

# **An Introduction to Medical Statistics with R**

**1st Edition**

Konstantinos I. Bougioukas, PhD

June 12, 2022



# Table of contents

<b>Preface</b>	<b>1</b>
License . . . . .	2
<b>1 Examples</b>	<b>3</b>
1.1 intro 2 . . . . .	3
<b>2 Introduction to R and R studio</b>	<b>9</b>
Learning Objectives . . . . .	9
2.1 R and RStudio basics . . . . .	11
<b>3 Activities</b>	<b>13</b>
✎ Activity 1 . . . . .	13
<b>4 Summary</b>	<b>15</b>
<b>References</b>	<b>17</b>

## *Table of contents*

# List of Figures

2.1 Elephant . . . . . 12

## *List of Figures*

# List of Tables





# Preface

This textbook is intended as an introduction to some of the statistical ideas and concepts important to life-science field. It is written for postgraduate students, with most topics also being suitable for undergraduate medical students. The textbook is beneficial to other branches of biological sciences like dental, veterinary, agriculture, etc. which have a lot in common with medical field in the methodology. The textbook is also intended for a wider range of readers who use statistical methods in their research.

While statistics books focus on mathematics, this textbook focuses on using a computer to conduct data analysis. That means using a statistical software package, in this case the [R](#) programming language for statistics and graphics. Our aim is to keep a balance between mathematical rigor and readability as well as learning R and statistics simultaneously.

## *List of Tables*

Most of the examples discussed in this textbook are based on scientific studies whose data are publicly available. For each example, we provide the step-by-step application of R. Readers are encouraged to follow these steps while reading the textbook so that they can learn statistics through hands-on experience.

All sections of this textbook are reproducible as they were made using [Quarto](#)<sup>®</sup> that includes R code used to produce the figures, tables and results shown in the book. Quarto<sup>®</sup> is an open-source scientific and technical publishing system built on [Pandoc](#).

To learn more about Quarto books visit <https://quarto.org/docs/books>.

## License

This textbook is **free to use**, and is licensed under the [Creative Commons Attribution-NonCommercial-NoDerivs 4.0](#) License. If you'd like to give back, please consider reporting a typo or leaving a pull request at [github.com/rstudio-education/hopr](https://github.com/rstudio-education/hopr).

# 1 Examples

This is a new book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

```
1 + 1
```

```
[1] 2
```

## 1.1 intro 2

**Note** Not all functions have (or require) arguments.

## 1 Examples

**i** This book was originally created using [bookdown](#) and published at <https://rstudio-education.github.io/hopr/>. This site is a port of the original book source to the [Quarto](#) publishing system in order to provide an example of its use.

### Repetition

If you repeat a number in your index, R will return the corresponding value(s) more than once in your “subset.” This code will return the first row of `deck` twice:

```
x <- c(1, 2, 3)
x
```

Complete the following code to make a function that returns the first row of a data frame:

```
deal <- function(cards) {
  # ?
}
```

Use the preceding ideas to write a `shuffle` function. `shuffle` should take a data frame and return a shuffled copy of the data frame.

### ! Never attach

In R's early days, it became popular to use `attach()` on a data set once you had it loaded. Don't do this! `attach` recreates a computing environment similar to those used in other statistics applications like Stata and SPSS, which crossover users liked. However, R is not Stata or SPSS. R is optimized to use the R computing environment, and running `attach()` can cause confusion with some R functions.

What does `attach()` do? On the surface, `attach` saves you typing. If you attach the `deck` data set, you can refer to each of its variables by name; instead of typing `deck$face`, you can just type `face`. But typing isn't bad. It gives you a chance to be explicit, and in computer programming, explicit is good. Attaching a data set creates the possibility that R will confuse two variable names. If this occurs within a function, you're likely to get unusable results and an unhelpful error message to explain what happened.

⚠ = is an assignment operator

Be careful not to confuse `=` with `==`. `=` does the same thing as `<=`: it assigns a value to an object.

You can compare any two R objects with a logical operator; how-

## 1 Examples

ever, logical operators make the most sense if you compare two objects of the same data type. If you compare objects of different data types, R will use its coercion rules to coerce the objects to the same type before it makes the comparison.

### Exercise: How many Aces?

Extract the `face` column of `deck2` and test whether each value is equal to `ace`. As a challenge, use R to quickly count how many cards are equal to `ace`.

#### ! Important

Some R commands may take a long time to run. You can cancel a command once it has begun by pressing `ctrl + c`. Note that it may also take R a long time to cancel the command.

That's the basic interface for executing R code in RStudio. Think you have it? If so, try doing these simple tasks. If you execute everything correctly, you should end up with the same number that you started with:

1. Choose any number and add 2 to it.
2. Multiply the result by 3.
3. Subtract 6 from the answer.
4. Divide what you get by 3.

**i** Note

1. To create an R object, choose a name and then use the less-than symbol, `<`, followed by a minus sign, `-`, to save data into it. This combination looks like an arrow, `<-`. R will make an object, give it your name, and store in it whatever follows the arrow. So `a <- 1` stores 1 in an object named `a`.
2. When you ask R what's in `a`, R tells you on the next line.
3. You can use your object in new R commands, too. Since `a` previously stored the value of 1, you're now adding 1 to 2.

**!** Important

As you preview your book, chapters will be rendered and updated. However, if you make changes to global options (e.g. `_quarto.yml` or included files) you need to fully re-render your book to have all of the changes reflected. Consequently, you should always fully `quarto render` your book before deploying it, even if you have already previewed changes to some chapters with the preview server.

## 1 *Examples*



# 2 Introduction to R and R studio

## Learning Objectives

- Describe the purpose and use of each pane in the RStudio IDE
- Locate buttons, options in the RStudio IDE
- Work with R projects
- Understand the concept of functions and packages
- Use R as a calculator
- Use mathematical functions and relational operators
- Understand special values (NA, Inf, NaN)
- Understand the concept of objects in R
- Set legal names to objects

## *2 Introduction to R and R studio*

- Work with assignment operators

## **2.1 R and RStudio basics**

### **2.1.1 Installing R and RStudio**



Figure 2.1: Elephant

## 3 Activities

### Activity 1

- (a) Write R code to assign the value 20 to the name `num_1`.
- (b) Which of the following is a valid object name in R?
- `2.True`
  - `else`
  - `I_am_not_a_valid_name`
  - `I_am_a_Pretty#_name`

### 3 *Activities*

## 4 Summary

In summary, this book has no content whatsoever.

`1 + 1`

[1] 2

## 4 *Summary*



# References

Knuth, Donald E. 1984. "Literate Programming." *Comput. J.* 27 (2): 97–111. <https://doi.org/10.1093/comjnl/27.2.97>.

