

# Reporting with R Markdown

---

**Dr. Kapil Kumar**

Associate Professor, Department of Statistics  
Chaudhary Charan Singh University, Meerut, India  
kapilstats@gmail.com

Five-day Training Programme on Data Analysis and Modeling using R  
(23–27 March 2026)

Department of Statistics, University of Lucknow, with the National Statistics  
Office (CQCD), Zonal Office, Lucknow

**25 March 2026 · Lecture 2 of 2**

# Where we left off

---

```
golf <- read.csv("data/Golf.csv") # 40 drives each
sapply(golf, function(v) shapiro.test(v)$p.value)

## Current      New
## 0.3788 0.3065

var.test(golf$New, golf$Current)$p.value

## [1] 0.4465

tt <- t.test(golf$New, golf$Current, var.equal = TRUE)
round(c(tt$conf.int, tt$p.value), 3) # 95% CI, p

## [1] -6.934 1.384 0.188
```

## Lecture 1 concluded

Normality and equal variances are plausible, so the pooled  $t$ -test applies: no evidence of a difference in mean distance ( $p = 0.19$ ).

**Today** Turn this analysis into a self-updating report.

# Plan of the session

---

1. Why reproducible reports
2. Anatomy of an R Markdown file
3. Chunks, inline code, tables and figures
4. From analysis to report: the golf study
5. Knitting, common errors and good practice

**By the end you can** write the YAML header, Markdown text and code chunks of an `.Rmd` file, control what the reader sees, put tables, figures and live numbers in the text, and knit one source into HTML, PDF and Word.

PART 1

# Why reproducible reports

---

# An analysis is not finished until it is reported

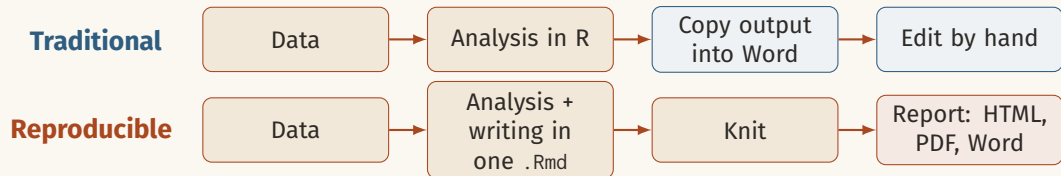
---

- A study ends with a **clear, organised report**: objective, method, results and what they mean
- Usual practice: analyse in R, then *copy* numbers and graphs into Word
- Slow, error-prone, and hard to update when the data change
- R Markdown keeps text, code, output, tables, figures and interpretation in **one file**

**Central idea** The report is *generated* from one source file, never assembled by hand.

**Official statistics** Monthly bulletins are re-run whenever new data arrive: exactly where regeneration beats copy-and-paste.

# Two workflows



**Copy and paste** numbers copied wrongly; new data mean redoing each table by hand; text and output drift apart.

**Knit** code and output stay linked; an update is one click; one source gives several formats.

# Traditional reporting versus R Markdown

---

| Aspect               | Copy into Word | R Markdown |
|----------------------|----------------|------------|
| Analysis and writing | separate       | integrated |
| Updating             | manual         | automatic  |
| Copy-paste errors    | likely         | avoided    |
| Reproducibility      | weak           | strong     |
| Transparency         | limited        | high       |

**Output alone is not a report**  
Neither is code alone, nor interpretation alone. All three must be connected, in one place.

# Reproducible research and the good report

---

A report is **reproducible** if another person can

- take the same data and run the same code,
- obtain the same output,
- see how each conclusion was reached.

This makes results transparent and easy to verify, and keeps the conclusions consistent with the analysis.

**A good statistical report is** clear and logically organised; reproducible; supported by numerical and graphical evidence; interpreted in words; written in formal language.

**Where it helps:** official bulletins, project and consultancy reports, dissertation chapters, practical-class reports.

PART 2

# Anatomy of an **R** Markdown file

---

# Software and packages

---

- **R** for computation, **RStudio** for editing and knitting
- **rmarkdown**: `render()` turns an `.Rmd` into a report
- **knitr**: runs the code chunks; `kable()` for tables
- **tinytex**: a small  $\text{\LaTeX}$  installation, needed only for PDF

Start with HTML output; add PDF once  $\text{\LaTeX}$  is installed.

```
# install once
# install.packages(c("rmarkdown", "knitr",
#                    "tinytex"))
# tinytex::install_tinytex() # PDF only
library(rmarkdown)
library(knitr)
col_main <- "#A9471F" # colours for plots
col_alt  <- "#2F5D80"
```

# Creating a new R Markdown file in RStudio

---

1. Open RStudio; make an RStudio *project* for the work
2. **File** → **New File** → **R Markdown...**
3. Enter a title and author; choose HTML, PDF or Word
4. Save it as `report.Rmd` in the project folder, next to a `data/` folder
5. Click **Knit** (Ctrl+Shift+K)

**Useful keys**   Insert a chunk:  
Ctrl+Alt+I (Mac: Cmd+Option+I).  
Run the current chunk:  
Ctrl+Shift+Enter.

**First knit**   The template RStudio creates already knits. Knit it once before changing anything: then you know the installation works.

# The three parts of an .Rmd file

---

```
---  
title: "Golf ball test"  
output: html_document  
---  
  
# Introduction  
We compare two golf balls.  
  
```${r read-data}  
golf <- read.csv("data/Golf.csv")  
summary(golf)  
```
```

**1. YAML header** between the --- lines: title, author, output.

**2. Markdown text** the narrative: headings, paragraphs, lists, emphasis.

**3. Code chunks** R code in backtick fences; output inserted on knitting.

# Part 1: the YAML header

---

```
---
title: "Report Writing using R"
author: "Dr. Kapil Kumar"
date: "`r Sys.Date()`"
output:
  html_document:
    toc: true
  number_sections: true
  word_document: default
  pdf_document:
    toc: true
---
```

- Metadata: title, author, date
- output: one or more formats, each with options (toc = table of contents)
- ``r Sys.Date()`` inserts the knitting date
- The Knit button uses the *first* format listed

**YAML is strict** Indent with spaces, never tabs; a space after each colon; quote a title that contains a colon.

## Part 2: Markdown text

---

| You type                            | The reader sees               |                 |
|-------------------------------------|-------------------------------|-----------------|
| # Introduction                      | <b>Introduction</b>           | level-1 heading |
| ## Objective                        | <b>Objective</b>              | level-2 heading |
| <b>**bold**</b> and <i>*italic*</i> | <b>bold</b> and <i>italic</i> | emphasis        |
| `Golf.csv`                          | Golf.csv                      | code font       |
| - first item                        | • first item                  | bulleted list   |
| 1. first step                       | 1. first step                 | numbered list   |
| [CRAN](https://cran.r-project.org)  | CRAN                          | link            |
| $\bar{x} = \sum x_i / n$            | $\bar{x} = \sum x_i / n$      | TeX mathematics |

Plain text with a few symbols, readable even before knitting. Leave a **blank line** before every heading, list and new paragraph.

## Part 3: code chunks

---

```
```{r chunk-name, option = value}  
# R code goes here  
```
```

```
```{r golf-summary}  
summary(golf)  
```
```

Code and output appear in the report in the order they occur in the file.

**Name every chunk** Names organise a long report, show *where* knitting failed and name the figure files. Short, unique, no spaces.

**Order matters** Chunks run top to bottom in one R session: create an object before you use it.

PART 3

# Chunks, inline code, tables and figures

---

# Chunk options: what does the reader see?

---

| Option                                        | Effect                                 | Typical use           |
|-----------------------------------------------|----------------------------------------|-----------------------|
| <code>echo = FALSE</code>                     | runs; shows the output, hides the code | polished reports      |
| <code>eval = FALSE</code>                     | shows the code, does not run it        | teaching syntax       |
| <code>include = FALSE</code>                  | runs; shows nothing                    | setup, reading data   |
| <code>results = "hide"</code>                 | shows the code, hides printed output   | long output           |
| <code>message = FALSE, warning = FALSE</code> | no messages or warnings                | package start-up text |
| <code>fig.width, fig.height</code>            | figure size in inches                  | readable plots        |
| <code>fig.cap = "..."</code>                  | caption, numbered figure               | every figure          |
| <code>out.width = "70%"</code>                | displayed width                        | fitting the page      |

---

Options go in the chunk header, separated by commas: ````{r boxplot, echo = FALSE, fig.cap = "Distances"}`

# Four patterns you will use every day

---

```
```{r}
x <- rnorm(100)
summary(x)
```
```

Show and run: code and output appear.

```
```{r, echo = FALSE}
mean(x)
```
```

Hide the code: only the result appears.

```
```{r, eval = FALSE}
y <- rnorm(50)
mean(y)
```
```

Show, do not run: syntax for the reader.

```
```{r, include = FALSE}
z <- x^2
```
```

Run silently: z exists for later chunks.

# The setup chunk: options for the whole report

---

```
```${r setup, include = FALSE}
knitr::opts_chunk$set(
  echo      = TRUE,    # show code
  message = FALSE,    # no messages
  warning  = FALSE,    # no warnings
  fig.width = 6, fig.height = 4)
library(ggplot2)      # packages used later
set.seed(123)         # same random numbers
```
```

- The first chunk sets *defaults* for all chunks
- An option in a chunk header overrides the default for that chunk
- Load every package here, once

**Set the seed** The original handout drew `rnorm()` without `set.seed()`: its numbers changed at every knit. With simulation, a seed is what makes a report reproducible.

# Inline R code: numbers inside sentences

```
d <- mean(golf$New) - mean(golf$Current)
round(c(current = mean(golf$Current),
      new = mean(golf$New), diff = d), 1)

## current      new      diff
##   270.3    267.5    -2.8
```

**Typed by hand**, “the new ball averaged 267.5 yards” is wrong as soon as a data error is corrected.

This deck does the same with `\Sexpr{}` in a knitr `.Rnw` file.

## You write

The new ball averaged  
``r round(mean(golf$New), 1)`` yards,  
against ``r round(mean(golf$Current), 1)``  
for the current ball.

## The reader sees

The new ball averaged 267.5 yards,  
against 270.3 for the current ball.

# Tables with `knitr::kable()`

```
tab <- data.frame(  
  n      = sapply(golf, length),  
  Mean   = sapply(golf, mean),  
  SD     = sapply(golf, sd),  
  Median = sapply(golf, median))
```

`kable()` turns a data frame into a formatted table in HTML, PDF or Word; round with `digits`, and let the caption say what the table shows.

Fixed from the handout: its `kable()` printed all 32 rows of `mtcars` but described “the first six”.

```
kable(tab, digits = 1, booktabs = TRUE,  
      caption = "Driving distance (yards)")
```

Driving distance (yards)

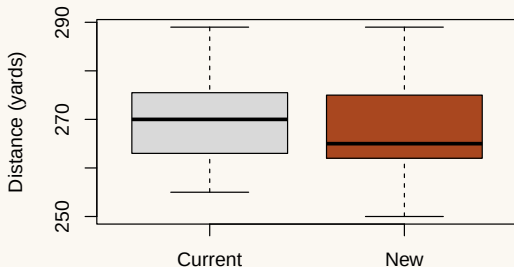
|         | n  | Mean  | SD  | Median |
|---------|----|-------|-----|--------|
| Current | 40 | 270.3 | 8.8 | 270    |
| New     | 40 | 267.5 | 9.9 | 265    |

# Figures: one chunk, one well-labelled graph

```
```{r boxplot, fig.cap = "Distances"}  
boxplot(golf, ylab = "Distance (yards)")  
```
```

**Then say what it shows** Medians of 270 (current) and 265 (new) yards; similar spread (IQR 12.25 and 12.5); the boxes overlap heavily. The picture agrees with the  $t$ -test.

```
par(mar = c(3, 4, 1, 1))  
boxplot(golf, ylab = "Distance (yards)",  
        col = c("grey85", col_main))
```



PART 4

# From analysis to report

---

# Writing up the golf result: weak versus strong

---

**Weak** “ $p = 0.188$ .”

“Not significant, so the two balls are equally good.”

“The boxplot is given below.”

**Strong** On average the new ball travelled 2.8 yards less than the current ball (95% CI for New – Current:  $-6.9$  to  $1.4$  yards; pooled  $t$ -test,  $p = 0.19$ ). The data give no evidence that the new ball travels farther.

- Direction, size in yards and uncertainty, not just a  $p$ -value
- “Not significant” is not “equal”: the interval still allows anything from a loss of 6.9 yards to a gain of 1.4 yards

# The golf report, part 1: header and set-up

---

```
---
title: "Does the New Golf Ball Travel Farther?"
author: "Dr. Kapil Kumar"
date: "25 March 2026"
output:
  pdf_document:
    number_sections: true
    extra_dependencies: ["float"]
fontsize: 11pt
fontfamily: mathpazo
geometry: "a5paper, margin=12mm"
---
```

float with `fig.pos = "H"` keeps each figure where it is written; `fontfamily` picks the typeface.

```
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE,
  fig.width = 4.6, fig.height = 2.4,
  fig.align = "center", fig.pos = "H")
```

# Introduction

**Objective:** to find out whether a new golf
ball travels farther than the current one,
from 40 drives (in yards) with each ball.
```

`echo = FALSE`: the reader sees results, not code.  
File: `example/golf_report.Rmd`, with `Golf.csv` in `example/data/`.

## Part 2: data and methods

---

```
# Data and methods

```{r analysis}
golf <- read.csv("data/Golf.csv")
sw <- sapply(golf, function(v)
  shapiro.test(v)$p.value)
fp <- var.test(golf$New, golf$Current)$p.value
tt <- t.test(golf$New, golf$Current,
  var.equal = TRUE)
d <- mean(golf$New) - mean(golf$Current)
```
```

The whole analysis of Lecture 1 in one chunk, reading the data with a *relative* path.

There are `nrow(golf)` drives per ball and `sum(is.na(golf))` missing values. Shapiro--Wilk tests gave `sp$ = round(sw[1], 2)` (current) and `round(sw[2], 2)` (new); the `F` test for equal variances gave `sp$ = round(fp, 2)`. We therefore used the pooled two-sample `t`-test.

### The reader sees

#### 2 Data and methods

There are 40 drives per ball and 0 missing values. Shapiro--Wilk tests gave  $p = 0.38$  (current) and 0.31 (new); the  $F$  test for equal variances gave  $p = 0.45$ . We therefore used the pooled two-sample  $t$ -test.

## Part 3: the results table

```
# Results
```

```
```{r summary-table}
tab <- data.frame(
  n    = sapply(golf, length),
  Mean = sapply(golf, mean),
  SD   = sapply(golf, sd))
knitr::kable(tab, digits = 1,
  caption = "Driving distance (yards)")
```
```

Row names `Current` and `New` come from the columns of `golf`; `digits = 1` rounds every number in the table.

### The reader sees

Table 1: Driving distance (yards)

|         | n  | Mean  | SD  |
|---------|----|-------|-----|
| Current | 40 | 270.3 | 8.8 |
| New     | 40 | 267.5 | 9.9 |

## Part 4: the results paragraph

```
The new ball went `r round(abs(d), 1)` yards  
`r ifelse(d < 0, "less", "more")` on average  
than the current ball (95% CI for  
New - Current: `r round(tt$conf.int[1], 1)`  
to `r round(tt$conf.int[2], 1)` yards;  
$t$ = `r round(tt$statistic, 2)`,  
$p$ = `r round(tt$p.value, 3)`).
```

**No number typed by hand** Even the word “less” is computed. If the data file is corrected, knit again: the table, the figure and this sentence all update together.

### The reader sees

The new ball went 2.8 yards less on average than the current ball (95% CI for New – Current: -6.9 to 1.4 yards;  $t = -1.33$ ,  $p = 0.188$ ).

## Part 5: the figure and the conclusion

```
```{r boxplot, fig.cap = "Driving distances."}
par(mar = c(2.5, 4, 1, 0.5))
boxplot(golf, ylab = "Distance (yards)",
        col = c("grey85", "#F1E8DA"))
```
```

# Conclusion

The 95% CI includes zero: no evidence that the new ball travels farther. The boxplots show heavily overlapping distances.

The conclusion is the one passage typed without inline R: re-read it whenever the data change.

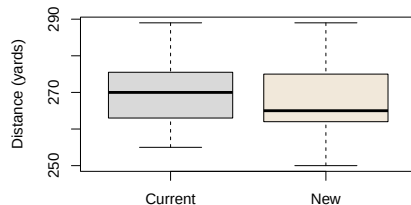


Figure 1: Driving distances.

### 4 Conclusion

The 95% CI includes zero: no evidence that the new ball travels farther. The boxplots show heavily overlapping distances.

# One command, a two-page report

## Does the New Golf Ball Travel Farther?

Dr. Kapil Kumar

25 March 2026

### 1 Introduction

**Objective:** to find out whether a new golf ball travels farther than the current one, from 40 drives (in yards) with each ball.

### 2 Data and methods

There are 40 drives per ball and 0 missing values. Shapiro-Wilk tests gave  $p = 0.38$  (current) and  $0.31$  (new); the  $F$  test for equal variances gave  $p = 0.45$ . We therefore used the pooled two-sample  $t$ -test.

### 3 Results

Table 1: Driving distance (yards)

|         | <i>n</i> | Mean  | SD  |
|---------|----------|-------|-----|
| Current | 40       | 270.3 | 8.8 |
| New     | 40       | 267.5 | 9.9 |

The new ball went 2.8 yards less on average than the current ball (95% CI for New - Current: -6.9 to 1.4 yards;  $t = -1.33$ ,  $p = 0.188$ ).

1

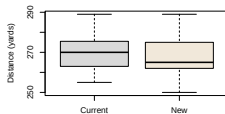


Figure 1: Driving distances.

### 4 Conclusion

The 95% CI includes zero: no evidence that the new ball travels farther. The boxplots show heavily overlapping distances.

2

```
rmarkdown::render(  
  "example/golf_report.Rmd")
```

- Title block from the YAML
- Numbered sections from # headings
- Table 1 from `kable()`; Figure 1 from the boxplot chunk
- Every number in the text from inline R

## A second example: fuel economy of 32 cars (mtcars)

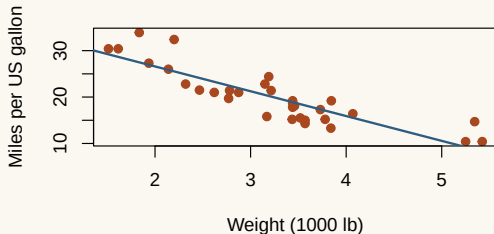
```
model <- lm(mpg ~ wt, data = mtcars)
round(cbind(estimate = coef(model),
            confint(model)), 2)
```

```
##           estimate 2.5 % 97.5 %
## (Intercept)   37.29 33.45 41.12
## wt           -5.34 -6.49 -4.20
```

```
round(summary(model)$r.squared, 3)
```

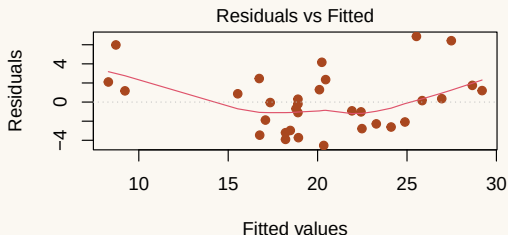
```
## [1] 0.753
```

```
par(mar = c(4, 4, 1, 1))
plot(mpg ~ wt, data = mtcars, pch = 19,
     col = col_main, xlab = "Weight (1000 lb)",
     ylab = "Miles per US gallon")
abline(model, lwd = 2, col = col_alt)
```



# Check the model, then write

```
par(mar = c(4, 4, 2, 1))  
plot(model, which = 1, sub.caption = "",  
      id.n = 0, col = col_main, pch = 19)
```



U-shaped residuals: the line misses a curvature.

```
model2 <- lm(mpg ~ wt + I(wt^2), data = mtcars)  
round(summary(model2)$r.squared, 3)  
  
## [1] 0.819
```

**Written up** Among these 32 cars, each extra 1000 lb is associated with 5.3 fewer miles per gallon on average (95% CI 4.2 to 6.5). The line is a first approximation ( $R^2$  0.75; quadratic 0.82).

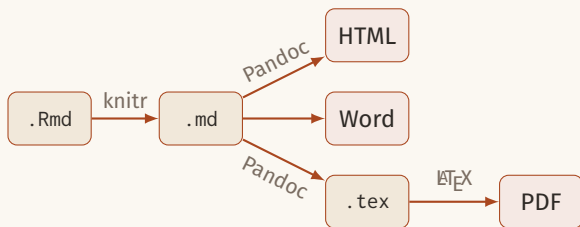
PART 5

# Knitting and good practice

---

# What happens when you click Knit

---



1. A **new, clean** R session runs every chunk, top to bottom
2. Text and results are woven into a Markdown file
3. Pandoc converts it to the format in the YAML (PDF via  $\text{\LaTeX}$ )

**Consequence** The `.Rmd` file, not your workspace, is the complete and authoritative record of the analysis.

# Output formats: one source, three reports

---

```
output: html_document
```

```
output: pdf_document
```

```
output: word_document
```

HTML and Word need nothing extra; PDF needs  $\text{\LaTeX}$  (TinyTeX).

```
# from the console: all formats in the YAML  
rmarkdown::render("report.Rmd", output_format = "all")
```

**Which format?** HTML for screens and e-mail; Word when colleagues will edit; PDF for submission and printing.

**Going further** `params:` in the YAML turns a report into a template, knitted once per district or month.

# Knitting starts from a clean session

---

- An object made only in the console does *not* exist while knitting: “object ‘x’ not found”
- Load every package and read every data file *inside* the .Rmd
- Relative paths: `read.csv("data/Golf.csv")`. The handout used an absolute path (`D:/...`), which fails on every other computer
- Do not change the working directory inside a report

**Self-contained** Whoever receives the folder (.Rmd + data/) must be able to knit it unchanged.

**A trap** `render()` typed in the console uses your current workspace and can hide a missing object; the Knit button (fresh session) is the real test.

# Common errors while knitting, and the fix

---

| Message or symptom                                                                          | Cause                                                             | Fix                                                                          |
|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------|------------------------------------------------------------------------------|
| Scanner error: mapping values are not allowed<br>code printed as text; document stops early | YAML: bad indent, tab or an unquoted colon<br>unmatched backticks | spaces only; quote such titles<br>three backticks open and close every chunk |
| there is no package called ...                                                              | package not installed                                             | install once in the console; keep <code>library()</code> in the file         |
| object 'x' not found                                                                        | made in the console, or used before created                       | create everything in the file, in order                                      |
| LaTeX failed to compile                                                                     | no $\text{\LaTeX}$ for PDF                                        | <code>tinytex::install_tinytex()</code> , or knit to HTML                    |
| cannot open file 'data/Golf.csv'                                                            | wrong path or folder                                              | data next to the report; relative paths                                      |

# Good practice in report writing

---

## Writing

- informative headings
- outputs explained in words
- captions on tables and figures
- conclusions the data support

## Code

- named chunks, run top to bottom
- a seed for any simulation
- no needless messages or warnings
- relative paths; knit to test

**“Code, output, end” is not a report** Write what was done, why, what was obtained and what it means.

# A pattern for writing interpretations

---

1. **Describe** the result, with number and units
2. **Explain** what it means
3. **Connect** it to the question
4. **Conclude** cautiously: uncertainty, limitations

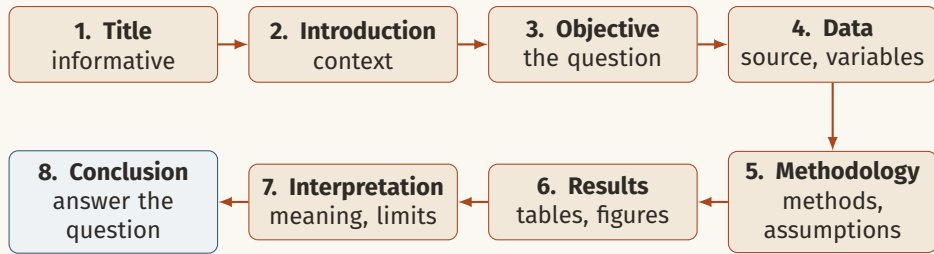
For a model, name the response and the explanatory variable, the direction of the association, and whether the effect matters in practice.

## A good report answers

1. What was studied?
2. How was it analysed?
3. What was found?
4. What does it mean?

# A suggested structure for a statistical report

---



The golf report followed a short version of this skeleton. In an `.Rmd` each box becomes a # heading.

# Exercise

---

Prepare an R Markdown report on `iris` or `airquality` with

1. a title, an introduction and the objective
2. a short description of the data set
3. summary statistics and one `kable()` table
4. at least two captioned plots
5. at least one inline R expression
6. interpretation after each important output, and a conclusion

**Then** Knit to HTML first; then try Word and PDF from the same file.

**`airquality` has gaps** Ozone and Solar.R contain NAs: count them, say how you handled them, and use `na.rm = TRUE` where needed.

# Key take-aways

---

1. **One source file** holds the text, the code, the output and the interpretation; the report is generated, not assembled.
2. An `.Rmd` has three parts: **YAML header**, **Markdown text** and **code chunks**.
3. **Chunk options** decide what the reader sees; **inline R** puts live numbers in sentences.
4. Knitting starts from a **clean session**: load packages and read data inside the file, with relative paths and a seed.
5. **Interpret every important output**: direction, size with units, uncertainty, limitations.
6. Knit early and often; HTML first, then Word and PDF.

# References

---

- Xie, Y., Allaire, J. J., and Golemund, G. (2018). *R Markdown: The Definitive Guide*, CRC Press, Boca Raton.
- Xie, Y., Dervieux, C., and Riederer, E. (2020). *R Markdown Cookbook*, CRC Press, Boca Raton.
- Wickham, H., Çetinkaya-Rundel, M., and Golemund, G. (2023). *R for Data Science*, 2nd Ed., O'Reilly, Sebastopol. Free online: [r4ds.hadley.nz](https://r4ds.hadley.nz)
- R Core Team (2024). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna. <https://www.R-project.org>
- RStudio Desktop (Posit): <https://posit.co/download/rstudio-desktop/>

# Thank you

---

Questions and discussion