

Practical 1

Execution of the basic commands , Array, List and Frames

Data Types

1) Numeric

Decimal values are called numeric in R. It is the default computational data type. If we assign a decimal value to a variable x , x will be of numeric type.

Code:

```
x = 10.5    # assign a decimal value
x           # print the value of x
class(x)    # print the class name of x
is.integer(x) # is k an integer?
```

2) Integer

```
y = as.integer(3) # coerce a numeric value  
y                # print the value of y  
class(y)         # print the class name of y  
is.integer(y)    # is y an integer?
```

ALITER:

```
y = 3L # coerce a numeric value  
y  
is.integer(y) # is y an integer?
```

```
y=as.integer(3.14)  
y  
is.integer(y) # is y an integer?
```

3) Complex

```
z = 1 + 2i      # create a complex number  
z              # print the value of z  
class(z)       # print the class name of z
```

```
sqrt(-1+0i)    # square root of -1
```

OR

```
sqrt(as.complex(-1))
```

4) Logical

```
x = 1; y = 2  # sample values
z = x > y     # is x larger than y?
z            # print the logical value
class(z)     # print the class name of z
```

Standard logical operations are "&" (and), "|" (or), and "!" (negation).

```
u = TRUE; v = FALSE
u & v      # u AND v
u | v      # u OR v
!u         # negation of u
```

5) Character

```
x = as.character(3.14)
x          # print the character string
class(x)   # print the class name of x
```

Two character values can be concatenated with the paste function.

```
fname = "Joe"; lname = "Smith"
paste(fname, lname)
```

However, it is often more convenient to create a readable string with the sprintf function, which has a C language syntax.

```
sprintf("%s has %d dollars", "Sam", 100)
```

To extract a substring, apply the substr function. Following example shows how to extract the substring between the third and twelfth positions in a string.

```
substr("Mary has a little lamb.", start=3, stop=12)
```

To replace the first occurrence of the word "little" by another word "big" in the string, apply the sub function.

```
sub("little", "big", "Mary has a little lamb.")
```

Data Structures

1) Vectors

```
# Create a vector.  
V= c('red','green',"yellow")  
V  
(class(V))  
V= c(1,2,3)  
V  
(class(V))  
V= c(true,false)  
V  
(class(V))  
length(V)
```


Combining Vectors

```
A= c('red','green',"yellow")  
B=c(1,4,6)  
C(A,B)
```

Vector Arithmetics

```
a = c(1, 3, 5, 7)  
b = c(1, 2, 4, 8)  
8*a  
a*b  
a-b  
a+b  
a/b
```

Recycling Rule

If two vectors are of unequal length, the shorter one will be recycled in order to match the longer vector

```
u = c(10, 20, 30)  
v = c(1, 2, 3, 4, 5, 6, 7, 8, 9)  
u + v
```

Vector Index (Retrieve or Strip a vector member)

```
s=c("a","b","c","d","e","f","g")  
s[4]  
s[-2]  
s[10]  
s[c(2,3)]  
s[c(2,3,3)]  
s[c(2,3,2,1)]  
s[2:4]
```

Logical Index Vector

To retrieve the the second and fourth members of s, define a logical vector L of the same length, and have its second and fourth members set as TRUE

```
L = c(FALSE, TRUE, FALSE, TRUE, FALSE)  
s[L]
```

Named Vector Members

```
v = c("Mary", "Sue")  
names(v) = c("First", "Last")  
v  
v["First"]  
v[c("Last", "First")]
```

2) Matrices

```
A = matrix(  
  c(2, 4, 3, 1, 5, 7), # the data elements  
  nrow=2,               # number of rows  
  ncol=3,               # number of columns  
  byrow = TRUE)        # fill matrix by rows  
A  
A[2, 3]                # element at 2nd row, 3rd column  
A[2, ]                 # the 2nd row  
A[, 3]                 # the 3rd column  
A[, c(1,3)]            # the 1st and 3rd columns  
dimnames(A) = list(  
  c("row1", "row2"),   # row names  
  c("col1", "col2", "col3")) # column names  
A                      # print A
```

3) List

A **list** is a generic vector containing other objects.

For example, the following variable x is a list containing copies of three vectors n, s, b, and a numeric value 3.

Create a list

```
list("Red", "Green", c(21,32,11), TRUE, 51.23, 119.1)
```

```
list(c(2,5,3),21.3,sin)
```

```
n = c(2, 3, 5)
s = c("aa", "bb", "cc", "dd", "ee")
b = c(TRUE, FALSE, TRUE, FALSE, FALSE)
x = list(n, s, b, 3)    # x contains copies of n, s, b
x
x[2]
x[c(3,4)]
```

4) Arrays

```
# Create an array.  
a=array(c('green','yellow','red'),dim = c(4,4,4))  
a
```

```
vector1 =c(5,9,3)  
vector2 =c(10,11,12,13,14,15)  
result = array(c(vector1,vector2),dim = c(3,3,2))  
result
```

5) Factors

```
a=c('green','yellow','red') # Create a vector
```

```
factor(a)    # Create a factor object
```

```
nlevels(factor(a))
```

6) Data Frames

Create the data frame.

```
n = c(2, 3, 5)
s = c("aa", "bb", "cc")
b = c(TRUE, FALSE, TRUE)
df = data.frame(n, s, b)
df
```

Create the data frame.

```
gender = c("Male", "Male", "Female"),
height = c(152, 171.5, 165),
weight = c(81, 93, 78),
Age = c(42, 38, 26)
df = data.frame(gender, height, weight, Age)
df
```