Bite Size R

Data Structures

Keith Hurley

April 12, 2019

Data Structures

R has a relatively small number of data types. However, these simple types can be more useful when aggregated into more complex data structures.

List

The most basic data structure in R is the list. A list is simple a group of values. The values do NOT have to be of the same data types. We create lists with the “list” function.

```r
l<-list(TRUE, 1L, 2.34, "abc")
l
```

```
## [[1]]
## [1] TRUE
##
## [[2]]
## [1] 1
##
## [[3]]
## [1] 2.34
##
## [[4]]
## [1] "abc"
```

Lists can also store other complext data structures, like more lists!

```r
l2<-list(FALSE, 100, l)
l2
```

```
## [[1]]
## [1] FALSE
##
## [[2]]
## [1] 100
##
## [[3]]
## [[3]][[1]]
## [1] TRUE
##
## [[3]][[2]]
## [1] 1
##
## [[3]][[3]]
## [1] 2.34
##
## [[3]][[4]]
## [1] "abc"
```

Notice the subscripts, there's items 1, 2, and 3. Then item 3 has subscripts of 1, 2, 3, and 4. As you can already guess, we can get pretty complex with our data structures using only lists!

Vectors

Vectors are the most common data structure in R and resemble a list. However, each item in a vector must have the same simple data type. Vectors are created using the "c()" function.

```
#numeric vector
v_numeric<-c(1.5, 2, 75, 100.1)
v_numeric
```

```
## [1]   1.5   2.0  75.0 100.1
```

```
#character vector
v_character<-c("Charlie", "Sue", "Nathan", "Betty")
v_character
```

```
## [1] "Charlie" "Sue"     "Nathan"  "Betty"
```

Notice how the character strings are enclosed with double quotation marks.

Matrices and Arrays

After vectors we move into more complex, multi-dimensional structures. Matrices (matrix) are two dimensional structures (think in terms of a grid) where all data items must have the same data type and each dimension must of the same length (i.e. each row must have 5 columns). An array is similar to a matrix, but may contain more than 2 dimensions. Some advanced calculations may depend on matrices, arrays, other complex, multi-dimensional structures, but I have rarely found the need to use them in my day to day work and we won't be talking about them in bite-sized R.

Dataframe

The bread and butter data structure in R is the dataframe. It is also the central component in tidy data philosophy that we'll discuss in a different snack. Think of a dataframe much like a table in a conventional database. There are rows of columns. Each column represests a variable and in R is represented by a vector of same data type values. Each row is a record and in r represents a collection of single values from each column.

Let's create a dataframe using the "data.frame" function.

```
myDataFrame<-data.frame(
  Name=c("Bluegill", "Catfish", "Minnow", "Tuna"),
  Size=c(10,24,1,88),
  IsEdible=c(TRUE, TRUE, FALSE, TRUE)
)

myDataFrame
```

```
##       Name Size IsEdible
## 1 Bluegill   10     TRUE
## 2  Catfish   24     TRUE
## 3   Minnow    1    FALSE
## 4     Tuna   88     TRUE
```

As you can see, in a dataframe we also store column names in addition to the data itself. In fact, dataframes will also store row names, although I seldom find that useful and generally let R use default row names (i.e. row numbers).

We can look at the names of the columns in our dataframe with the “names” function.

```
names(myDataFrame)
```

```
## [1] "Name"     "Size"     "IsEdible"
```

We can also use the “names” function to rename our columns by assigning it a vector of new names.

```
names(myDataFrame) <- c("FishName", "FishSize", "IsFishEdible")

myDataFrame
```

```
##   FishName FishSize IsFishEdible
## 1 Bluegill       10         TRUE
## 2  Catfish       24         TRUE
## 3   Minnow        1        FALSE
## 4     Tuna       88         TRUE
```

We can also get data from a specific column by using the $ operator.

```
myDataFrame$FishName
```

```
## [1] "Bluegill" "Catfish"  "Minnow"   "Tuna"
```

Wrap Up

Dataframes (and thereby necessity also vectors) are going to be the main data structure that we use in bite size R. When we start loading and using the tidyverse, you may find that our dataframes start being referred to as tibbles. At this point in your R education, just consider tibbles to be a different type of dataframe that allows us to look at large dataframes more efficiently in R. For the purposes of R, dataframes and tibbles will be used interchagably.

You are now armed with a new understanding of how R takes a small number of simple data types and allows us to construct complex data structures. Our time being introduced to how data is handled in R is coming to a close and we are about to start getting to the fun stuff…doing things with data! Happy snacking and see you next time.