PY1617: Foundations of R for Statistics and Data Science

Component Notebook

Dr Abigail Page & Dr James Winters

Contents

1 Introduction to R

Welcome to the R component of **PY1617 Employability: Foundations of R for Statistics and Data Science**

This notebook brings together all material for the R component of the module, spanning ten weeks of teaching. Teaching is shared across the term:

1) Weeks 1–5 are delivered by Dr Abigail Page and focus on core R skills, data handling and visualisation

2) Weeks 7–10 are delivered by Dr James Winters and focus on programming concepts and simulation

By the end of the module, students should feel confident reading, writing and running R code, and understand how these skills support statistical analysis and research.

In this first week, the focus is on **orientation**: understanding what R is, why we are using it, how it fits into data science and statistics and getting everything set up correctly on your computer.

1.1 Overview of the R Component

This component runs across **ten teaching weeks**, starting from the very basics of using R and progressing to writing functions and simulating data by the end of the term.

Teaching is shared across the module: - **Weeks 1–5** are taught by **Dr Abigail Page** and focus on foundational R skills, data handling, and visualisation - **Weeks 7–10** are taught by **Dr James Winters** and focus on programming concepts and simulation

By the end of the component, you will be able to use R confidently as a tool for working with data in academic, research, and workplace contexts.

1.2 Aims of the Component

The R component aims to develop **foundational skills in R** that underpin data analysis, statistics, and data science.

Specifically, the component aims to: - Build confidence in using R and RStudio - Develop good habits for writing clear, reproducible code - Introduce core data workflows used in statistics and research - Prepare you for later statistics modules in Years 1 and 2 - Develop digital and data literacy skills valued in the workplace

---

1.3 Learning Objectives

By the end of this component, you should be able to:

- Use **R and RStudio** confidently, including scripts, projects, and packages
- Understand and work with **data types, objects, and data frames** in R
- Apply a **tidy data workflow** to clean, transform, and reshape data
- Create and label **new and recoded variables**
- **Summarise and visualise data** effectively using ggplot2
- Use basic **programming concepts** (loops and functions) to automate tasks
- Write **clear, reproducible code** suitable for academic and workplace contexts
- Use these skills to **simulate data**

You are not expected to master all of these immediately — they are goals for the **end of the module**.

1.4 What Is Data Science?

Data science is the process of turning data into understanding and insight.

This typically includes: - Managing data
- Cleaning and preparing data
- Exploring and visualising data
- Modelling and interpreting results
- Communicating findings clearly

Data science is best thought of as a **workflow**, rather than a single technique.

1.4.1 The data science workflow

Throughout this module, we will refer to the **data science cycle**, which describes how data are: - Imported
- Prepared and transformed
- Explored and visualised
- Modelled
- Communicated

This R component focuses primarily on the **foundational stages of this cycle** — the skills that everything else depends on.

1.5 What Is R?

R is a programming language designed specifically for **data analysis, statistics, and graphics**.
It was first developed in **1996** and is now widely used in academia, research, and industry.

R is: - A **command-driven** language - Highly flexible and extensible - Widely used for statistical analysis and data visualisation

You may have encountered other statistical software such as **SPSS, Stata, or MATLAB**. R differs from these in that it is: - Script-based rather than menu-driven - Highly customisable - Open source and free to use

---

1.6 What Is RStudio?

RStudio is the software environment we will use to work with R.

You can think of: - **R** as the engine
- **RStudio** as the dashboard

RStudio provides: - A script editor - A console for running code - Tools for viewing data, plots, and files - Integrated help and documentation

1.6.1 Why we use RStudio

RStudio is widely used because: - It is **free** and open source - There is a large and active **R community** - Thousands of **packages** extend R's functionality - Solutions to common problems are easy to find online - It produces **high-quality, publication-ready graphics**

1.7 Challenges of Learning R

Learning R can feel challenging at first, especially if you do not have a programming background.

Common difficulties include: - A steep initial learning curve - Small errors in code causing unexpected problems - Not knowing how to phrase questions when searching for help

This is normal.

A useful rule of thumb: > Knowing **what to search for** is a large part of working effectively in R.

Over time, you will build familiarity with common patterns and solutions.

1.8 Installing R and RStudio

Before you can start using R, you need to install **both R and RStudio**.

1.8.1 Installing R

1. Go to: https://cran.r-project.org

2. Select your operating system (Windows, macOS, or Linux)

3. Download and install the **latest version** of R

Always install R **before** installing RStudio.

1.8.2 Installing RStudio

1. Go to: https://www.rstudio.com/products/rstudio/download/

2. Download **RStudio Desktop (free version)**

3. Install the latest version for your operating system

Once installed, you will **open RStudio**, not R itself.
Opening RStudio automatically starts R in the background.

1.9 First Steps in RStudio

When you open RStudio for the first time, you will see several panes: - The **console** (where code runs) - The **script editor** (where code is written and saved) - The **environment** (where objects are stored) - Panels for plots, files, and help

You should always work using **R scripts**, not by typing everything directly into the console.

1.9.1 Opening an R script

In RStudio: - Go to **File** → **New File** → **R Script**

This creates a script where you can write and save your code.

1.10 Basic R Expressions

R uses the **assignment operator** `<-` to store values.

Example:

```
x <- 10
x
```

```
## [1] 10
```

2 Workflow in R

2.1 Files, folders, and projects

2.2 Working directories

2.3 R scripts and packages

3 Data in R

3.1 Data types and objects

3.2 Vectors and data frames

3.3 Functions in R

4 Data Manipulation in R

4.1 The tidyverse

4.2 Recoding and transforming variables

4.3 Missing data (NAs)

4.4 Wide and long data

5 Data Visualisation in R

5.1 Summarising data

5.2 Introduction to ggplot2

5.3 Interpreting plots

6 Conditional Programming

6.1 Logical values

6.2 if and else statements

7 Pipes and Loops

7.1 Pipes (%>%)

7.2 for loops

7.3 Repeating operations

8 Writing Functions from scratch

8.1 Why write functions?

8.2 Function structure

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.