Brackets [ ] Braces ( ) & Parentheses { }

Willis Wilson
10/30/2017

Introduction

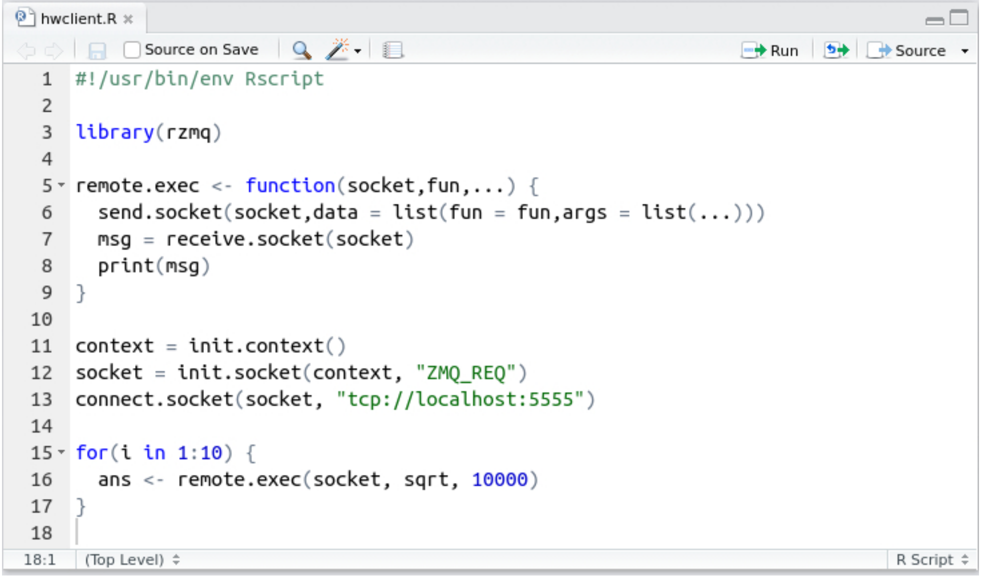
R is amazingly powerful and can be used as a tool for a variety of tasks including but not limited to: data analysis, data visualization, coding, and data manipulation. The key to performing these tasks in R is not just having good data to analyze, but also being able to look deep into the data to access solutions and predictions related the questions you are exploring. This requires a significant amount of knowledge on how to perform functions and access the data you are interested in using R. This is done using the various packages and functions available within the R software package. These functions are what gives R it's computing power and makes it such am amazing software and tool. Essential to every package and function are braces, brackets, and parentheses. You will use at least one and in many cases, all of these to execute commands and perform data analysis and manipulation within the R software package. This article is focused on doing an in-depth review on braces, brackets, and parentheses and exemplifying how they can be leveraged to get the most from the R Software Package and accomplish your computing goals.

My introduction to R was full of challenges. Having no significant data science experience, this was my first bit of exposure to data analysis software. While the general idea of computing in R made sense, in my early R encounters I made many small errors and experienced difficulties using R to access the information within the data that I was interested in. After countless hours of practice, YouTube videos, assignments, tears, and patience, I finally began to be able to explore the true power of R and effectively perform data analysis using the R software package. Looking back, it is clear that my early struggles with R were rooted in my lack of understanding of how to use braces {}, brackets [], and parentheses () to perform data exploration and analysis within the R software package. The first of the three you are likely to encounter are parenthesis, so I will begin by discussing parenthesis, then proceed to discuss braces and brackets and how they all work together. The goal of this article is to provide a simple and easy to follow guide through the uses of braces {}, brackets [], and parentheses () so that you, the reader, can avoid or escape the many struggles that arise in many users' introduction to the R software package.

Parentheses ( )

Parentheses are a primitive function in R and can be used to perform a variety of tasks within R. The main uses for parenthesis within R is to house function inputs, define order of operation, indexing into an array, and grouping. Something unique about calling a function using parenthesis is that the output value is automatically printed. The main thing to remember is that when calling a function, use parenthesis.

Parenthesis Example



```
#!/usr/bin/env Rscript

library(rzmq)

remote.exec <- function(socket,fun,...) {
  send.socket(socket,data = list(fun = fun,args = list(...)))
  msg = receive.socket(socket)
  print(msg)
}

context = init.context()
socket = init.socket(context, "ZMQ_REQ")
connect.socket(socket, "tcp://localhost:5555")

for(i in 1:10) {
  ans <- remote.exec(socket, sqrt, 10000)
}

```

The above example gives many example of how parenthesis are used as a part of functions within R

Brackets [ ]

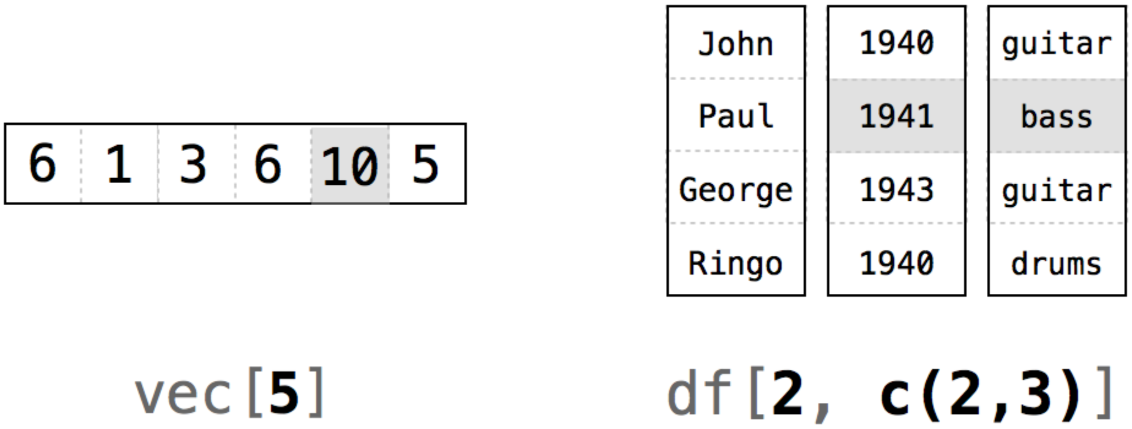
Bracket notation is one of the keys to R. The key to bracket notation is, you guessed it, BRACKETS! Within R, the square brackets are used to subset vectors and data frames, create an array or matrix, delete elements, and/or group function outputs. The two leading uses of bracket notation is to subset variables and observations. While there are multiple ways to subset a list in R, but brackets and double brackets are the most common.

Single vs. Double Brackets	Use
The single bracket [x]	This extracts a sub-list from the data frame
The double brackets [[x]]	This extracts & removes a level of hierarchy form the data frame ***(this can also be done using the $ operator)***

Bracket notation are used to subset variables within R. Bracket notation allows you to access the values within the data frame. Data Frames within R resemble spreadsheets. They are comprised of rows and columns that possess the values you are interested in. Each value within a data frame has an exact location that is found by using the values specific row and column. Brackets lets you select, or subset, data from a vector, matrix, array, list, or data frame. Bur remember, the results you get using sub-setting depend, at least partially, on the data type.

Sub-setting can be used in a variety of ways. When you only want to access subset variables (or columns) we use the second index and leave the first index blank. By leaving an index blank you indicate that you want to keep all the elements in that dimension. If you leave the first index blank, you are telling R that you want all the row values. To call consecutive columns or rows you have two options, you can use either a colon as a separator between the beginning and end values or you can use the c() function.

Brackets Example



```
vec[5]
```

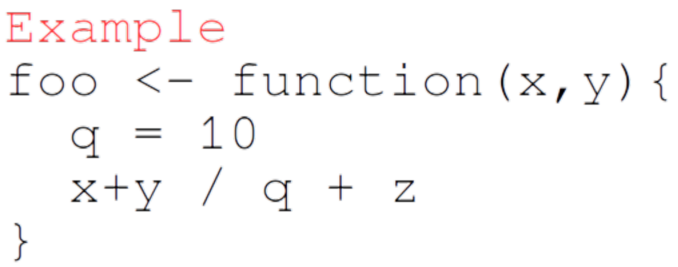
```
df[2, c(2,3)]
```

In the above example *vec[5]* returns 10. Because the value 10 is the fifth element in the vector. In above data frame *df[2, c(2,3)]* give the second and third elements of the df data frame, *1942* and *base*.

Braces

Curly brackets are a primitive function within R and can be used to denote a block of code within a function, create a cell array, or get content from a cell array. Plainly put, they are used to evaluate a series of expressions. The leading use of curly braces is to define a function. When calling a function using curly braces the value of last expression is printed. Additionally, Curly braces can be used very similarly to the way parentheses are used. They are frequently used to enclose an expression that already uses parentheses. I'm many instances this could also be accomplished using an additional set of parentheses.

Braces Example



```
Example
foo <- function(x,y){
   q = 10
   x+y / q + z
}
```

In the above example, you see the parentheses used to house the arguments of the function, curly braces used to house the actual body of the function.

Conclusion

The goal of this post is to increase you understanding of the Braces { }, Brackets [n], and Parentheses (n). They are essential when performing data exploration and analysis within the R software package. Combining your new-found knowledge with practice and the many packages, commands, and functions within R you will be able to take advantage of the seemingly infinite capabilities of R.

References

1. Markdown Tutorial
2. DataCamp.com "Writing functions in R"
3. Parentheses and Braces
4. Difference Between Square Brackets and Curly Brackets in R
5. R in a Nutshell, 2nd Edition
6. R Markdown CheatSheet
7. "Data Vectors in R"
8. "Lists in R"
9. SUBSETTING DATA | R LEARNING MODULES
10. "R accessors explained"
11. "R Subsetting, Extracting and Bracket Notation"