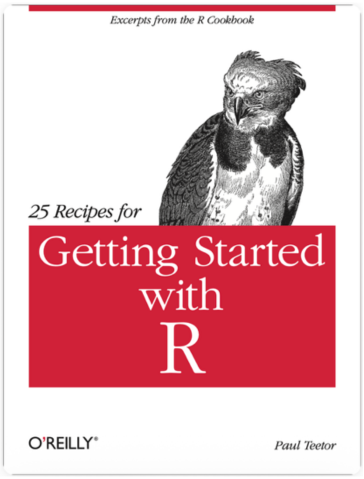
🧠 R Recitation Worksheet
Getting Started with R



📦 1. Installing and Starting R

🔹 1.1 Downloading and Installing R

Goal: Install and run R on your system.

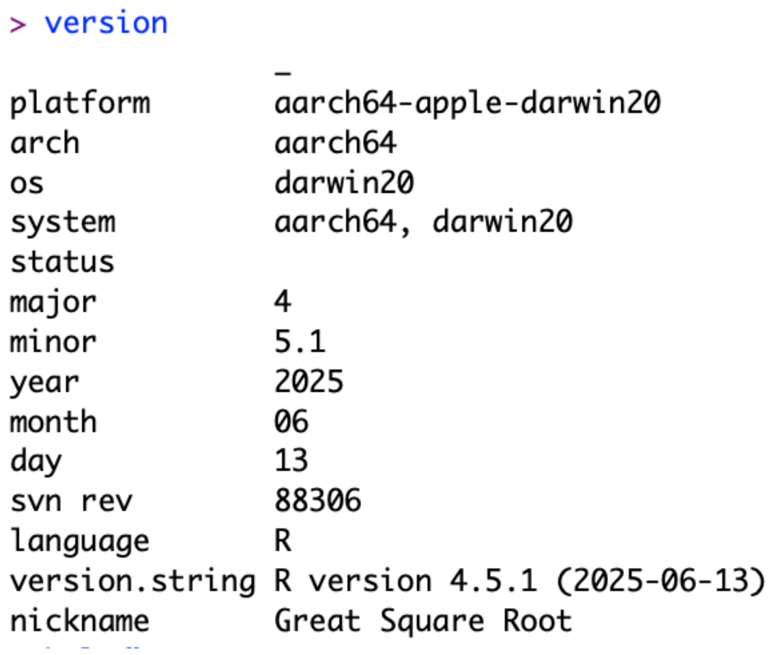
Steps:

- Visit https://cran.r-project.org
- Choose a CRAN mirror near your location.
- Download and install R for your operating system (Windows, macOS, or Linux).

Test your installation:

```
> version
```

✅ You should see your R version printed in the console as below.



```
> version
               _
platform       aarch64-apple-darwin20
arch           aarch64
os             darwin20
system         aarch64, darwin20
status
major          4
minor          5.1
year           2025
month          06
day            13
svn rev        88306
language       R
version.string R version 4.5.1 (2025-06-13)
nickname       Great Square Root
```

🔷 1.2 Getting Help on a Function

Goal: Learn how to use R's built-in help system.

```
help(mean)
?sd
args(mean)
example(mean)
```

💡 help() or ? opens documentation.
💡 args() shows function arguments.
💡 example() runs example code from the help page.

Exercise: Open help for the median function and explore its examples.

🔷 1.3 Viewing R Documentation

Goal: Access the built-in manuals and package references.

```
help.start()
```

This opens a browser window with the full documentation table of contents. Explore the **"Packages"** and **"Search Engine & Keywords"** links.

🔷 1.4 Searching the Web for Help

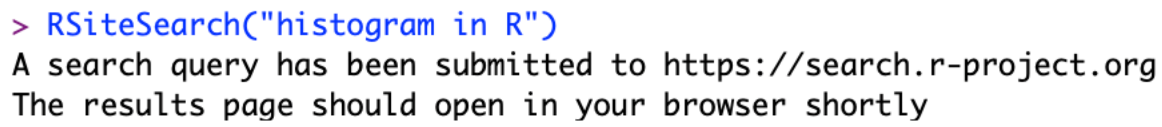
Goal: Find external help when local docs are not enough.

```
> RSiteSearch("linear regression")
```

Useful websites:

- rseek.org — R-specific Google search
- [Stack Overflow]() — Programming Q&A
- [CrossValidated]() — Statistical Q&A

Exercise: Try searching "histogram in R" with RSiteSearch().

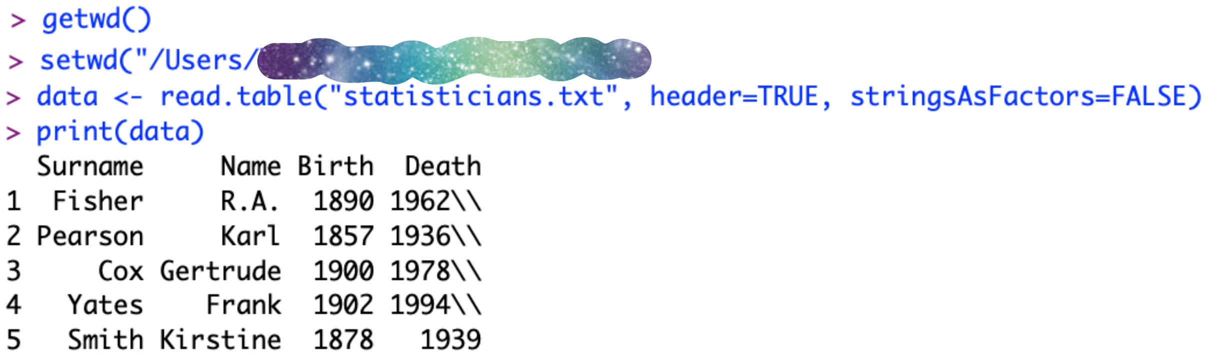


```
> RSiteSearch("histogram in R")
A search query has been submitted to https://search.r-project.org
The results page should open in your browser shortly
```

📊 2. Reading Data and Basic Operations

🔷 2.1 Reading Tabular Data Files

Goal: Read a simple text table into R.



```
> getwd()
> setwd("/Users/[redacted]
> data <- read.table("statisticians.txt", header=TRUE, stringsAsFactors=FALSE)
> print(data)
  Surname     Name Birth  Death
1  Fisher     R.A.  1890 1962\\
2 Pearson     Karl  1857 1936\\
3     Cox Gertrude  1900 1978\\
4   Yates    Frank  1902 1994\\
5   Smith Kirstine  1878   1939
```

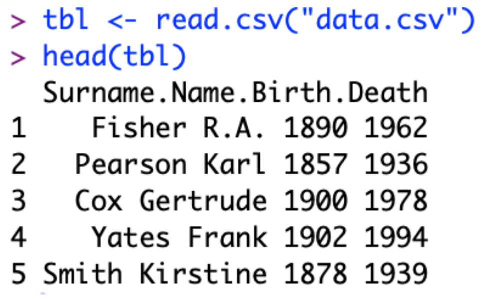
Notes:

- header=TRUE → first line contains column names
- sep=":" or sep="\t" → specify delimiter if not whitespace
- stringsAsFactors=FALSE → prevents automatic conversion to factors

Exercise: Create your own short .txt table and read it using read.table().

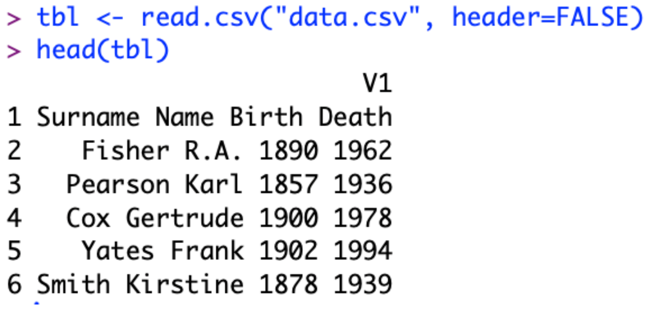
🔷 2.2 Reading from CSV Files

Goal: Load data from a .csv file.



```
> tbl <- read.csv("data.csv")
> head(tbl)
  Surname.Name.Birth.Death
1     Fisher R.A. 1890 1962
2   Pearson Karl 1857 1936
3   Cox Gertrude 1900 1978
4     Yates Frank 1902 1994
5 Smith Kirstine 1878 1939
```

If there's no header row:

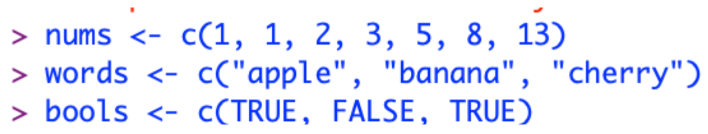


```
> tbl <- read.csv("data.csv", header=FALSE)
> head(tbl)
                        V1
1 Surname Name Birth Death
2     Fisher R.A. 1890 1962
3   Pearson Karl 1857 1936
4   Cox Gertrude 1900 1978
5     Yates Frank 1902 1994
6 Smith Kirstine 1878 1939
```

2.3 Creating a Vector

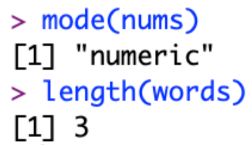
Goal: Create numeric, character, and logical vectors.

The function c() **combines values into a vector or list**



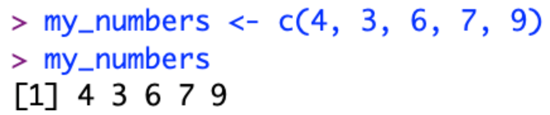
```
> nums <- c(1, 1, 2, 3, 5, 8, 13)
> words <- c("apple", "banana", "cherry")
> bools <- c(TRUE, FALSE, TRUE)
```

Check types and lengths:



```
> mode(nums)
[1] "numeric"
> length(words)
[1] 3
```

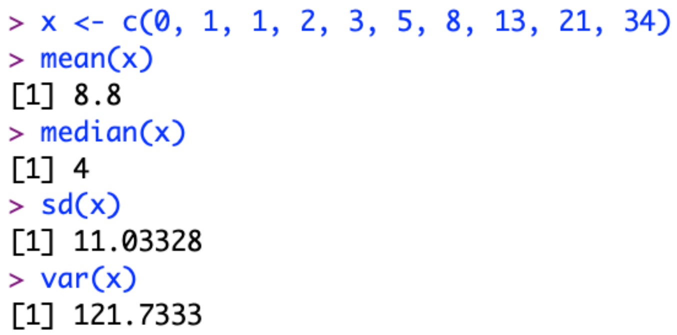
Exercise: Create a numeric vector of 5 numbers called my_numbers.



```
> my_numbers <- c(4, 3, 6, 7, 9)
> my_numbers
[1] 4 3 6 7 9
```

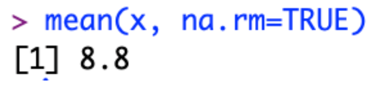
2.4 Computing Basic Statistics

Goal: Calculate descriptive statistics for a vector or data frame.



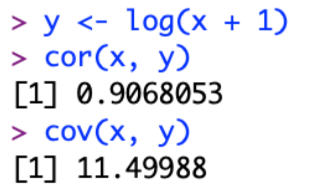
```
> x <- c(0, 1, 1, 2, 3, 5, 8, 13, 21, 34)
> mean(x)
[1] 8.8
> median(x)
[1] 4
> sd(x)
[1] 11.03328
> var(x)
[1] 121.7333
```

Handling missing values:



```
> mean(x, na.rm=TRUE)
[1] 8.8
```

Correlations:

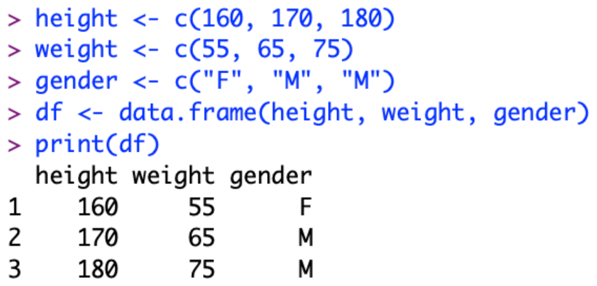


```
> y <- log(x + 1)
> cor(x, y)
[1] 0.9068053
> cov(x, y)
[1] 11.49988
```

Exercise: Create two numeric vectors and compute mean, sd, and cor.

2.5 Initialising a Data Frame

Goal: Combine multiple vectors into a structured data frame.

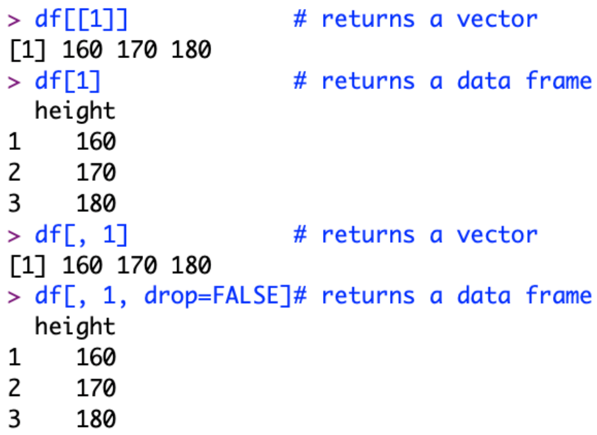


```
> height <- c(160, 170, 180)
> weight <- c(55, 65, 75)
> gender <- c("F", "M", "M")
> df <- data.frame(height, weight, gender)
> print(df)
  height weight gender
1    160     55      F
2    170     65      M
3    180     75      M
```

Exercise: Create a data frame named student with columns: name, age, and grade.

2.6 Selecting Columns by Position

Goal: Learn to select columns from a data frame.

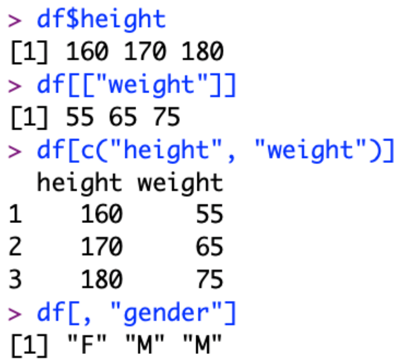


```
> df[[1]]            # returns a vector
[1] 160 170 180
> df[1]              # returns a data frame
  height
1    160
2    170
3    180
> df[, 1]            # returns a vector
[1] 160 170 180
> df[, 1, drop=FALSE]# returns a data frame
  height
1    160
2    170
3    180
```

Exercise: Select columns 2 and 3 from your data frame.

2.7 Selecting Columns by Name

Goal: Access data frame columns by name instead of index.



```
> df$height
[1] 160 170 180
> df[["weight"]]
[1] 55 65 75
> df[c("height", "weight")]
  height weight
1    160     55
2    170     65
3    180     75
> df[, "gender"]
[1] "F" "M" "M"
```

Exercise: Use $ to access and print one column from your data frame.

🎯 Practice Tasks

1. Read a CSV file using read.csv() and compute its column means.
2. Create a numeric vector with some missing values and compute the mean with and without na.rm=TRUE.
3. Make a simple data frame and extract one column using three different methods.