

Tutorial 1: Introduction to Data Visualisation

Dr. Kate Saunders

Table of contents

Learning Objectives	1
Preparation	1
Exercise 1: R Packages	1
Exercise 2: Assign a Variable	2
Exercise 3: Basic Computation	3
Exercise 4: Vectors	4

Learning Objectives

- Get you started using R
- Learn your way around RStudio.
- Become familiar with R, RStudio, RStudio projects and Quarto files.
- Become confident doing basic computation in R.

Preparation

- Install R and RStudio
- Watch/Read the Introductory materials on R provided

Exercise 1: R Packages

R has many standard functions, like `mean`, `sum` and `sqrt`. Other functions are stored in R packages that can be loaded as needed for specific tasks.

One of the common packages is the `tidyverse` it contains lots of useful functions for data wrangling and visualisation. The below code checks if the package is already in your package library. If the package is not there it then uses the `install.packages` function to download it.

```
if(!require(tidyverse)){  
  install.packages("tidyverse")  
  library(tidyverse)  
}
```

You will only need to install a package once.

Whenever you start a new session in RStudio you will need to load the R packages you need.

```
library(tidyverse)
```

Install the `ohwhaley` R package:

```
if(!require(remotes)){  
  install.packages("remotes")  
}
```

```
remotes::install_github("fontikar/ohwhaley")
```

Your turn:

1. Load `ohwahely` package using `library(ohwhaley)`
2. Then run my favourite R function `ohwhaley::say()`

If at any point today you start feeling overwhelmed, enter `ohwhaley::say()` into your console until you feel better. Then remind yourself learning new things is hard and ask for help from a peer or your tutor.

Exercise 2: Assign a Variable

R is more than just as a calculator, you can store things in a **variable**. Here the word variable has a slightly different meaning to the usual statistical meaning.

Let us start with a simple command, type in `a <- 5`. Here we assign the value 5 to the variable `a`.

```
# This is an R chunk. Enter your code here. Text in R chunks needs to be prefaced with a # s  
a <- 5
```

Note that you can use either `<-` or `=` to assign variables.

Your turn:

1. Let try `b = 4`
2. We can store more than just numbers in a variable. Try storing your name. e.g. `name = 'Jane Doe'`.
3. We can use `str(name)`, `str(a)` to check the structure of the variable, it will show you that variable name contains character and variable a contains numeric.
4. R is case sensitive. We create a variable `name` in previous step, try type `NAME`

Exercise 3: Basic Computation

Let continue with: `x <- 5`, `y <- sqrt(16)`, `z <- -3`, and `w <- x + y + z`

```
x <- 5
y <- sqrt(16)
z <- -3
w <- x + y + z
```

`sqrt(16)` is a function for square root of 16, therefore the value for y is 4 To find the value of w, type it in. The answer should be 6.

```
w
```

```
[1] 6
```

R can be used as a calculator, to subtract use `-`, to multiply use `*`, to divide use `/` and to take powers use `^`.

To see all assigned variables:

```
ls() # lists out all the variables
```

```
[1] "a"           "object"      "pandoc_dir"  "quarto_bin_path"
[5] "w"           "x"           "y"           "z"
```

You can also look in the environment tab in the top right hand panel of Rstudio to see what variables have been stored.

Your turn:

1. Try $h = x^2$ and $f = z/y$.

Exercise 4: Vectors

A vector is a list of similar type objects and is a basic data structure in R that can hold values. Previously w had only one value, but we can store multiple values in w by using the command `c()`. Here `c` stands for “combine” or “concatenate”.

For example try:

```
Consumption <- c(50, 40, 25, 0)
Consumption
```

```
[1] 50 40 25 0
```

```
str(Consumption)
```

```
num [1:4] 50 40 25 0
```

To create a vector of characters:

```
Drink <- c('Coke', 'Pepsi', 'Coke', 'Homebrand')
Drink
```

```
[1] "Coke"      "Pepsi"     "Coke"      "Homebrand"
```

```
str(Drink)
```

```
chr [1:4] "Coke" "Pepsi" "Coke" "Homebrand"
```

Sometimes when we apply a function to a vector, we apply the function to each element.

Your turn:

1. Find the log consumption:

```
logcons <- log(Consumption)
str(logcons)
```

2. Let's create two variables: `x1` contains "5,6,3,6,4" `y1` contains "3,7,5,1,1"
3. Compute the mean for `x1` and `y1` by using `mean()`
4. We can find the variance of the final pair using the `var` function.
5. We can find the correlations between `x` and `y` using the `cor` function.

Material developed by Dr. Kate Saunders and past lecturers of ETX2250/ETF5922

© Copyright 2024 Monash University