

Data in R

Sean Hellingman ©

Data Visualization and Manipulation through Scripting (ADSC1010)

shellingman@tru.ca

Fall 2024



THOMPSON RIVERS UNIVERSITY

Topics

- 2 Introduction
- 3 Types of Data
- 4 Importing Data

- 5 Exporting Data
- 6 Exercises and References

Introduction

- It is very important to understand the different types (classes) of data in R.
- There are different rules and operators that apply to different data types.
 - Can't use + to add two characters together.
- There are *six* basic types of data in R.

1. Numeric

- Numeric data are numbers that contain a decimal.
- *They can technically be whole numbers also.*
- Example: `Num.Object <- 5.25`

2. Integers

- Integers are data that **do not** contain a decimal.
- Example: `Int.Object <- 5`

3. Logical

- Logical (Boolean) data take on the value of either TRUE or FALSE.
- Another type of logical data called NA represents missing data.
- We will discuss a bit more about missing data later.

4. Character

- Character data are used to represent strings.
- Character strings are something like a word or multiple words.
- Characters can be ordered and turned into *factors*.

5. Complex

- The complex data type is used to specify purely imaginary values in R.
- The suffix `i` is used to denote the imaginary part.
- **We will not be working with this data type in this course.**

6. Raw

- The raw data type specifies values as raw bytes.
- Can use `charToRaw()` to convert character data to raw data.
- Can use `rawToChar()` to convert raw data to character data.
- **We will not be working with this data type in this course.**

Example 1

- Use the `class()` function in R to determine the class of the following objects:
 - `V1 <- 3.14`
 - `V2 <- "3.14"`
 - `V3 <- "Thompson Rivers University"`
 - `V4 <- TRUE`
 - `V5 <- "TRUE"`
 - `V6 <- NA`

Logical Tests for Class

- Can use `is.[className]()` to test if an object is a specific class.
- Examples:
 - `is.numeric()` returns TRUE if the object is numeric.
 - `is.character()` returns TRUE if the object is a character.
 - `is.logical()` returns TRUE if the object is logical.

Converting Between Classes

- Can use `as.[className]()` to convert (coerce) an object to a specific class.
- Examples:
 - `as.numeric()` converts the object to numeric.
 - `as.character()` converts the object to a character.
- **Be careful** you might end up with some unexpected results by doing this.

Logical Tests & Coercion Functions

Type	Logical Test	Coercing
Character	<code>is.character</code>	<code>as.character</code>
Numeric	<code>is.numeric</code>	<code>as.numeric</code>
Logical	<code>is.logical</code>	<code>as.logical</code>
Factor	<code>is.factor</code>	<code>as.factor</code>
Complex	<code>is.complex</code>	<code>as.complex</code>

Example 2

- In R, run the following code:
 - `V1 <- 3.14`
 - `is.character(V1)`
 - `is.numeric(V1)`
 - `V1.Char <- as.character(V1)`
 - `V1.Char`
 - `is.numeric(V1.Char)`
 - `V1.num <- as.numeric(V1.Char)`
- What happens?

Importing and Exporting Data

Importing Data

- Most of the datasets you will work with come from outside of R.
- There are example dataset that are built into R packages.
- Import from saved files or spreadsheets (Typically CSV files).
- Can also import data directly from the internet (stock data).

Importing Data from R Packages

- Some R packages have built in datasets that are downloaded when the package is.
- To load a dataset from an existing R package:
 - 1 Load the package in R using `library(package_name)`
 - 2 load the dataset using `data("package_name")`

Example 3

- Load the *MASS* package in R.
- Load the *birthwt* dataset from the *MASS* package.
- Spend a bit of time familiarising yourself with the data.

Importing Data from CSV Files

- The `read.csv()` command can read CSV files.
 - `data <- read.csv("filename.csv", header = TRUE)` or
`data <- read.csv("filename.csv")`
if your CSV file has a header.
 - `data <- read.csv("filename.csv", header = FALSE)`
if your CSV file doesn't have a header.
- Note that we did not specify the file location
 - This only works if the CSV file is stored in the **Working Directory**.
 - If the file is not in the Working Directory, need to specify the file location.
 - Or change Working Directory to the file location using `setwd("file path")` first.

Example 4

- Download the *Football22.csv* dataset from moodle.
- Save the file in an appropriate location to work with.
- Set the proper working directory in R Studio.
- Import the dataset into your R markdown environment.

Importing Other Data Types

- R can also read data from other file types, such as:
 - `read.fwf()` reads fixed-width records.
 - `read.table()` reads tabular data files.
 - MySQL database.
 - R can also read Tabular or CSV data from the Web.
 - Or read data from HTML tables.
- More information available in the *An introduction to R* (1) textbook or online.

Importing Yahoo Finance Data

- Use the *quantmod* package to get data from Yahoo Finance.
 - Load the package: `library(quantmod)`
 - Select your symbols: `vecOfSyms = c("AAPL", "MSFT")`
 - Run the `getSymbols()` function on selected dates.
 - `getSymbols(vecOfSyms, source = "yahoo", from = "1999-12-31", to = "2000-02-01")`

Example 5

- Import the Apple (AAPL) and Microsoft (MSFT) Yahoo Finance data from 2012-12-31 to 2013-12-31.

Exporting R Data

- You can directly save R data objects within your working directory.
 - To save data: `save(data1, file = "data.RData")`
 - This is less computationally demanding than writing a new CSV file.
 - **Remember:** R will save the data to your working directory.

Writing a New CSV File

- To save a matrix or data frame in a file using the CSV format:

```
write.csv(data, file="filename", row.names=FALSE)
```

or

```
write.csv(data, file="filename", row.names=TRUE)
```

- Generally, we will use the second option as our variables will usually have names.
- Note: the *readr* package is required for this.

Example 6

- Use the `subset()` function to create a dataframe with only La Liga teams from the *Football22.csv* dataset.
 - `Football22_LaLiga <- subset(Football22, Football22$League == "La Liga")`
- Use the `write.csv()` to create a new CSV for the La Liga data.
- Use the `save()` to save the La Liga data as R data.
- Navigate to the file on your computer and notice the differences.

Other Ways to Export Data

- There are other ways to export data:
 - `write.table()`
 - `fwrite()` from the *read.table* package
 - `write_tsv()` for tab delimited files

Exercise 1

- What do you expect from each of the following? Execute and confirm.
 - `is.integer(44L)`
 - `is.integer(44)`
 - `is.integer(as.integer(44))`
 - `as.integer(is.integer(44))`
 - `is.integer(as.integer(is.integer(44)))`
 - `is.integer(as.numeric(is.integer(44)))`
 - `is.integer(44.2L)`

Exercise 2

- What do you expect from each of the following? Execute and confirm.
 - `is.logical(TRUE)`
 - `is.logical("TRUE")`
 - `as.numeric(TRUE)`
 - `is.numeric(as.numeric(TRUE))`
 - `is.integer(as.numeric(TRUE))`
 - `as.logical(as.numeric(FALSE))`
 - `as.numeric("Go Canucks Go")`
 - `is.numeric(as.numeric("Go Canucks Go"))`

Exercise 3

- Load the *Boston* dataset from the *MASS* package in R.
- Familiarise yourself with the data, is there anything interesting?
- Practice changing directories and saving the data in different formats.
 - Feel free to delete the saved data after the exercise.

References & Resources

- ① Douglas, A., Roos, D., Mancini, F., Couto, A., & Lusseau, D. (2023). *An introduction to R*. Retrieved from <https://intro2r.com/>
- <https://www.r-bloggers.com/2021/09/r-data-types/>
- <https://cran.r-project.org/web/packages/MASS/MASS.pdf>