Data Structures Exercises (Part-1)

r-bloggers.com/data-structures-exercises-part-1

R Programming has various Data Structures for efficient manipulation of Data.
Following are the list of data structures supported by R.

1. Vectors
2. Lists
3. Matrix
4. Data frame

This exercise helps through various operations of R Data structures.

Exercise 1

Create an atomic vector of Character type, double type, logical type , integer type, complex type and raw type

Exercise 2

Check whether each of the created data structure is of vector type as well as check the class of each of the data.

Exercise 3

Create a List of heterogeneous data, which include numeric, character and logical vectors and prints the lists.

Exercise 4

Create a matrix of 3 rows and 4 columns which stores data from 1 to 12 and arrange the value row wise.

Exercise 5

Create a matrix of 3 rows and 4 columns which stores random data and arrange the matrix column wise

Exercise 6

Create two 2 x 2 matrices with random values and perform all arithmetic operations with those two and print the resultant matrices

Exercise 7

Create a random matrix having values between 1 and 1000 and print the matrix and transpose of the matrix

Exercise 8

Create Data Frames which contain details of 5 employees and display the same.

Exercise 9

Create Data Frames which contain details of 5 employees and display summary of the data

Exercise 10

Create a Data Frame of 5 employees and display the details of First and Last Employee and Names of All employees.

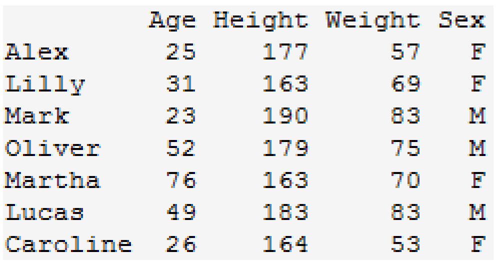
Data frame exercises

In the exercises below we cover the basics of data frames. Before proceeding, first read section 6.3.1 of An Introduction to R, and the help pages for the cbind, dim, str, order and cut functions.

Answers to the exercises are available here.

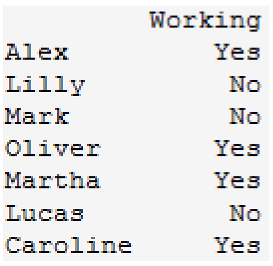
For other parts of this series please follow the tag: dataframes.

Exercise 1
Create the following data frame, afterwards invert Sex for all individuals.



```
         Age Height Weight Sex
Alex      25    177     57   F
Lilly     31    163     69   F
Mark      23    190     83   M
Oliver    52    179     75   M
Martha    76    163     70   F
Lucas     49    183     83   M
Caroline  26    164     53   F
```

Exercise 2
Create this data frame (make sure you import the variable Working as character and not factor).



```
         Working
Alex         Yes
Lilly         No
Mark          No
Oliver       Yes
Martha       Yes
Lucas         No
Caroline     Yes
```

Add this data frame column-wise to the previous one.
a) How many rows and columns does the new data frame have?
b) What class of data is in each column?

Exercise 3
Check what class of data is the (built-in data set) state.center and convert it to data frame.

Exercise 4

Create a simple data frame from 3 vectors. Order the entire data frame by the first column.

Exercise 5
Create a data frame from a matrix of your choice, change the row names so every row says id_i (where i is the row number) and change the column names to variable_i (where i is the column number). I.e., for column 1 it will say variable_1, and for row 2 will say id_2 and so on.

Exercise 6
For this exercise, we’ll use the (built-in) dataset VADeaths.

a) Make sure the object is a data frame, if not change it to a data frame.
b) Create a new variable, named Total, which is the sum of each row.
c) Change the order of the columns so total is the first variable.

Exercise 7
For this exercise we’ll use the (built-in) dataset state.x77.

a) Make sure the object is a data frame, if not change it to a data frame.
b) Find out how many states have an income of less than 4300.
c) Find out which is the state with the highest income.

Exercise 8
With the dataset swiss, create a data frame of only the rows 1, 2, 3, 10, 11, 12 and 13, and only the variables Examination, Education and Infant.Mortality.
a) The infant mortality of Sarine is wrong, it should be a NA, change it.
b) Create a row that will be the total sum of the column, name it Total.
c) Create a new variable that will be the proportion of Examination (Examination / Total)

Exercise 9
Create a data frame with the datasets state.abb, state.area, state.division, state.name, state.region. The row names should be the names of the states.

a) Rename the column names so only the first 3 letters after the full stop appear (e.g. States.abb will be abb).

Exercise 10
Add the previous data frame column-wise to state.x77
a) Remove the variable div.
b) Also remove the variables Life Exp, HS Grad, Frost, abb, and are.
c) Add a variable to the data frame which should categorize the level of illiteracy:
[0,1) is low, [1,2) is some, [2, inf) is high.
d) Find out which state from the west, with low illiteracy, has the highest income, and what that income is.

List exercises

In the exercises below we cover the basics of lists. Before proceeding, first read section 6.1-6.2 of An Introduction to R, and the help pages for the sum, length, strsplit, and setdiff functions.

Answers to the exercises are available here.

Learn more about lists in the online courses Learn By Example: Statistics and Data Science in R , The Comprehensive Statistics and Data Science with R Course and R Programming: Advanced Analytics In R For Data Science

Exercise 1

If:

p <- c(2,7,8), q <- c("A", "B", "C") and

x <- list(p, q),

then what is the value of x[2]?

a. NULL

b. "A" "B" "C"

c. "7"

Exercise 2

If:

w <- c(2, 7, 8)

v <- c("A", "B", "C")

x <- list(w, v),

then which R statement will replace "A" in x with "K".

a. x[[2]] <- "K"

b. x[[2]][1] <- "K"

c. x[[1]][2] <- "K"

Exercise 3

If a <- list ("x"=5, "y"=10, "z"=15), which R statement will give the sum of all elements in a?

a. sum(a)

b. sum(list(a))
c. sum(unlist(a))

Exercise 4
If Newlist <- list(a=1:10, b="Good morning", c="Hi"), write an R statement that will add 1 to each element of the first vector in Newlist.

Exercise 5
If b <- list(a=1:10, c="Hello", d="AA"), write an R expression that will give all elements, except the second, of the first vector of b.

Exercise 6
Let x <- list(a=5:10, c="Hello", d="AA"), write an R statement to add a new item z = "NewItem" to the list x.

Exercise 7
Consider y <- list("a", "b", "c"), write an R statement that will assign new names "one", "two" and "three" to the elements of y.

Exercise 8
If x <- list(y=1:10, t="Hello", f="TT", r=5:20), write an R statement that will give the length of vector r of x.

Exercise 9
Let string <- "Grand Opening", write an R statement to split this string into two and return the following output:

```
[[1]]
[1] "Grand"

[[2]]
[1] "Opening"
```

Exercise 10
Let:
y <- list("a", "b", "c") and

q <- list("A", "B", "C", "a", "b", "c").
Write an R statement that will return all elements of q that are not in y, with the following result:

```
[[1]]
[1] "A"

[[2]]
[1] "B"

[[3]]
[1] "C"
```

Want some extra practice with lists? Please take a look here