Chapter 2: Data Structures

Kevin Kam Fung Yuen
PhD, Senior Lecturer, School of Business, Singapore University of Social Sciences
kfyuen@suss.edu.sg, kevinkf.yuen@gmail.com

May 26, 2019

Contents

1 Vector

1.1 Create a vector

A vector is a series of element values of the same datatype, which are grouped together. In R, the *c()* function is used to create a vector by grouping elements.

```
v = c(1,3,6)
v
u = c(8,3,5)
u
```

The *c()* function also can be used to add elements to a vector.

```
v  # recall v in R Environment
v=c(v,5,2)
v
u=c(u,4,1)
u
z=c(v,u)
z
```

The *rep()* command is used to create data in the repeated times of a pattern.

```
c=rep(1:5,4)
c
```

The : or seq() operator is used to create a sequence of number. Exanples are as below.

```
1:9
seq(-2, 2, by=.5)
```

1.2 Vector operations

The arithmetic operations for the vectors.

```
u #recall u
v #recall v
(u+v)
(u-v)
(u*v)
(u/v)
(2*v)
(u/v)^2
```

The vector can be processed by the build-in basic statistic functions.

```
#recall in a vector, z
z

# number of elements in z
length(z)
# find the minimun number in a vector
min(z)
# find the maximun number in a vector
max(z)
# sum all numbers in a vector
sum(z)
# find the mean of all numbers in a vector
mean(z)
# find the index of the minimun number in z
which.max(z)
# find the index of the maximun number in z
which.min(z)
```

```r
# return the index of element 4 in z.
which(z==4)
# other functions
median(z)
sum(z)
sd(z)
quantile(z)
```

The integer numbers are used as indices to query the elements in a vector with specific definition.

```r
#recall z
Z

# 3nd element of z
z[3]

# first four elements of z
z[1:4]

# return all but the 4th element of z
z[-4]

# values in z that are < 5
z[z<5]
```

We can create a vector of various datatypes structures. Quite often, when we perform a vector operation, errors may be created due to inconsistence of datatypes structures.

```r
(V1 = paste("X", 1:10, sep=""))

typeof(V1)

(V2 = c(TRUE, FALSE,TRUE))

typeof(V2)

(V3 =1:5)

typeof(V3)
```

2 Matrix

2.1 Create Matrix

A vector is a group of single dimensional data. If we need to create two-dimension data, *matrix()* function is one of the methods. To create a matrix, we use matrix() function to take a vector of data to form a matrix with n rows and m cols. Please ensure that the length of vector and the matrix dimensions should be consistent.

```r
# Create a Matrix called A
A=matrix(c(1:10), nrow=2, ncol=5)

# Create a Matrix called B
B=matrix(c(5,3,6,4,2,6,9,0,8,7),nrow=2,ncol=5)

#display matrices A and B
A
B
```

2.2 Matrix Operations

Arithmetic operations for the matrices typically include scalar product, power, element-by-element multiplication, and Matrix-Matrix Multiplication.

```r
# scalar product
2*A

2*A+3

#power
B^2

A+B

A-B

A/B

# element-by-element multiplication
A*B

#tranposition
t(B)

# Matrix-Matrix Multiplication or scalar("inner") product
A %*% t(B)
```

We use the statistic functions to process the matrices.

```r
A #recall A
nrow(A) # number of rows in A
ncol(A) # number of columes in A
rowSums(A)
colSums(A)
rowMeans(A)
colMeans(A)
apply(A,2,max) # max of each columns
apply(A,1,min) # min of each row
```

We can use indices to access matrix elements.

```
#To access matrix elements
# matrixName[rowNo, colNo]
A #recall A
A[2,3]  # 2nd row, 3rd column element
A[2,]   # 2nd row
A[,3]   # 3nd column of the matrix
A[c(1,2),3:5]   # submatrix of  rows 1-2 and  columns 3-5
```

We can assign row name and colume name for a matrix.

```
#recall A
A #recall A
rownames(A)=c("x1","x2")
colnames(A)=c("y1","y2","y3","y4","y5")
#A display A again
A
```

3 Data Frames

A matrix contains the values of the same datatype. If there are different datatypes, we can use data.frame() to create a data frame, instead of matrix. If there is a missing value, we can use NA.

```
(x1 = c(23, 26, 20, 21, 24))
(x2 = c("red", "black", "yellow", NA, "white"))
(y = c(TRUE,TRUE,TRUE,FALSE, TRUE))
#create data frame by using the above three vectors
(my.data <- data.frame(x1,x2,y))
typeof(my.data)
class(my.data)
#set attribute
(names(my.data) = c("age","color","decision")) #variable name)
my.data
```

we can use *str()* to display the structure of an R object.

```
str(my.data)
```

There are different ways to query the elements of a data frame.

```
my.data[,2] # column 2 of dataframe
my.data[,c("age","decision")] # columns age and decision from dataframe
my.data$color #  to get values of color in the dataframe
my.data$decision
```

We may change an element value in the data frame.

```
my.data # recall
my.data[1, 3] = FALSE
my.data # updated
```

4 List

Data frame presents the data in table form. In other words, the size is uniform. if the size for each variable is not the same, *list* is the ideal data type to install the value List presents the data in list form. We can use list() function to create a list of components of different data structures.

```
x1  #recall
x2  #recall
y   #recall
my.list =list(x1,x2,y)
my.list
```

Define names of the components in a list

```
my.list =list(age=x1,color=x2,decision=y)
my.list
```

We can use *[[]]* or *$* to access components in a list.

```
my.list #recall
my.list[[3]]
my.list$decision
```

5 Missing Values and NULL

As we can see in the example above, there is an NA in "color" component, NA means "Not Available" or "Missing Value". We can use is.na() function to detect if there is any NA value. The function is.na() will return a vector of result if the input is a vector of values. If we want to know where the NA values are, we can use which() to search them.

```
my.list$color #recall
is.na(my.list$color) # check if any NA value for each element
```

Once we find that there is some NA values in the dataset, we can use *which()* to find the index/indices for NA value(s).

```
which(is.na(my.list$color))
```

The *which()* function is used to find the index/indices of TRUE values as is.na return TRUE or FALSE values.

We may remove any row(s) with missing value(s) to create a new dataset. The na.omit() returns the result after incomplete values are removed. The function na.fail() returns the object if it does not contain any missing values, and signals an error otherwise.

```
my.data  #recall
na.fail(my.data) # error message, so we need to clean
```

Now let's see what happen, if *na.omit()* is used.

```
(my.data.clean=na.omit(my.data)) # remove any row(s) with missing value(s)
#try again
na.fail(my.data.clean) # no error
```

Some special kinds of "missing" values, which are produced by numerical computations, are called Not a Number, NaN, values.

```
0/0
4/0
100/0
100
0/100
4/0 -4/0
Inf- Inf
```

The is.na() function is TRUE for both NA and NaN values, whilst is.nan(xx) is only TRUE for NaN values.

```
is.na(0/0)
is.nan(0/0)
is.nan(NA)
is.na(NA)
```

A NULL object is used for the value of an object which is absent. It should not be confused with a vector or list of zero length. The NULL object has no type and no modifiable properties. There is only one NULL object in R, to which all instances refer. The function is.null() is to test if a variable is a NULL object. We cannot set attributes on NULL.

```
(A=NULL)
(B=3)
is.null(A)
is.null(B)
(A=A+1)
(B=B+1)
```

6 Summary

By the end of this unit, we have learnt the environment and installation of R and RStudio, as well as basic operations and concepts in R languages with using R studio.

Concepts and techniques include working directory and R Project, variable assignments, data objects, constants, arithmetic operations, vector and matrix operations, data frame, list, and missing values.

7 Exercises

7.1 Exercise 1

Show your R code to create a vector x as below.

```
## [1] -0.5 -0.3 -0.1  0.1  0.3  0.5
```

7.2 Exercise 2

Show your R code to create a matrix X as below.

```
##   a b c d e f g h
## A 1 1 1 1 0 0 0 0
## B 1 1 1 1 0 0 1 0
## C 1 1 1 1 0 0 0 0
## D 1 1 1 1 1 0 0 0
## E 0 0 0 1 1 1 0 0
## F 0 0 0 0 1 1 0 0
## G 0 1 0 0 0 0 1 0
## H 0 0 0 0 0 0 0 1
```

7.3 Exercise 3

There is a data called *iris* in R. show the R commends 1. display *iris* dataset and its structure 2. display *Species* in the dataset 3. display data where Species is *versicolor* 4. calculate the mean of *Sepal.Width* of all data. 5. calculate the minimum of *Sepal.Width* of *versicolor*.

7.4 Exercise 4

1. Which one(s) is(are) the valid vector?

a. A=(2,10,4,5,6)
b. B= 2:10
c. C=letters[2:4]
d. D=seq(-2,2,0.5)

2. We have the codes below:

```
A= iris
B= matrix(1:12,3)
C= list(A=A,B=B)
```

Which one(s) is(are) correct?

a. C is a list including a data.frame and a matrix.
b. typeof(A) is a data.frame.
c. class(B) is an integer.
d. The above code will produce error(s).

3. Which statement(s) does(do) return TRUE?

a. is.na(0/0)
b. is.na(5/0)
c. is.na(NULL)
d. is.na(0)

References

R Core Team. (2019a). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

R Core Team (2019b), An Introduction to R, free download at
https://cran.r-project.org/doc/manuals/r-release/R-intro.html