Introduction to R

Lists, Arrays, and Dataframes

Sumit Mishra

Krea University | WSDS002

Contents

Lists

The most flexible method to store information in `R`. Consider the following example. I want to store the following variables- my name (`my_name`), courses that I teach in the second trimester (`my_courses`), and the days of the week my classes are scheduled (`class_days`)- in a list. We will do this in two steps (but you can always achieve the same in one shot). The first step will be to create the three vectors, and then roll them into a list (`my_list`).

```
# create three vectors
my_name <- "Sumit"
my_courses <- c("Statistics with R", "Macroeconomics")
class_days <- c(1, 3)
# create the list
my_list <- list(my_name, my_courses, class_days)
str(my_list)
```

```
## List of 3
##  $ : chr "Sumit"
##  $ : chr [1:2] "Statistics with R" "Macroeconomics"
##  $ : num [1:2] 1 3
```

```
print(my_list)
```

```
## [[1]]
## [1] "Sumit"
##
## [[2]]
## [1] "Statistics with R" "Macroeconomics"
##
## [[3]]
## [1] 1 3
```

Let's familiarize ourselves with our new friend `list`. There are objects within the list `my_list` which can be called by using index within double square parenthesis `[[X]]`. So, for instance,if you want to pull the vector `my_course`, you will write `my_list[[2]]`. Within each vector, there are objects, and these can also be easily gleaned using square parenthesis `[]`. For example, you want to extract the first object of the second vector of the list. Here's how you can achieve this. You should type `my_list[[2]][1]`.

```
## [1] "Statistics with R" "Macroeconomics"
```

```
## [1] "Statistics with R"
```

Matrices

We will now build the same dataset (with information on course and class day) into a matrix form. Any matrix has m rows and n columns. Let's visualize how our matrix will look like: two rows with names of the courses and the days.

```
info_t2 <- c("SwR", "MAC", 1,3)
mat_t2 <- matrix(info_t2, nrow = 2, ncol = 2)
print(mat_t2)
```

```
##      [,1]  [,2]
## [1,] "SwR" "1"
## [2,] "MAC" "3"
```

The structure of a matrix is as follows: `Matrix Name[Row Index, Column Index]`

- Any column of a matrix can be called by typing `Matrix Name[, Column Index]`
- Any row of a matrix can be gleaned by writing `Matrix Name[Row Index, ]`
- Please note that the index can be a number or a set of integers. Each object within a matrix can also have a name (just like a vector). All you need to do is to supply an argument called `dimnames =` into `matrix()`. You need to define a list that contains the names.

```
mat_t2 <- matrix(info_t2, nrow = 2, ncol = 2,
                 dimnames = list(c("R1", "R2"),c("Course", "Day")))
print(mat_t2)
```

```
##    Course Day
## R1 "SwR"  "1"
## R2 "MAC"  "3"
```

Arrays

Arrays are generalized forms of matrices. Consider that I want to store information for two terms - term 2 and term 5- onto an object.

```
info_t5 <- c("IEM", "AEC", 2,4)
mat_t5 <- matrix(info_t5, nrow=2, ncol=2)
```

We can use the `array` function.

```
ar_comb <- array(c(mat_t2, mat_t5), dim = c(2,2,2))
dimnames(ar_comb)[[3]] <- c("Term II", "Term V")
print(ar_comb)
```

```
## , , Term II
##
##      [,1]  [,2]
## [1,] "SwR" "1"
## [2,] "MAC" "3"
##
## , , Term V
##
##      [,1]  [,2]
## [1,] "IEM" "2"
## [2,] "AEC" "4"
```

Data Frames

Data frames in `R` are the spreadsheet equivalent objects in R with m rows and n columns. Before we jump into `tidyverse`, let's use vectors to create data frames. As an example, let's construct the Beatles catalog (`beatles.catalog`) using vectors containing the names of the albums (`album`), the year (`year`), and the number of tracks (`num.tracks`). The function that we will invoke is `data.frame()`.

```
album <- c("Please Please Me", "Rubber Soul", "Magical Mystery Tour")
year <-  c(1963, 1965, 1967)
num.tracks <- c(14,14,11)
beatles.catalog <- data.frame(album, year, num.tracks)
str(beatles.catalog)
```

```
## 'data.frame':    3 obs. of  3 variables:
##  $ album     : chr  "Please Please Me" "Rubber Soul" "Magical Mystery Tour"
##  $ year      : num  1963 1965 1967
##  $ num.tracks: num  14 14 11
```

album	year	num.tracks
Please Please Me	1963	14
Rubber Soul	1965	14
Magical Mystery Tour	1967	11

You can see that there are three columns and three rows in a data frame. Calling a row using index is recommended. A few examples:

- `beatles.catalog[2,]` will get you the second row of the dataset.
- `beatles.catlog[2:3,]` returns the second and the third rows.

However, it is recommended that you should use the column name to call a column. We will do this using the Beatles catalog we created. We want to print the names (and the corresponding years) of the albums (let'say).

```
beatles.catalog[,"album"]
```

```
## [1] "Please Please Me"     "Rubber Soul"          "Magical Mystery Tour"
```

```
beatles.catalog[, c("album", "year")]
```

```
##                  album year
## 1     Please Please Me 1963
## 2          Rubber Soul 1965
## 3 Magical Mystery Tour 1967
```

It is worth noting here that data frames are just a special case of lists. You can always create a list, and transform that list into a dataframe, except that the resulting data frame will have columns of equal length. Example: I create a list with the following vectors- `name` (of length 1), `course` (of length 3), and `term` (of length 3), and then convert it into a data frame.

```
myCourseList <- list(name = "Sumit",
                     course = c("ITR", "MAC", "IEM"),
                     term = c("T1", "T2", "T5"))
df <- data.frame(myCourseList)
```

name	course	term
Sumit	ITR	T1
Sumit	MAC	T2
Sumit	IEM	T5

Simple operations on data frames

- `names()`: glance at the column names. You can also use it to set new names for the columns.

```
names(df)
```

```
## [1] "name"   "course" "term"
```

- `dim()`: tells you the dimension of a data frame.

```
dim(df)
```

```
## [1] 3 3
```

- `head()`: prints the first six rows of a data frame.

```
head(mtcars) #mtcars is an example dataset in R
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

- `tail()`: prints the last six rows of a data frame.

```
tail(mtcars)
```

```
##                 mpg cyl  disp  hp drat    wt qsec vs am gear carb
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

- `summary()`: produces the summary statistics for the data frame.

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

- adding a new column can be done using:
  - `Data Frame$New Column ← Shazam`
  - `Data Frame$New Column ← Operation on Old Column`

```
#create number of students enrolled for each of the courses
df$num_stu <- c(50,60,12)
```

name	course	term	num_stu
Sumit	ITR	T1	50
Sumit	MAC	T2	60
Sumit	IEM	T5	12

```
#creating a new column using existing columns
a_df <- data.frame(x = sample(1:100,6), y = sample(1:10,6))
a_df$z <- a_df$x*a_df$y
```

x	y	z
1	3	3
60	4	240
5	9	45
46	8	368
31	10	310
57	5	285

Done for the day

```
## Sorry, this silly GIF is only available in the the HTML version of the notes.
```