

Contents

About	3
How to use this notebook	5
1 Introducing Statistical Programming with R and RStudio	7
1.1 Creating and Exploring Vector Objects	7
1.2 Importing and Exporting Data	8
1.3 <u>Exercise 2.5</u>	9
1.4 <u>Exercise 2.6</u>	9
1.5 Bonus Task: Importing Data from Github Repositories	9
2 The Structure of Data Objects	11
3 Manipulating and Tidying Data	13
4 Reproducible Workflows with R Markdown	15
5 Data Visualisation	17
6 Working with Special Data Types: Text Data	19
7 Programming with R	21
8 Practising Data Cleaning and Visualisation: Formative Peer-Review Case Study	23

About

Welcome to SOST70023 Data Cleaning and Visualisation using R! This notebook will host the materials for all practical exercises for this course unit.

How to use this notebook

Chapter 1

Introducing Statistical Programming with R and RStudio

To complete the exercises, ensure that you work in your newly created project in RStudio and writing your answers in your new R script.

1.1 Creating and Exploring Vector Objects

1.1.1 Exercise 1.1

Create a vector object that contains the following values: 70, 8, 50, 100. Name this vector `num_vct` and print the contents.

- a. Use the `class()` function to find out what type of object `num_vct` is.
- b. Multiply this vector by 2.
- c. Divide the vector by 10.
- d. Subtract 5 from the vector.
- e. Add 8 to the vector.

1.1.2 Exercise 1.2

Create a character vector object that contains the following: I, love, programming, with, R. Name this vector `char_vct` and print the contents.

- a. Use the `class()` function to find out what type of object `char_vct` is.
- b. Perform tasks b to e from Exercise 1. What results do you obtain? Why?

1.1.3 Exercise 1.3

Create a new numeric vector that contains the following values: 5, 7, 9, 10. Name it `num_vct2` and print the contents.

- a. Add `num_vct2` to `num_vct`.
- b. Divide `num_vct` by `num_vct2`.
- c. Multiply `num_vct2` by `num_vct`.
- d. Subtract `num_vct` from `num_vct2`

1.1.4 Exercise 1.4

The number of elements in a vector is referred to as the length of the vector. With small vectors such as those you have already created, you can simply count these. Alternatively, the `length()` function can be used.

Use the length function to obtain the length of the following objects: `num_vct`, `char_vct`, `num_vct2`.

1.1.5 Exercise 1.5

Create a vector object that contains the following logical and numeric values: TRUE, 6, FALSE, 10, FALSE. Name it `new_vct` and print the output.

Does the output match the contents you entered when you created the vector? Why?

1.2 Importing and Exporting Data

Throughout the course units of this programme, you will utilise real datasets to develop your data analysis and interpretation skills.

For this exercise, you will navigate to the UK Data Service (UKDS) website and acquire the *British Social Attitudes Survey (2019), Poverty and Welfare: Open Access Teaching Dataset*.

1.2.1 Exercise 2.1

Navigate to the British Social Attitudes Survey (2019) webpage on the UK Data Service website.

Scroll down to the end of the webpage and download the dataset in both SPSS and STATA formats in your R project working directory.

To access the SPSS data file, unzip folder and open the folders in the following sequence: UKDA-8850-spss > spss > spss25. Follow the same approach for accessing the STATA file. Place these two files in your root R project working directory or within a sub-folder in this directory.

- a. Import the dataset in SPSS format; name this object `bsas_spss`.
- b. Import the dataset in STATA format; name this object `bsas_stata`.

1.2.2 Exercise 2.2

- a. How many observations and variables do the `bsas_spss` and `bsas_stata` objects have? Are these identical for both objects?
- b. Use the view function to explore these data objects.

1.2.3 Exercise 2.3

Export **both** `bsas_spss` and `bsas_stata` data objects in `.RData` format.

1.2.4 Exercise 2.4

Export `bsas_spss` as a **single** object in `.RDS` format.

1.3 Exercise 2.5

Export the `bsas_stata` object as a `.csv` file.

1.4 Exercise 2.6

- a. Import the `.csv` file you have created in 2.5 and name this object `bsas_stata2`. View the contents of the `bsas_stata2` and `bsas_stata` objects. What are the differences between the two, if any? Why?
- b. Import the file created in 2.4 and name this object `bsas_spss2`.
- c. Load the file created in 2.3.

1.5 Bonus Task: Importing Data from Github Repositories

Many different types of data can be imported in R using either base R functions or functions from packages. However, R is not limited to importing 'hard copy'

files from your machine but also supports direct import of data files located on websites for example.

A Github Repository is a cloud-based, online platform that allows programming users to openly share research projects, associated documentation, data files, and comments with other users and/or the public. Explore the Johns Hopkins Whiting School of Engineering COVID-19 Github Repository.

1. Open the `csse_covid_19_data` folder found at the beginning of the page.
2. Then open the `csse_covid_19_time_series` folder.
3. Click on `time_series_covid19_confirmed_global.csv` link.
4. Click on View Raw.
5. The webpage will now show you a long series of numbers. This is the ‘data file’ that you need to import. Note that this file is a `.csv` file. The same base R function you have previously used will also work in this case.

To import the data file, create a new object called `covid_johnshop`. Within the base R `.csv` function, paste the full data file web link. Do not forget to enclose this link with quotation marks.

View the contents of this object. How many observations and variables does this object have? What class is this object?

Chapter 2

The Structure of Data Objects

Chapter 3

Manipulating and Tidying Data

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 4

Reproducible Workflows with R Markdown

Chapter 5

Data Visualisation

Chapter 6

Working with Special Data Types: Text Data

Chapter 7

Programming with R

Chapter 8

Practising Data Cleaning and Visualisation: Formative Peer-Review Case Study