R Self-Study Guide: Data Structures, Functions, and Visualization

Samuel Irungu Kigotho

2025-10-22

Introduction

This R script is a tutorial on fundamental R concepts: vectors, matrices, arrays, data frames, lists, functions, loops, apply-family functions, and data visualization with ggplot2

Vectors

- A vector is simply a list of items that are of the same type.
- *c()* combines values into a vector (the simplest R data structure).
- *val* is a numeric vector with 6 elements.

```
val = c(1, 2, 3, 4, 5, 6)
val
```

```
## [1] 1 2 3 4 5 6
```

```
class(val)
```

```
## [1] "numeric"
```

- Mixing numbers and text converts all elements into character type.
- R uses coercion — all elements in a vector must be the same type.

```
x = c(1, 2.0, 3.0, 4, 5, "Hello", "OK")
x
```

```
## [1] "1"     "2"     "3"     "4"     "5"    
## [6] "Hello" "OK"
```

```
class(x)
```

```
## [1] "character"
```

- Naming elements inside a vector — useful for labeling values.
- Access by label: temperature[“morning”].

```
temperature = c(morning = 20, before_noon = 23, after_noon = 25, evening = 22, night = 18)
temperature["morning"]
```

```
## morning 
##      20
```

Basic Operations with Vectors

- Vectors of equal length can be added element-wise.

```
jan_price = c(10, 20, 30)
june_price = c(20, 25, 33)
mar_price = jan_price +june_price
mar_price
```

```
## [1] 30 45 63
```

Character Vectors

```
items = c("potato", "rice", "oil")
class(items)
```

```
## [1] "character"
```

Matrix

A matrix is a two dimensional data set with columns and rows. A column is a vertical representation of data, while a row is a horizontal representation of data. A matrix can be created with the *matrix()* function. Specify the *nrow* and *ncol* parameters to get the amount of rows and columns: - Lets Combines all prices into one matrix. - *nrow* = 3 → matrix will have 3 rows. - *byrow* = TRUE fills the matrix row by row instead of column by column.

```
all_prices = matrix(c(jan_price, mar_price, june_price), nrow = 3)
all_prices
```

```
##      [,1] [,2] [,3]
## [1,]   10   30   20
## [2,]   20   45   25
## [3,]   30   63   33
```

```
class(all_prices)
```

```
## [1] "matrix" "array"
```

```
all_prices2 = matrix(c(jan_price, mar_price, june_price), nrow = 3, byrow = TRUE)
all_prices2
```

```
##      [,1] [,2] [,3]
## [1,]   10   20   30
## [2,]   30   45   63
## [3,]   20   25   33
```

Array

An array is a data structure that can hold multi-dimensional data. In R, the array is objects that can hold two or more than two-dimensional data - Lets combine 6 vectors into a 3×3×2 array: - 3 rows → items - 3 columns → months - 2 layers → years (2018 and 2017)

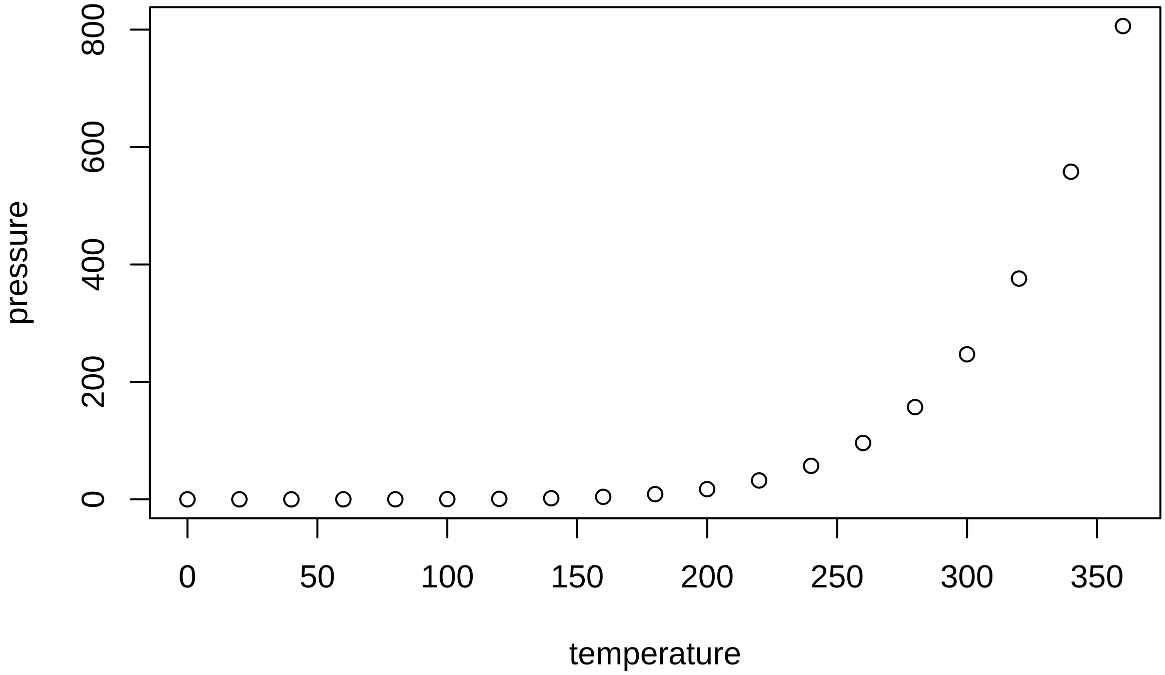
```
# Create six vectors
jan_2018 = c(10, 11, 20)
mar_2018 = c(20, 22, 25)
june_2018 = c(30, 33, 33)
jan_2017 = c(10, 10, 17)
mar_2017 = c(18, 23, 21)
june_2017 = c(25, 31, 35)
```

```
combined = array(c(jan_2018, mar_2018, june_2018, jan_2017, mar_2017, june_2017), dim = c(3, 3, 2))
combined
```

```
## , , 1
##
##      [,1] [,2] [,3]
## [1,]   10   20   30
## [2,]   11   22   33
## [3,]   20   25   33
##
## , , 2
##
##      [,1] [,2] [,3]
## [1,]   10   18   25
## [2,]   10   23   31
## [3,]   17   21   35
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.