code (Ctrl + Enter)
- Saving your script

How to look for help?

- ?function_name
- help(function_name)
- example(function_name)

Just google it or ask ChatGPT!

Writing Maintainable R Code in Notebooks

- Why do I use R Notebook not R script?

What is an R Notebook?

- Mix code + narrative (Markdown)
- Reproducible and readable
- Output as HTML/Word/PDF

The advantages of R Notebook

Feature	R Script (Plain text code only)	R Notebook (Mix code and rich text)
Supports Markdown	❌	✅
Inline Output	❌	✅
Reproducibility	⚠ (manual)	✅ (self-contained)
Visual Output (plots/tables)	Separate	Embedded
Export to report format	❌	✅ (HTML, PDF, Word)

Code Style Best Practices

1. Use meaningful variable names

Tips	Example
Use descriptive names	patient_scores not ps
Use lowercase + underscores	mean_height, raw_data
Avoid abbreviations (unless clear)	cognitive_score, not cog
Don’t start with numbers	✅ stage1_data / ❌ 1stage
Avoid reserved words	❌ mean, data, if, df
Be consistent in the writing style	Make a comment the purpose of your codes first, then write the script

Some examples

❌ Common Bad Names

r

Copy Edit

```r
x <- read.csv("data.csv")       # too vague
df1 <- subset(df, cond == 1)    # unclear
temp <- data.frame(...)         # what is "temp"?
```

✅ Good Alternatives

r

Copy Edit

```r
baseline_data <- read.csv("baseline_data.csv")
treatment_group <- subset(baseline_data, group == "treatment")
average_score <- mean(treatment_group$tot_score, na.rm = TRUE)
```

2. Comment your code always

You can use “#” in your scripts, and R ignores anything after a “#”

- Explains **why** you're doing something (not just what)
- Helps **collaborators** understand your logic
- Reminds **you** what you were thinking
- Essential for **reproducibility**

r Copy Edit

```r
# Filter
baseline_data <- data[data$stage %in% c("T0", "T1"), ]
```

r Copy Edit

```r
# Filter only baseline observations (T0 and T1 stages)
baseline_data <- data[data$stage %in% c("T0", "T1"), ]
```

3. Modularize your code

- **Breaking your code into small, well-defined pieces** (functions or scripts) that do **one thing well**.
  - Use different chunks in your R notebook.
  - Each chunk may be for one purpose.

File Structure

- Organize your input and output folders
- Use relative paths， here::here()， and it makes your project portable and reproducible.

R Notebook Structure

- Title & author
- Setup chunk (load packages, set seed)
- Clear sectioning with headers

R workshop: data check and data cleaning tips

Lan Zhou

2025/5/30

1 Abstract

This workshop introduces best practices for writing reproducible and maintainable R code, including using RStudio, organizing code in R Notebooks, and performing data cleaning efficiently.

2 Preparation

2.1 load packages

Part 3: Data Workflow in R

- Read data
- Preview data
  - Data format
  - Data types
  - Data structure
- Check and clean data
  - Missing value
  - Data type
  - Duplicates
  - Data range and outliers
  - Validate the categorical variables
  - Rename and recode variables
  - Transform variables
  - Select variables and filter participants
  - Merge two datasets into one
- Save data and figure
- Create your table 1

Data types and data structures

Data type	Description	Example
numeric	Real numbers (decimals)	1,2,3, 3.5, 90.001
integer	Whole numbers (L)	1L, 2L
character	Text strings	"apple", "R is great"
logical	Boolean values	TRUE, FALSE

Tips: You can use class() function to figure out which data type it is!

Data structure	Description	Example
vector	1D list of elements of the same type	c(1, 2, 3)
factor	Categorical data with levels	factor(c("male", "female"))
matrix	2D array of same type	matrix(1:6, nrow=2)
data frame	Table-like, columns can be different types	data.frame(name, age, score)
list	A container for any type (mix of types)	list(name="Anna", scores=c(90,85))

Reading Data——data frame in R

- readr::read_csv() # read csv file
- openxlsx::read.xlsx() # read xlsx file
- openxl::read_excel() # read xlsx file
- haven::read_sav() # read spss file

How to select a variable in a data frame? You can use the “$”mark, your_data$variable_name

Data format

"Long" format

country	year	metric
x	1960	10
x	1970	13
x	2010	15
y	1960	20
y	1970	23
y	2010	25
z	1960	30
z	1970	33
z	2010	35

"Wide" format

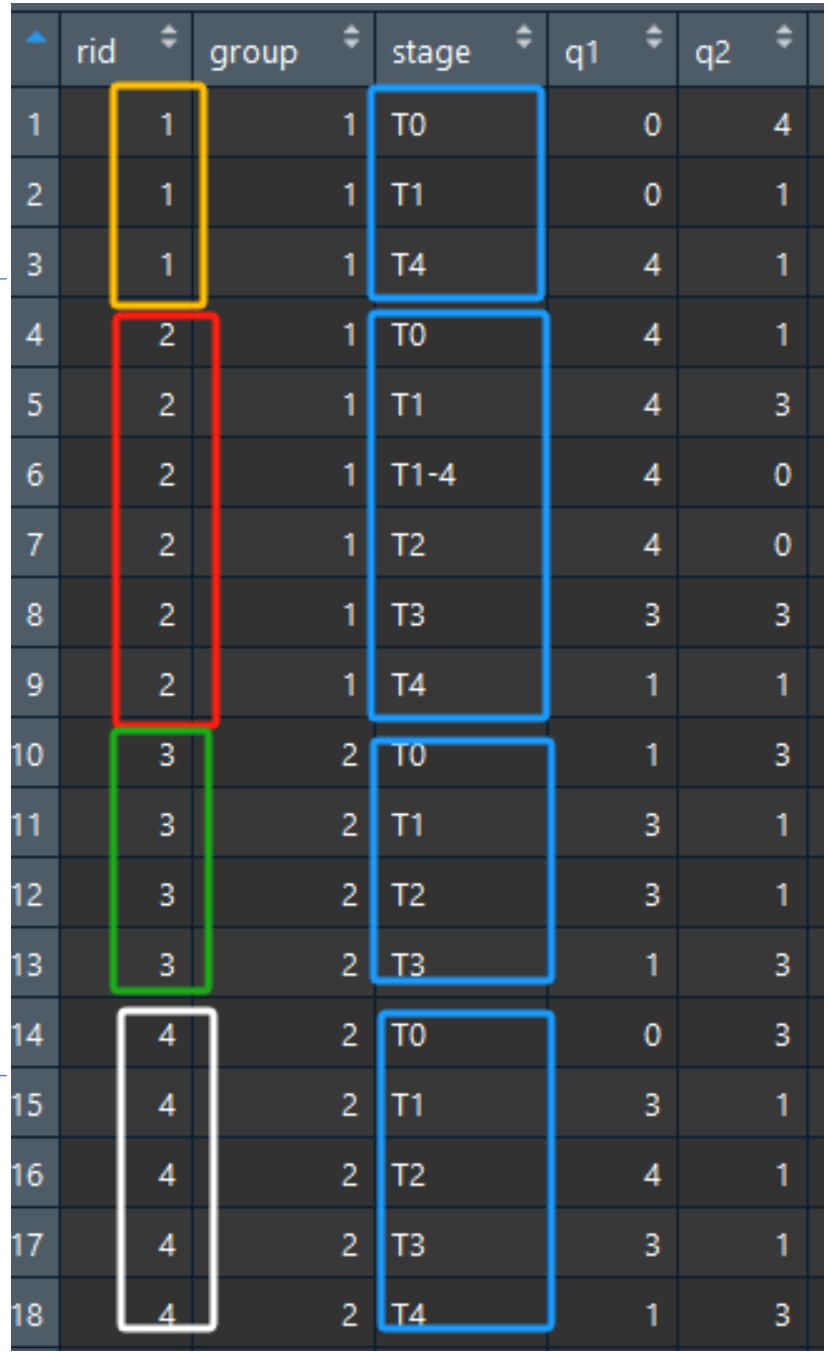
country	yr1960	yr1970	yr2010
x	10	13	15
y	20	23	25
z	30	33	35

If you want to reshape the long format data to wide format data, you can use pivot_wider() function. For tutorials: https://stackoverflow.com/questions/5890584/how-to-reshape-data-from-long-to-wide-format

Data format

Long format

- **Each participant was measured many times;**
- **Each subject ID has more than 1 row.**



	rid	group	stage	q1	q2
1	1	1	T0	0	4
2	1	1	T1	0	1
3	1	1	T4	4	1
4	2	1	T0	4	1
5	2	1	T1	4	3
6	2	1	T1-4	4	0
7	2	1	T2	4	0
8	2	1	T3	3	3
9	2	1	T4	1	1
10	3	2	T0	1	3
11	3	2	T1	3	1
12	3	2	T2	3	1
13	3	2	T3	1	3
14	4	2	T0	0	3
15	4	2	T1	3	1
16	4	2	T2	4	1
17	4	2	T3	3	1
18	4	2	T4	1	3

Checking Data

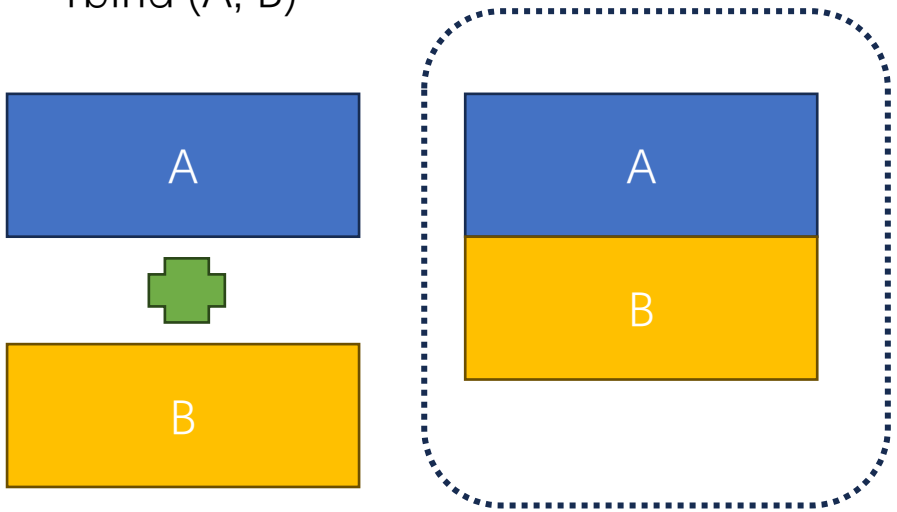
- str(), summary(), head()
- Checking missing data: anyNA(), is.na()

Cleaning Data

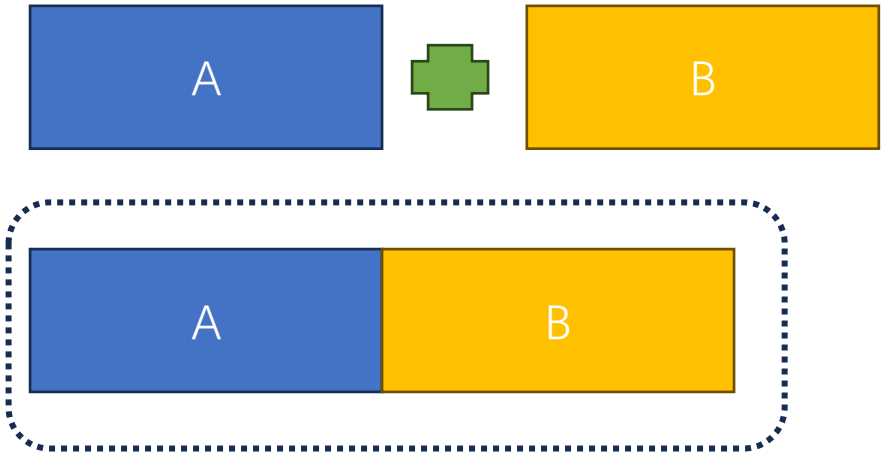
- Use dplyr: select(), filter(), mutate(), case_when()
- Factor handling
- Renaming and recoding

Combine two and more datasets into one file

rbind (A, B)

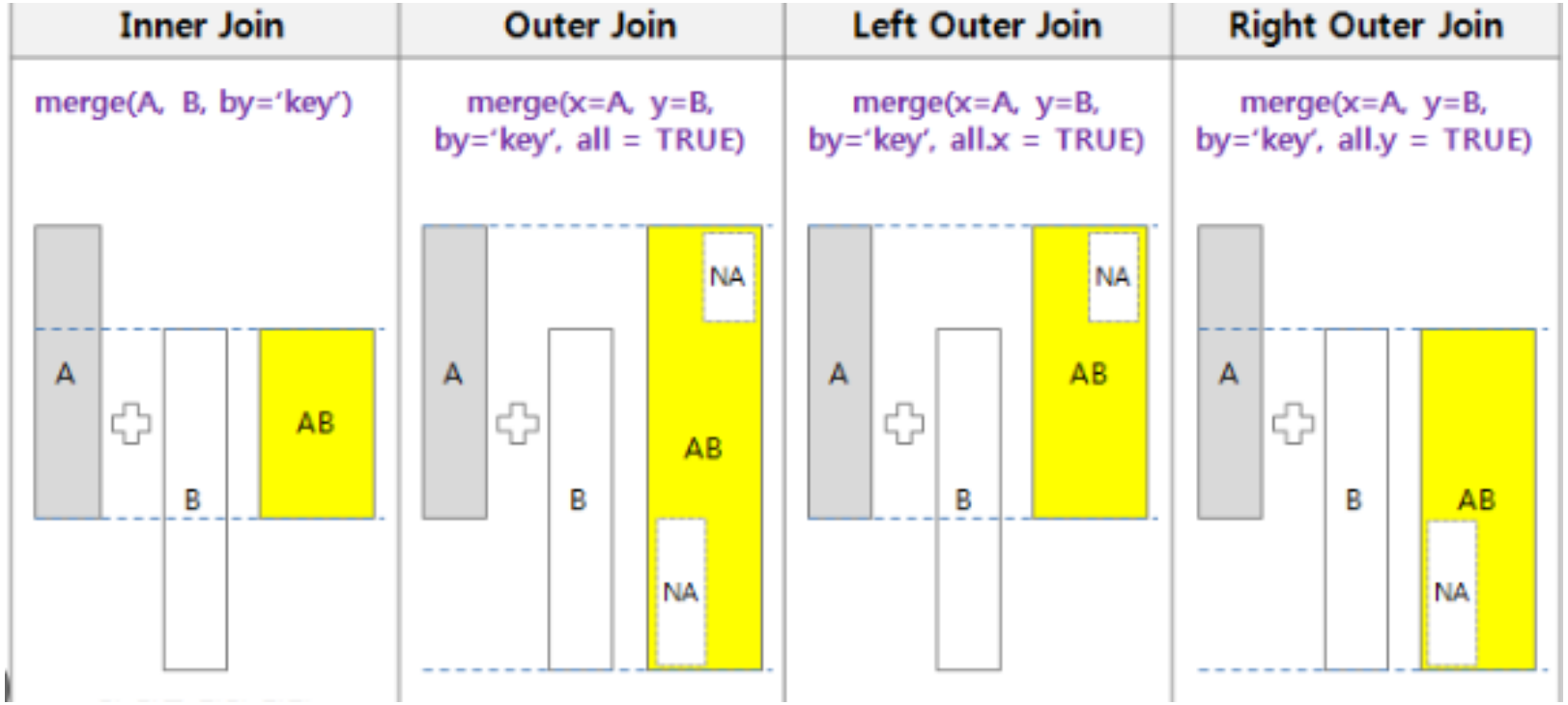


cbind (A,B)



stack one dataset under another

Merge two and more datasets into one file



Saving Data

- write_csv(),
- write_rds()

Create your table 1 for the descriptive table

```
table1(~ factor(sex) + age + factor(ulcer) + thickness | status, data=melanoma2)
```

	Alive (N=134)	Melanoma death (N=57)	Non-melanoma death (N=14)	Overall (N=205)
factor(sex)				
0	91 (67.9%)	28 (49.1%)	7 (50.0%)	126 (61.5%)
1	43 (32.1%)	29 (50.9%)	7 (50.0%)	79 (38.5%)
age				
Mean (SD)	50.0 (15.9)	55.1 (17.9)	65.3 (10.9)	52.5 (16.7)
Median [Min, Max]	52.0 [4.00, 84.0]	56.0 [14.0, 95.0]	65.0 [49.0, 86.0]	54.0 [4.00, 95.0]
factor(ulcer)				
0	92 (68.7%)	16 (28.1%)	7 (50.0%)	115 (56.1%)
1	42 (31.3%)	41 (71.9%)	7 (50.0%)	90 (43.9%)
thickness				
Mean (SD)	2.24 (2.33)	4.31 (3.57)	3.72 (3.63)	2.92 (2.96)
Median [Min, Max]	1.36 [0.100, 12.9]	3.54 [0.320, 17.4]	2.26 [0.160, 12.6]	1.94 [0.100, 17.4]

Tutorial: https://cran.r-project.org/web/packages/table1/vignettes/table1-examples.html

Resources

- R for Data Science book: https://r4ds.had.co.nz/
- Tidyverse documentation: https://www.rdocumentation.org/packages/tidyverse/versions/2.0.0
- dplyr examples: https://dplyr.tidyverse.org/

Thank you!

Q&A

l.zhou01@umcg.nl