R Manual

Amanda Araujo

Abstract

A compilation of basic comands and structures of R programming language. The *4dummies* guide that I needed in order to keep my sanity, instead of searching for the same questions for the thousandth time. We ought to be our best collaborator, and I guess the future me would thank me. It is a ongoing process.

1 Useful R build-in functions

`length(x)`: returns the size of a vector
`typeof(x)`: returns the type of object

1.1 Help! I need somebody help

It isn't exactly a function, but R* offers the Help for all the functions it has. It's just run the comand: `?function_name`.

*IDEs (an integrated development environment) as RStudio will open the function's page, that contains all its information and how to use it: arguments, what type of object it supports and more.

2 Vectors

Atention: In R the index count starts at 1.

2.1 Adding values in a vector

Function `append(vector, new value)`

```
#Example appending values in a vector
abacaxi <- c(1, 2, 3)
abacaxi <- append(abacaxi, 4)
print(abacaxi)

[1] 1 2 3 4
```

3 Lists

A list in R is almost a 'fruit salad' object: it holds different types of objects, you can store whatever you want: integers, characteres, vectors, strings, etc.

List is a data structure having components of mixed data types.

3.1 List of vectors: indexes

When working with R, there is a particular way of accessing an element inside a list of vectors.

```
#Example accessing element of a vector inside a list
mi <- c(1, 2, 3)
fa <- c(4, 5, 6)
sol <- list(mi, fa)

> sol
[[1]]
[1] 1 2 3

[[2]]
[1] 4 5 6

> sol[[1]]
[1] 1 2 3

> sol[[2]][1]
[1] 4
```

3.2 Plot data from lists

When trying to plot data directly from a list, one can get the following error:

```
Error in xy.coords(x, y, xlabel, ylabel, log) :
(list) object cannot be coerced to type 'double'.
```

Solutions: *for-loop*, *unlist*.

The best way to do it is to avoid the for-loop and unlist the lists in order to plot them. This is a way using unlist:

```
plot(unlist(l),unlist(k))
```

4 Functions

4.1 Multiple Returns

The `return()` function can return only a single object. If we want to return multiple values in R, we can use a list (or other objects) and return it.

```
#Example return multiple values
multireturn <- function() {
mylist <- list("color" = "red", "size" = 20, "shape" = "round")
return(mylist) }
```

5 Loops

5.1 For

```
    #For structure
    for (variable in vector) {
        <code>
    }
```

```
    range <- 5:10                                   #Create numeric range
    range                                           # Print range
   # 5  6  7  8  9 10
```

```
    for (a in 1:3) {
        print('oi')
    }

    [1] "oi"
    [1] "oi"
    [1] "oi"
```

6 Plotting

6.1 Color palletes

Here is where relies the fun!

One of the beauties of the art of plotting is the enormous amount of palletes available.

6.1.1 Wes Anderson

Wes Anderson R package with palletes inspired from his cinematical masterpieces (1).

```
    #Wes Anderson inspired palletes
    install.packages("wesanderson")
    library("wesanderson")

    names(wes_palettes)

    cols <- wes_palette("GrandBudapest2", 10, type = "continuous")
    cols2 <- wes_palette("Darjeeling1", 10, type = "continuous")
    cols3 <- wes_palette("Zissou1", 10, type = "continuous")
```

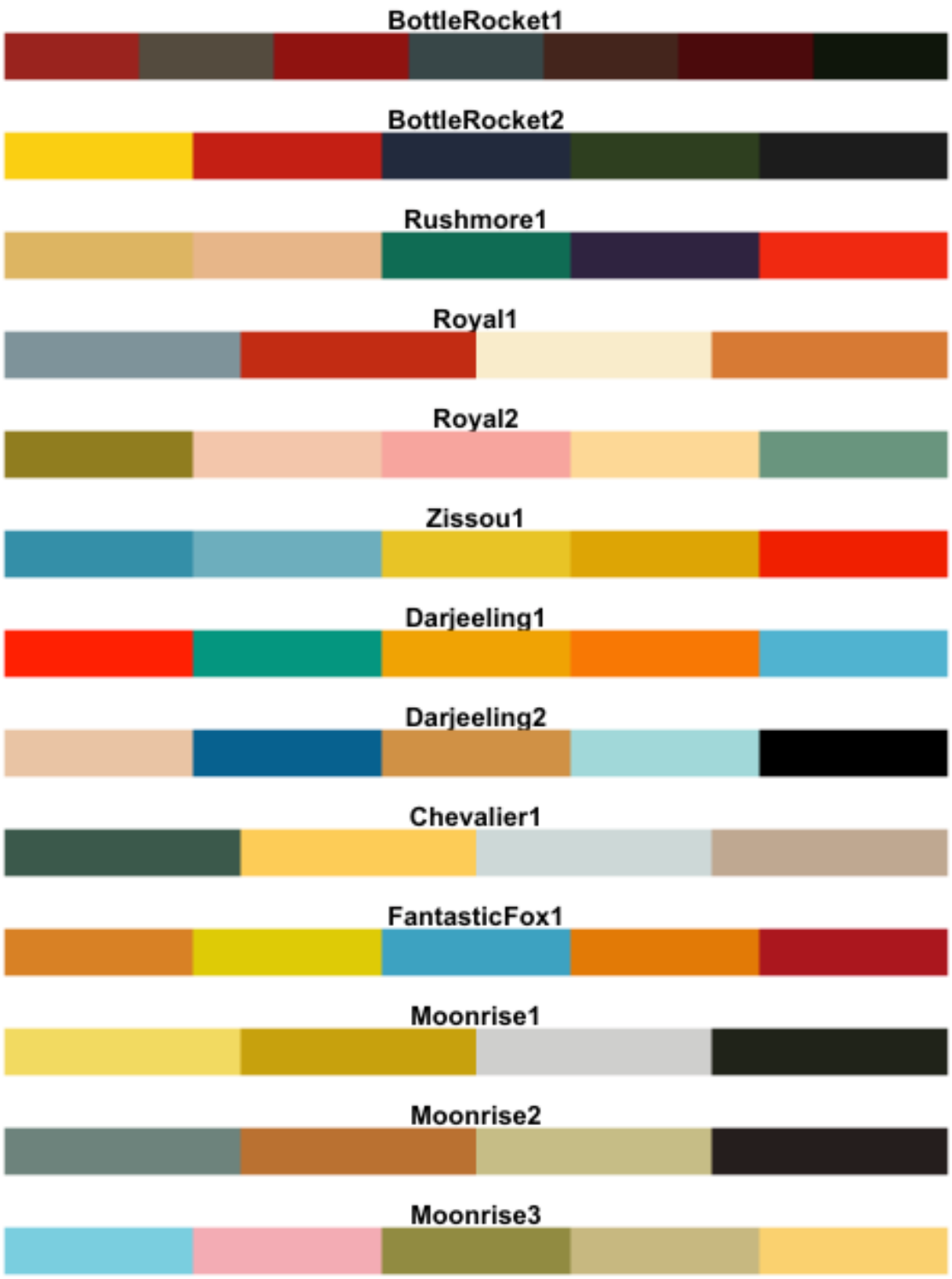

Figure 1: Wes Anderson R package `library("wesanderson")`