BIOSTATS

Akira Terui

2022-09-13

Contents

Chapter 1

Introduction

This textbook aims to introduce fundamental statistical techniques and their applications to biological data. A unique aspect of this book is the "flipped-order" introduction. Usually, statistics starts with theory; yet, I found it difficult for those unfamiliar with statistics. I will start with a real example of the method, followed by an explanation of an underlying theory/concept. The author is an ecologist, so some methods in this book might not be popular in other fields.

Chapter 2

Project Management

2.1 R Project

For this entire book, I will use `R Project` as a fundamental unit of work-space, in which all the relevant materials (e.g., R scripts `.R` and data files) are assembled together. There are many ways to organize your project, but I usually make a single `R Project` for a collection of scripts and data that will lead to a single publication (see example here). To setup an `R Project` you will need *R Studio* in addition to base *R*. While *R* is a stand-alone software, I strongly recommend to use it with *R Studio*. *R Studio* has many functions that help your data analysis. *R* and *R Studio* can be installed from the following websites:

- R (you can choose any CRAN mirror to download)
- R Studio

Once you open *R Studio*, you will see the following interface (Figure 2.1). There are three major panels in its first appearance – **Console**, **Environment**, and **Files**. **Console** is the place where you write your codes and execute calculation/data manipulation/analysis. **Environment** lists items you saved as an object. **Files** list any files in a designated location in your computer.

Although you can work on your data as it appears, **it is actually a BAD idea**. As you work on your project, numerous materials will be generated. How do you manage files? If you randomly locate those materials within your computer, you will lose necessary items sooner or later. For this reason, I assemble all the relevant materials in a single `R Project`. You can create a new `R Project` with the following procedure.

a. Go to `File > New Project` on the top menu
b. Select `New Directory`
c. Select `New Project`

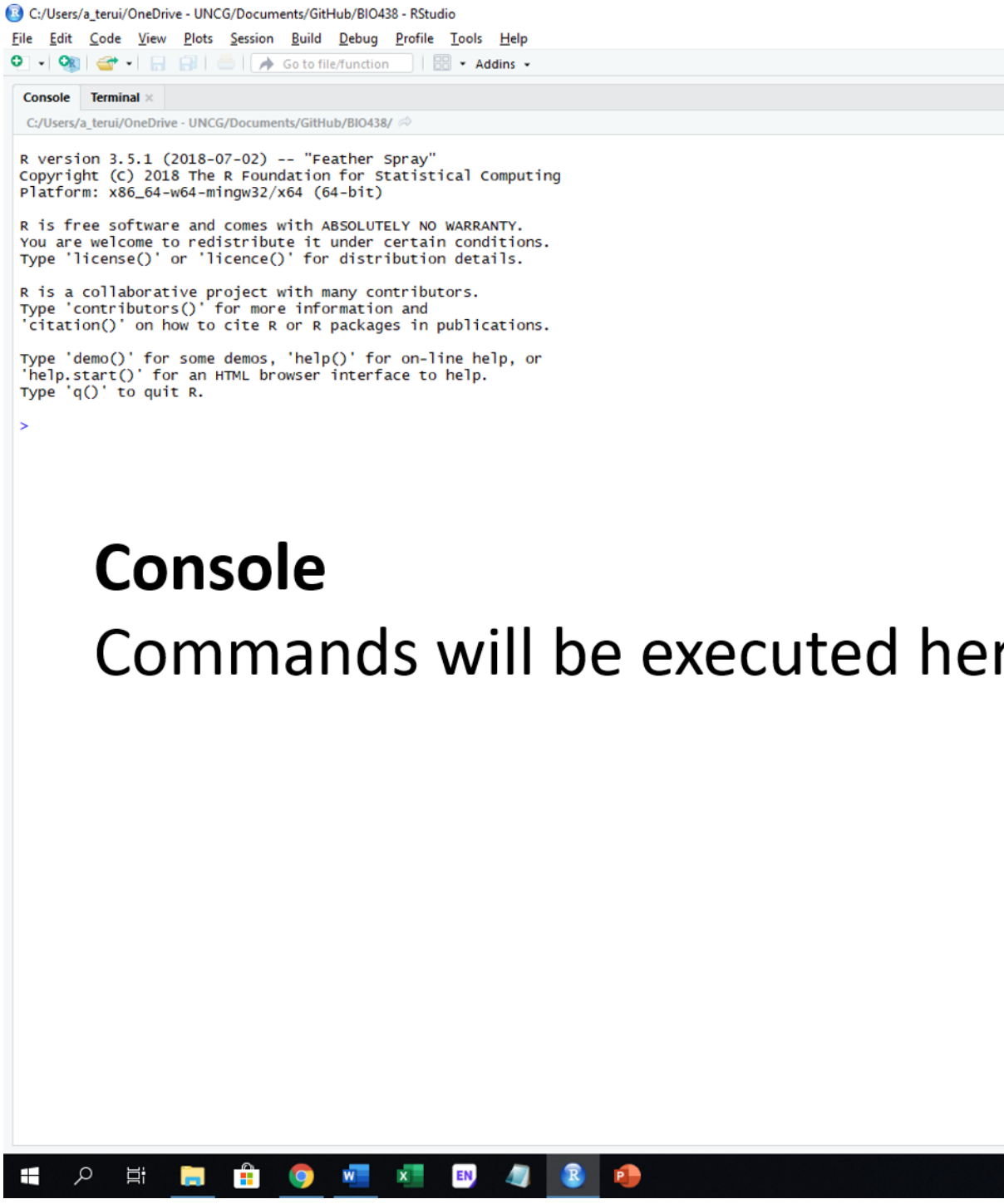


Figure 2.1: R Studio interface.

A new window pops up and prompts you to name a directory with a location in your computer. Click `Browse` to select a location for the directory.

IMPORTANT: When you locate your project directories in your computer, I would strongly recommend to create a designated space. For example, in my computer, I have a folder named `/r_project` in which all the `R Project` directories are located.

2.2 Internal Structure

The internal structure of an `R Project` is extremely important to navigate yourself (others once it's published). `R Project` will be composed of multiple types of files, typically `.R`, `.csv`, `.rds`, `.Rmd` among others. Unless those files are arranged in an organized manner, it is VERY LIKELY to make severe errors in coding. So take this seriously. Table 2.1 is my suggested subdirectory structure.

Table 2.1: Suggested internal structure of `R Project`

Name	Content
`readme.md`	Markdown file explaining contents in the `R Project`. A derived file from `README.Rmd`.
`/code`	Sub-directory for R scripts (`.R`).
`/data_raw`	Sub-directory for raw data before data manipulation (`.csv` or other formats). Files in this sub-directory MUST NOT be modified unless there are changes to raw data entries.
`/data_format`	Sub-directory for formatted data (`.csv`, `.rds`, or other formats).
`/output`	Sub-directory for result outputs (`.csv`, `.rds`, or other formats). This may include statistical estimates from linear regression models etc.
`/rmd`	(Optional) Sub-directory for Rmarkdown files (`.Rmd`). Rmarkdown allows seamless integration of R scripts and text.

2.3 File Name

As you proceed, the number of files will increase, perhaps exponentially. It is therefore critical to have **consistent naming rules** for your files. Here are some recommendations:

- **NO SPACE.** Instead, use underscore or hyphen.
  - Do: `script_week1.R` `script-week1.R`
  - Don't: `script week1.R`

- **NO UPPERCASE.** Use lowercase only for file names.
  - Do: `script_week1.R`
  - Don't: `Script_week1.R`
- **BE CONSISTENT.** Apply consistent naming rules within a project.
  - Do: R scripts for figures always start with a common prefix, e.g., `figure_XXX.R` `figure_YYY.R`(`XXX` and `YYY` specifies further details).
  - Don't: R scripts for figures start with random text, e.g., `XXX_fig.R` , `Figure_Y2.R` , `plotB.R`.

Chapter 3

DATA STRUCTURE

3.1 Overview

R has 6 basic **data types**.

- character: `"aquatic"`, `"ecology"` (no order)
- factor: similar to character, but has *levels* (alphabetically ordered by default)
- numeric: `20.0` , `15.5`
- integer: `3`, `7`
- logical: `TRUE` , `FALSE`
- complex: `1+2i` (complex numbers with real and imaginary parts)

These elements form one of the following **data structures**.

- **vector**: a series of elements. A single data type is allowed in a single vector
- **matrix**: elements organized into rows and columns. A single data type is allowed in a single matrix
- **data frame**: looks similar to a matrix, but allows different data types in different columns

3.2 Vector

3.2.1 Create Vector

Below are examples of atomic character vectors, numeric vectors, integer vectors, etc. There are many ways to create vector data. The following examples use `c()`, `:`, `seq()`, `rep()`:

```
## [1] 1 3 4 8
```

```
## [1] "a" "b" "c"
```

```
## [1]  TRUE FALSE FALSE
```

```
## [1] 1 2 3 4 5
```

```
## [1] 2 2 2 2 2
```

```
## [1] "a" "a" "a" "a" "a"
```

```
## [1] 1 2 3 4 5
```

```
##  [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8
## [20] 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3 4.4 4.5 4.6 4.7
## [39] 4.8 4.9 5.0
```

3.2.2 Check Features

R provides many functions to examine features of vectors and other objects, for example:

- `class()` - what kind of object (data structure) is it (high-level)?
- `typeof()` - what is the object's data type (low-level)?
- `attributes()` - does it have any metadata?
- `length()` - how long is it? What about two dimensional objects?
- `sum()` - what is the summed number of the data?
- `mean()` - what is the average number of the data?

Numeric Vector

```
## [1] 1.2 3.1 4.0 8.2
```

```
## [1] "numeric"
```

```
## [1] "double"
```

```
## [1] 4
```

```
## [1] 16.5
```

```
## [1] 4.125
```

Character Vector

```
## [1] "character"
```

```
## [1] 3
```

3.2.3 Access

Element ID

Use brackets `[]` when accessing specific elements in an object. For example, if you want to access element #2 in the vector `x`, you may specify as `x[2]`:

```
## [1] 2
```

```
## [1] 2 2
```

```
## [1] 2 3 2
```

Equation
R provides many ways to access elements that suffice specific conditions. You can use mathematical symbols to specify what you need, for example:

- `==` equal
- `>` larger than
- `>=` equal & larger than
- `<` smaller than
- `<=` equal & smaller than
- `which()` a function that returns element # that suffices the specified condition

The following examples return a logical vector indicating whether each element in x suffices the specified condition:

```
## [1]  TRUE  TRUE FALSE  TRUE FALSE
```

```
## [1] FALSE FALSE  TRUE FALSE  TRUE
```

You can access elements that suffice the specified condition using brackets, for example:

```
## [1] 2 2 2
```

```
## [1] 3 5
```

Using `which()`, you can see which elements (i.e., #) matches what you need:

```
## [1] 1 2 4
```

```
## [1] 3 5
```

3.3 Matrix

3.3.1 Create Matrix

Matrix is a set of elements (*single data type*) that are organized into rows and columns:

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

3.3.2 Check Features

R provides many functions to examine features of matrix data, for example:

- `class()` what kind of object (data type) is it (high-level)?
- `typeof()` what is the object's data type (low-level)?
- `attributes()` does it have any metadata?
- `dim()` how long are rows and columns?
- `rowSums()` what is the summed number of the data for each row?
- `colSums()` what is the summed number of the data for each column?

Integer Matrix

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
## [1] "matrix" "array"
```

```
## [1] "integer"
```

```
## [1] 3 3
```

Character Matrix

```
##      [,1] [,2]
## [1,] "a"  "d"
## [2,] "b"  "e"
## [3,] "c"  "f"
```

```
## [1] "matrix" "array"
```

```
## [1] "character"
```

```
## [1] 3 2
```

3.3.3 Access

When accessing matrix elements, you need to pick row(s) and/or column(s), for example:

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
```

```
## [3,]    3    6    9
```

```
## [1] 8
```

```
## [1] 2 5 8
```

```
##      [,1] [,2] [,3]
## [1,]    2    5    8
## [2,]    3    6    9
```

```
##      [,1] [,2]
## [1,]    4    7
## [2,]    5    8
## [3,]    6    9
```

You can assess each element with mathematical expressions just like vectors:

```
##       [,1]  [,2]  [,3]
## [1,] FALSE FALSE FALSE
## [2,]  TRUE FALSE FALSE
## [3,] FALSE FALSE FALSE
```

```
##       [,1] [,2] [,3]
## [1,] FALSE TRUE TRUE
## [2,] FALSE TRUE TRUE
## [3,]  TRUE TRUE TRUE
```

However, care must be taken when accessing elements, as it will be automatically converted to vector data:

```
## [1] 2
```

```
## [1] 3 4 5 6 7 8 9
```

`which()` needs an additional argument to return both row and column #:

```
##      row col
## [1,]   2   1
```

```
##      row col
## [1,]   3   1
## [2,]   1   2
## [3,]   2   2
## [4,]   3   2
## [5,]   1   3
## [6,]   2   3
## [7,]   3   3
```

3.4 Data Frame

Data frame is a set of elements that are organized into rows and columns, but differ from matrix in several ways.

- it allows *multiple data types* in different columns
- each column has its *name*
- you can access columns by name (using `$`)

Data frame is the most common data structure when manipulating ecological data. A data set loaded from a spread sheet (we will address this later) will be automatically recognized as a data frame. Here is an example:

Create a data frame
In the following example, variables `x` and `y` are organized into a single data frame `dat`. Variable are renamed when creating a data frame composed of `x` and `y`.

```
##    LakeType  TSS
## 1  Pristine  1.2
## 2  Pristine  2.2
## 3 Disturbed 10.9
## 4 Disturbed 50.0
## 5  Pristine  3.0
```

Call column names

```
## [1] "LakeType" "TSS"
```

Access by columns

```
## [1] "Pristine"  "Pristine"  "Disturbed" "Disturbed" "Pristine"
```

```
## [1]  1.2  2.2 10.9 50.0  3.0
```

You can access elements like a matrix as well:

```
## [1] "Pristine"  "Pristine"  "Disturbed" "Disturbed" "Pristine"
```

```
##   LakeType TSS
## 1 Pristine 1.2
```

```
##    LakeType  TSS
## 2  Pristine  2.2
## 4 Disturbed 50.0
```

3.5 Exercise

Download a template (R script file).

3.5.1 Vector

a. Create three numeric vectors with length 3, 6 and 20, respectively. Each vector must be created using different functions in R.
b. Create three character vectors with length 3, 6 and 20, respectively. Each vector must be created using different functions in R.
c. Copy the following script to your R script and perform the following analysis:

- Identify element IDs of `x` that are greater than `2.0`
- Identify element values of `x` that are greater than `2.0`

3.5.2 Matrix

a. Create a numeric matrix with 4 rows and 4 columns. Each column must contain identical elements.
b. Create a numeric matrix with 4 rows and 4 columns. Each row must contain identical elements.
c. Create a character matrix with 4 rows and 4 columns. Each column must contain identical elements.
d. Create a character matrix with 4 rows and 4 columns. Each row must contain identical elements.
e. Copy the following script to your R script and perform the following analysis:

- Identify element IDs of `x` that are greater than `2.0` (**specify row and column IDs**)
- Identify element values of `x` that are greater than `2.0` and calculate the mean.

3.5.3 Data Frame

a. Create a data frame of 3 variables with 10 elements (name variables as `x`, `y` and `z`. `x` must be `character` while `y` and `z` must be `numeric`.
b. Check the data structure (higher-level) of `x`, `y` and `z`
c. Copy the following script to your R script and perform the following analysis:

- Calculate the means of `temperature` and `abundance` for states `VA` and `NC` separately - use `tapply()` function

Chapter 4

VISUALIZATION

4.1 Overview

R has a number of functions that help visualize data. `graphics` provides R functions for base graphics (list of functions). I will use `iris`, the built-in data set in R, to show how `graphics` functions work.

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1            5.1         3.5          1.4         0.2     setosa
## 2            4.9         3.0          1.4         0.2     setosa
## 3            4.7         3.2          1.3         0.2     setosa
## 4            4.6         3.1          1.5         0.2     setosa
## 5            5.0         3.6          1.4         0.2     setosa
## 6            5.4         3.9          1.7         0.4     setosa
## 7            4.6         3.4          1.4         0.3     setosa
## 8            5.0         3.4          1.5         0.2     setosa
## 9            4.4         2.9          1.4         0.2     setosa
## 10           4.9         3.1          1.5         0.1     setosa
## 11           5.4         3.7          1.5         0.2     setosa
## 12           4.8         3.4          1.6         0.2     setosa
## 13           4.8         3.0          1.4         0.1     setosa
## 14           4.3         3.0          1.1         0.1     setosa
## 15           5.8         4.0          1.2         0.2     setosa
## 16           5.7         4.4          1.5         0.4     setosa
## 17           5.4         3.9          1.3         0.4     setosa
## 18           5.1         3.5          1.4         0.3     setosa
## 19           5.7         3.8          1.7         0.3     setosa
## 20           5.1         3.8          1.5         0.3     setosa
## 21           5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
```

```
## 23          4.6         3.6          1.0         0.2     setosa
## 24          5.1         3.3          1.7         0.5     setosa
## 25          4.8         3.4          1.9         0.2     setosa
## 26          5.0         3.0          1.6         0.2     setosa
## 27          5.0         3.4          1.6         0.4     setosa
## 28          5.2         3.5          1.5         0.2     setosa
## 29          5.2         3.4          1.4         0.2     setosa
## 30          4.7         3.2          1.6         0.2     setosa
## 31          4.8         3.1          1.6         0.2     setosa
## 32          5.4         3.4          1.5         0.4     setosa
## 33          5.2         4.1          1.5         0.1     setosa
## 34          5.5         4.2          1.4         0.2     setosa
## 35          4.9         3.1          1.5         0.2     setosa
## 36          5.0         3.2          1.2         0.2     setosa
## 37          5.5         3.5          1.3         0.2     setosa
## 38          4.9         3.6          1.4         0.1     setosa
## 39          4.4         3.0          1.3         0.2     setosa
## 40          5.1         3.4          1.5         0.2     setosa
## 41          5.0         3.5          1.3         0.3     setosa
## 42          4.5         2.3          1.3         0.3     setosa
## 43          4.4         3.2          1.3         0.2     setosa
## 44          5.0         3.5          1.6         0.6     setosa
## 45          5.1         3.8          1.9         0.4     setosa
## 46          4.8         3.0          1.4         0.3     setosa
## 47          5.1         3.8          1.6         0.2     setosa
## 48          4.6         3.2          1.4         0.2     setosa
## 49          5.3         3.7          1.5         0.2     setosa
## 50          5.0         3.3          1.4         0.2     setosa
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
## 56          5.7         2.8          4.5         1.3 versicolor
## 57          6.3         3.3          4.7         1.6 versicolor
## 58          4.9         2.4          3.3         1.0 versicolor
## 59          6.6         2.9          4.6         1.3 versicolor
## 60          5.2         2.7          3.9         1.4 versicolor
## 61          5.0         2.0          3.5         1.0 versicolor
## 62          5.9         3.0          4.2         1.5 versicolor
## 63          6.0         2.2          4.0         1.0 versicolor
## 64          6.1         2.9          4.7         1.4 versicolor
## 65          5.6         2.9          3.6         1.3 versicolor
## 66          6.7         3.1          4.4         1.4 versicolor
## 67          5.6         3.0          4.5         1.5 versicolor
## 68          5.8         2.7          4.1         1.0 versicolor
```

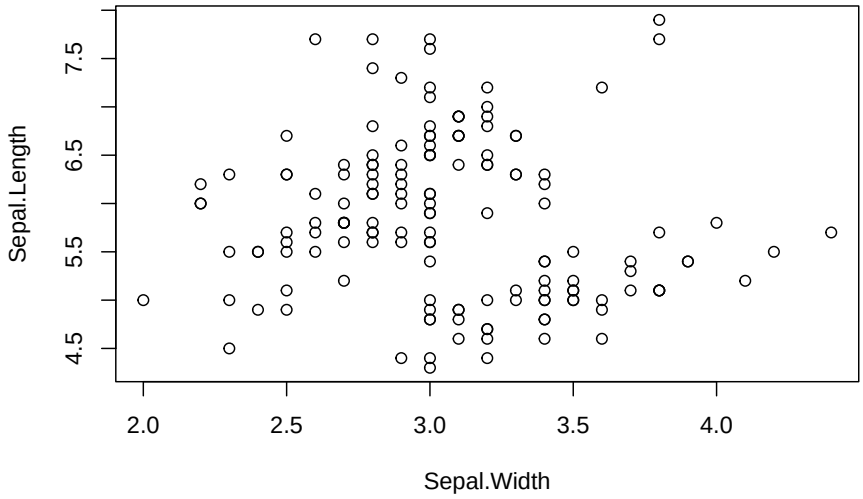
```
## 69          6.2         2.2          4.5         1.5 versicolor
## 70          5.6         2.5          3.9         1.1 versicolor
## 71          5.9         3.2          4.8         1.8 versicolor
## 72          6.1         2.8          4.0         1.3 versicolor
## 73          6.3         2.5          4.9         1.5 versicolor
## 74          6.1         2.8          4.7         1.2 versicolor
## 75          6.4         2.9          4.3         1.3 versicolor
## 76          6.6         3.0          4.4         1.4 versicolor
## 77          6.8         2.8          4.8         1.4 versicolor
## 78          6.7         3.0          5.0         1.7 versicolor
## 79          6.0         2.9          4.5         1.5 versicolor
## 80          5.7         2.6          3.5         1.0 versicolor
## 81          5.5         2.4          3.8         1.1 versicolor
## 82          5.5         2.4          3.7         1.0 versicolor
## 83          5.8         2.7          3.9         1.2 versicolor
## 84          6.0         2.7          5.1         1.6 versicolor
## 85          5.4         3.0          4.5         1.5 versicolor
## 86          6.0         3.4          4.5         1.6 versicolor
## 87          6.7         3.1          4.7         1.5 versicolor
## 88          6.3         2.3          4.4         1.3 versicolor
## 89          5.6         3.0          4.1         1.3 versicolor
## 90          5.5         2.5          4.0         1.3 versicolor
## 91          5.5         2.6          4.4         1.2 versicolor
## 92          6.1         3.0          4.6         1.4 versicolor
## 93          5.8         2.6          4.0         1.2 versicolor
## 94          5.0         2.3          3.3         1.0 versicolor
## 95          5.6         2.7          4.2         1.3 versicolor
## 96          5.7         3.0          4.2         1.2 versicolor
## 97          5.7         2.9          4.2         1.3 versicolor
## 98          6.2         2.9          4.3         1.3 versicolor
## 99          5.1         2.5          3.0         1.1 versicolor
## 100         5.7         2.8          4.1         1.3 versicolor
## 101         6.3         3.3          6.0         2.5  virginica
## 102         5.8         2.7          5.1         1.9  virginica
## 103         7.1         3.0          5.9         2.1  virginica
## 104         6.3         2.9          5.6         1.8  virginica
## 105         6.5         3.0          5.8         2.2  virginica
## 106         7.6         3.0          6.6         2.1  virginica
## 107         4.9         2.5          4.5         1.7  virginica
## 108         7.3         2.9          6.3         1.8  virginica
## 109         6.7         2.5          5.8         1.8  virginica
## 110         7.2         3.6          6.1         2.5  virginica
## 111         6.5         3.2          5.1         2.0  virginica
## 112         6.4         2.7          5.3         1.9  virginica
## 113         6.8         3.0          5.5         2.1  virginica
## 114         5.7         2.5          5.0         2.0  virginica
```

```
## 115          5.8         2.8          5.1         2.4  virginica
## 116          6.4         3.2          5.3         2.3  virginica
## 117          6.5         3.0          5.5         1.8  virginica
## 118          7.7         3.8          6.7         2.2  virginica
## 119          7.7         2.6          6.9         2.3  virginica
## 120          6.0         2.2          5.0         1.5  virginica
## 121          6.9         3.2          5.7         2.3  virginica
## 122          5.6         2.8          4.9         2.0  virginica
## 123          7.7         2.8          6.7         2.0  virginica
## 124          6.3         2.7          4.9         1.8  virginica
## 125          6.7         3.3          5.7         2.1  virginica
## 126          7.2         3.2          6.0         1.8  virginica
## 127          6.2         2.8          4.8         1.8  virginica
## 128          6.1         3.0          4.9         1.8  virginica
## 129          6.4         2.8          5.6         2.1  virginica
## 130          7.2         3.0          5.8         1.6  virginica
## 131          7.4         2.8          6.1         1.9  virginica
## 132          7.9         3.8          6.4         2.0  virginica
## 133          6.4         2.8          5.6         2.2  virginica
## 134          6.3         2.8          5.1         1.5  virginica
## 135          6.1         2.6          5.6         1.4  virginica
## 136          7.7         3.0          6.1         2.3  virginica
## 137          6.3         3.4          5.6         2.4  virginica
## 138          6.4         3.1          5.5         1.8  virginica
## 139          6.0         3.0          4.8         1.8  virginica
## 140          6.9         3.1          5.4         2.1  virginica
## 141          6.7         3.1          5.6         2.4  virginica
## 142          6.9         3.1          5.1         2.3  virginica
## 143          5.8         2.7          5.1         1.9  virginica
## 144          6.8         3.2          5.9         2.3  virginica
## 145          6.7         3.3          5.7         2.5  virginica
## 146          6.7         3.0          5.2         2.3  virginica
## 147          6.3         2.5          5.0         1.9  virginica
## 148          6.5         3.0          5.2         2.0  virginica
## 149          6.2         3.4          5.4         2.3  virginica
## 150          5.9         3.0          5.1         1.8  virginica
```

4.2 `plot()`

When plotting data, what's you need is to specify the **formula**. For example, if you want to visualize the relationship between `x` and `y` (`y` on the vertical axis and `x` on the horizontal axis), the formula would be `y ~ x` (the left side of the formula will be on the vertical axis). In the `iris` data set, the following columns are available: `Sepal.Length` `Sepal.Width` `Petal.Length` `Petal.Width` `Species`. In the following example, the relationship between `Sepal.Length` and `Sepal.Width` is

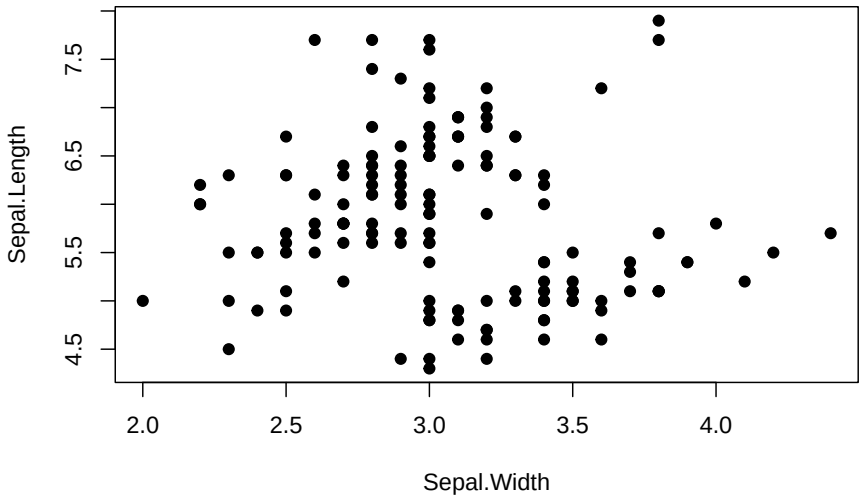
plotted:



The argument `data =` tells the function the data set from which the variables (`Sepal.Length` and `Sepal.Width`) are extracted. The above figure is the default setting, and you may customize it as necessary. Below is an example of how you may customize:

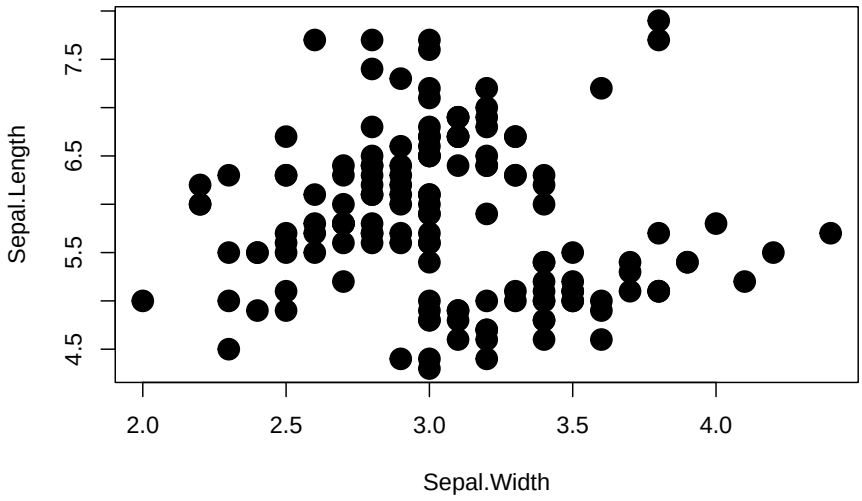
4.2.1 Symbol

`pch` argument. Choose from `1` to `25` (google **r plot pch** for details)



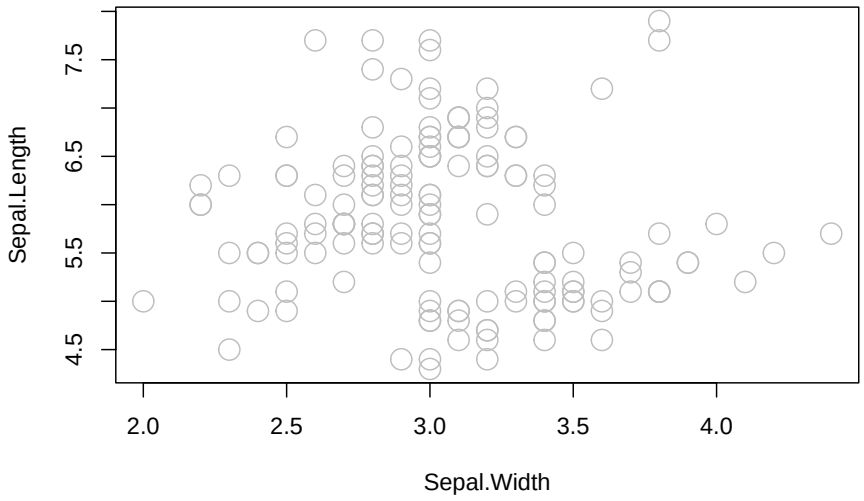
4.2.2 Symbol size

`cex` argument. `cex = 1` is the default value. `cex = 2` is as twice large as default value.



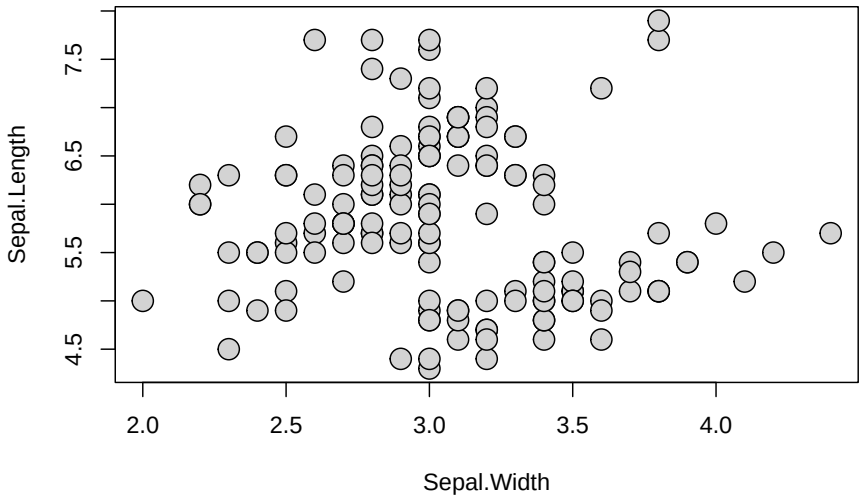
4.2.3 Symbol color (border)

`col` argument (quote `"color name"` when specifying). Google **r color name** for color options.



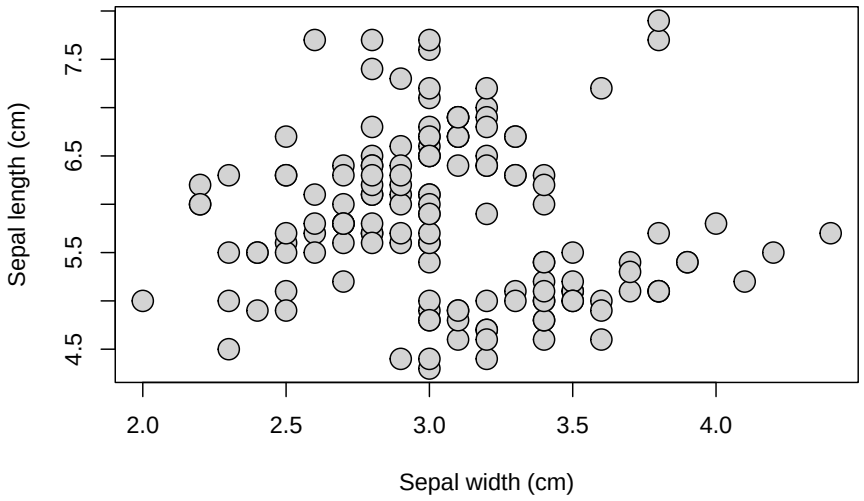
4.2.4 Symbol color (fill)

`bg` argument (quote `"color name"` when specifying). Available for a subset of symbol options (some symbols have pre-defined filled color).



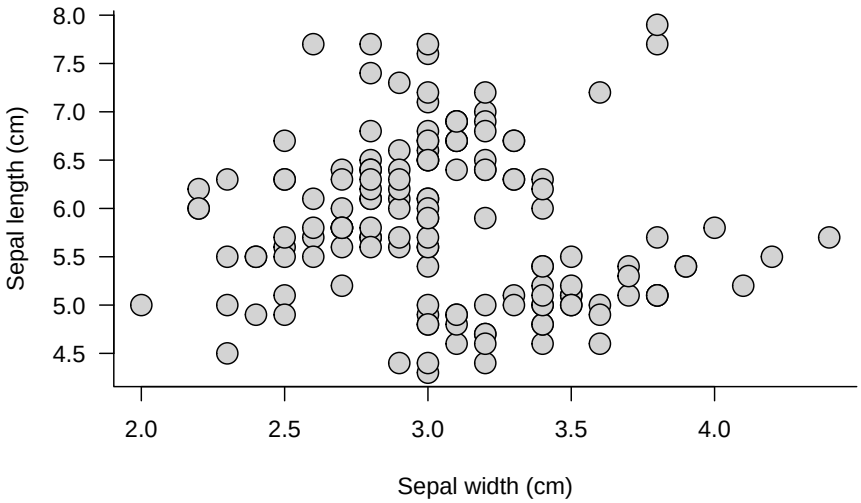
4.2.5 Label

`ylab` or `xlab` arguments. Provide `"quoted text"`.



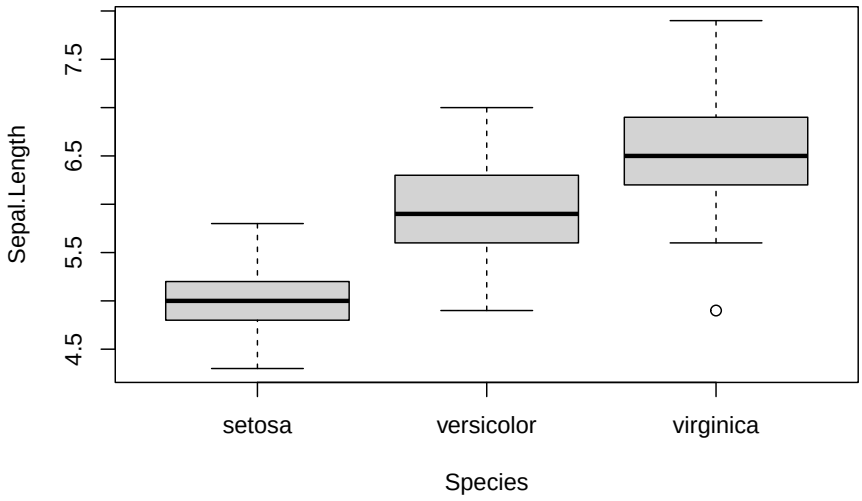
4.2.6 Axis

Delete axes with `axes = F` and re-draw with `box()` and `axis()` functions.



4.3 boxplot()

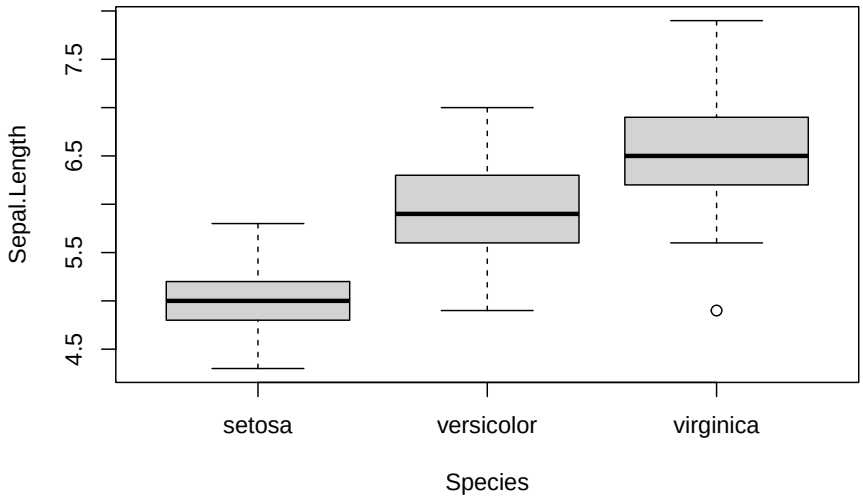
`boxplot()` is used when the x-axis is factor-type data (by default, `plot()` will produce a boxplot when x-axis is a factor variable). In the `iris` dataset, the column `Species` is a factor variable. Compare `Sepal.Length` among species using `boxplot()`.



You can customize as in `plot()`, but slighlty different.

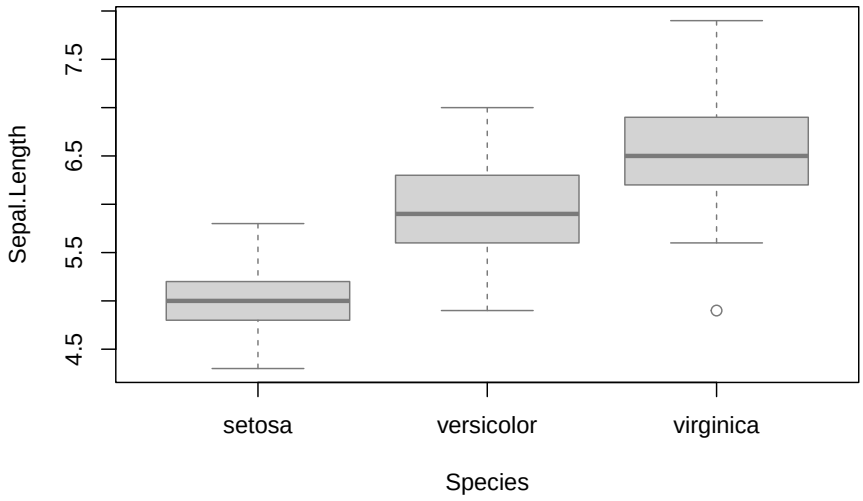
4.3.1 Box color

`col` argument.



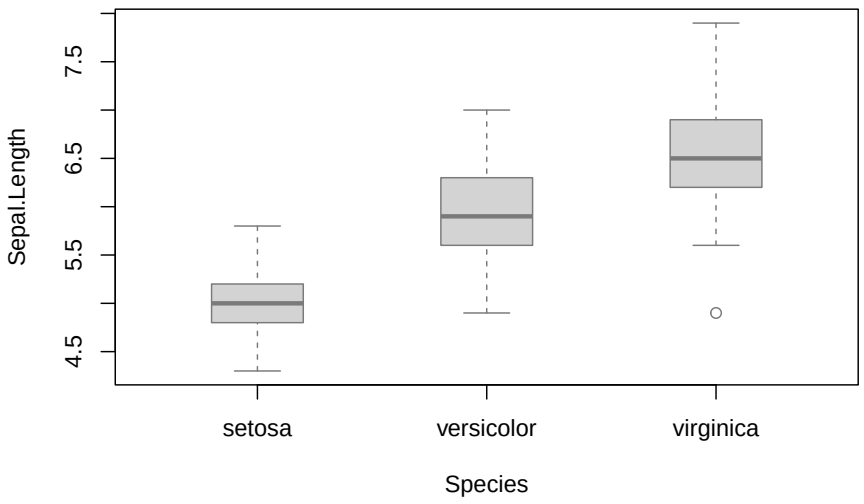
4.3.2 Border color

`border` argument.



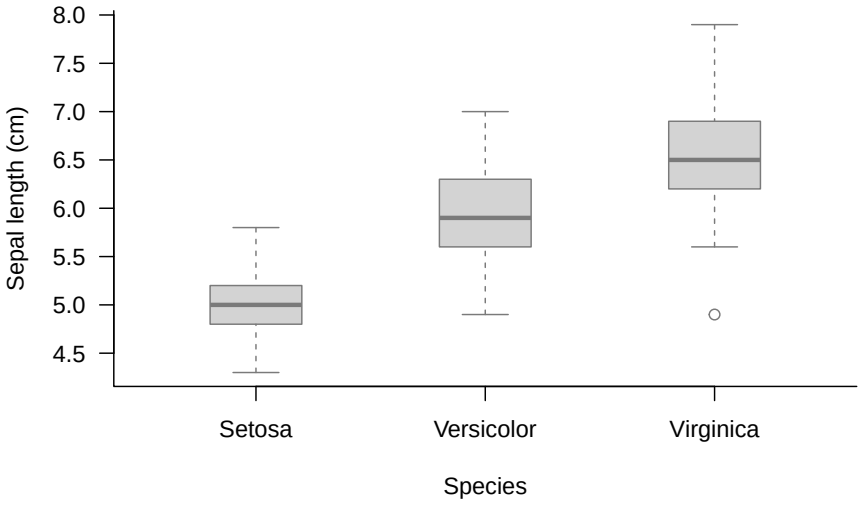
4.3.3 Box width

`boxwex` argument.



4.3.4 Axis

Delete axes with `axes = F` and re-draw with `box()` and `axis()` functions.



4.4 Exercise

Download a template (R script file).

4.4.1 `plot()` function

a. Plot the relationship between `Sepal.Width` and `Petal.Width`
b. Turn symbol color (border) into red.
c. Turn symbol color (fill) into red (set `pch = 21`).
d. Make symbol size larger.
e. Make L-shaped plot border (delete border lines on upper and right sides)

4.4.2 `boxplot()` function

a. Plot the relationship between `Petal.Width` and `Species`.
b. Turn box color (border) into `blue`.
c. Make box width narrower.