PSYNETS: R Essentials

The PSYNETS Team

2024-11-08

Contents

Chapter 1

Introduction

This book is designed for participants of the PSYNETS workshop to help them get acquainted with the basic functionalities of R and the R packages used throughout the sessions. It has been compiled based on the most frequently asked questions and challenges that participants have encountered during previous workshops. While not a comprehensive guide to R for network psychometrics, it serves as a quick reference to help you get started with R and the specific packages used in the workshop. The primary focus is on installing packages, loading data from various formats, and understanding the unique features of each packages covered in the course. Whenever you have a technical question about R, we recommend consulting this book first.

1.1 Structure of the book

This book is structured as follows...

Chapter 2

Introduction to R for Absolute Beginners

Welcome to R! This guide is designed to help absolute beginners get started with R, a powerful tool for statistical computing and data visualization. R can seem overwhelming at first, but this introduction will walk you through the basics, step by step. If you are already familiar with R, feel free to skip this section.

2.1 What is R?

R is a programming language and software environment specifically designed for statistical analysis and data visualization. It is widely used by data scientists, statisticians, and researchers.

2.2 Getting Started with R

2.2.1 Installing R and RStudio

To use R, you will need two things: 1. **R**: Download and install it from CRAN. 2. **RStudio**: A user-friendly interface for R, which you can download from RStudio's website.

2.2.2 RStudio Interface

When you open RStudio, you'll see four main panels: - **Source/Script Editor**: Write and save your R code here. - **Console**: Run code directly and

see the output. - **Environment/History**: See your data and objects. - **Files/Plots/Packages/Help**: Access files, view plots, manage packages, and get help. you can easily load data sets using this option.

2.3 Basic Syntax in R

2.3.1 Running Code

You can run code directly in the console or write it in the script editor and press `Ctrl + Enter` (or `Cmd + Enter` on a Mac) to run it. We always suggest you do the later, that way your code is saved.

2.3.2 Basic Calculations

At its simplest, R can be used as a calculator. For example:

```
1 + 1
5 * 3
10 / 2
2^3
```

2.3.3 Variables

You can store values in **variables** or as they sometimes called **objects**. A variable is like a box that holds data.

```
x <- 5
y <- 10
sum <- x + y
sum
```

2.3.4 Data Types

R has several basic data types:

- Numeric: Numbers like 42 or 3.14.
- Character: Text or strings like "Hello".
- Logical: `TRUE` or `FALSE`.

```r
num_var <- 42   # numeric
char_var <- "Hello, R!"  # string
log_var <- TRUE # logical
```

2.3.5 Functions

A function is a piece of code that performs a specific task. For example, `sqrt()` calculates the square root:

```r
sqrt(16)
```

There are many in-built functions in R, and you can also create your own functions. And of course, as you will see in the rest of the chapters, many of the functionalities you will use come from packages, which can be installed and loaded follows:

```r
install.packages("package_name")
library(package_name)
```

2.4 Working with Data

2.4.1 Vectors

A vector is a sequence of data elements of the same basic type. You can create a vector using the `c()` function:

```r
my_vector <- c(1, 2, 3, 4, 5)
my_vector
```

2.4.2 Data Frames

A data frame is a table-like data structure (i.e., a spreadsheet) that stores data. You can create a data frame using the `data.frame()` function:

```r
my_data <- data.frame(
  Name = c("A", "B", "C"),
  Age = c(25, 30, 35),
  Height = c(175, 182, 168)
)
my_data
```

```
##   Name Age Height
## 1    A  25    175
## 2    B  30    182
## 3    C  35    168
```

2.5 Loading Data into R

R can read data from various file formats, as already mentioned, you can also use the file tab for this. Here are some examples of how this if formally done through the console, for some common file formats. Before we start, make sure to set your working directory to the location where your data is stored. You can set your working directory using the `setwd()` function. Alternatively, you can use the `here` package to set your working directory to the location of your R script. You can also navigate on top of R studio to the session tab and set your working directory from there. Finally you can use the keyboard shortcut `Ctrl + Shift + H` to set your working directory.

2.5.1 Reading a CSV file

```r
# Make sure the 'data.csv' file is in your working directory
my_data <- read.csv("data.csv")
head(my_data)
```

2.5.2 Reading an Excel File

```r
# Install the readxl package if you haven't already
#install.packages("readxl")
library(readxl)

# Reading an Excel file
my_data <- read_excel("data.xlsx", sheet = 1)
head(my_data)
```

2.5.3 Reading Data from R's Built-in Datasets

Many R packages come with built-in datasets that you can use to practice. For example, the `mtcars` dataset is built into R:

```r
# Load the 'mtcars' dataset
data(mtcars)
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

2.6 Doing a statistical analysis

R is a powerful tool for statistical analysis. You can perform a wide range of statistical tests and analyses using R. Here is an example of how to perform a simple t-test.

2.6.1 T-Test

```r
# Subset the data into two groups
auto <- mtcars$mpg[mtcars$am == 0]   # Automatic cars
manual <- mtcars$mpg[mtcars$am == 1]   # Manual cars

# Perform a t-test
t_test_result <- t.test(auto, manual)
t_test_result
```

```
##
##  Welch Two Sample t-test
##
## data:  auto and manual
## t = -3.7671, df = 18.332, p-value = 0.001374
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.280194  -3.209684
## sample estimates:
## mean of x mean of y
##  17.14737  24.39231
```

2.6.2   Visualizing data

R is also great for data visualization. You can create a wide range of plots and graphs using R. Here is an example of how to create a boxlot. R has in built functions for many types of plots, and you can also use the `ggplot2` package for more advanced plots.

```r
boxplot(mpg ~ am, data = mtcars,
        main = "Miles Per Gallon (mpg) by Transmission Type",
        xlab = "Transmission (0 = Automatic, 1 = Manual)",
        ylab = "Miles Per Gallon (mpg)")
```

Miles Per Gallon (mpg) by Transmission Type
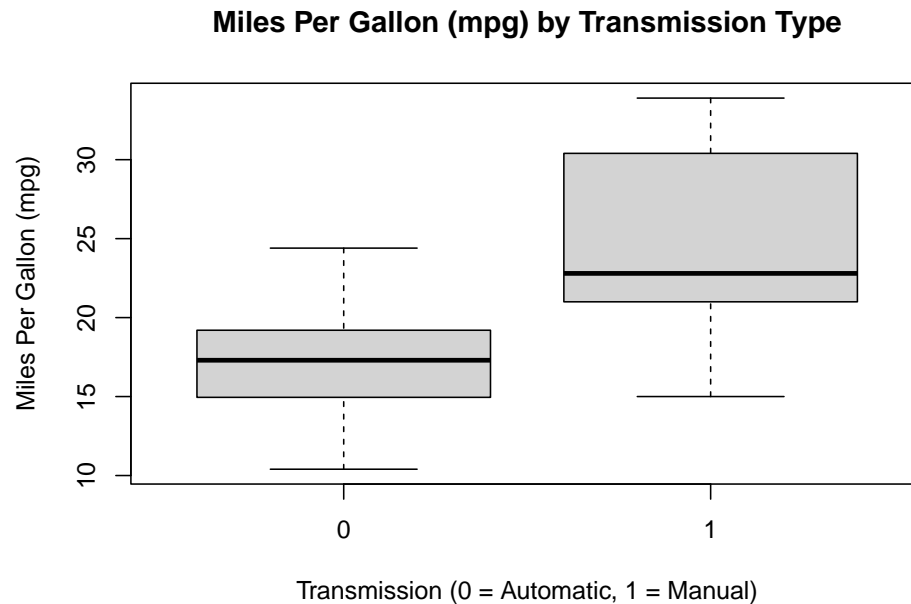
2.7   Additional Resources

This introduction is just the tip of the iceberg. R is a vast and powerful tool with a steep learning curve. Here are some additional resources to help you continue your journey with R:

- R for Data Science: A comprehensive guide to data science with R.
- RStudio Cheatsheets: Handy cheatsheets for various R topics.
- Stack Overflow: A great place to ask questions and find answers about R.
- R Documentation: Official documentation for R packages.

2.8 Conclusion

Congratulations! You've completed the introduction to R. We hope this guide has given you a solid foundation to start exploring R on your own.

Chapter 3

Getting prepared for PSYNETS

Chapter 4

Causality

Chapter 5

Corss-Sectional

Chapter 6

Time Series

Chapter 7

Advanced

Chapter 8

Bayesian

Chapter 9

Q&A