R Introduction

Peilin Chen

11/3/2017

R basic syntax and datatypes

1.Assign value to variable using

$< -$ or $=$

```
x<-1
print(x)
```

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

2. Atomic: character, numeric, integer, complex, logical (T/F).

Numbers in R are generally treated as numeric objects. You need to add L to the number to specify it as an integer. Numeric

```
x<-1  #assign 1 to x , will be treated as numeric
x
```

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

```
class(x)
```

```
## [1] "numeric"
```

Integer

```
x<-1L  #assign integer 1 to x
x
```

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

```
class(x)
```

```
## [1] "integer"
```

Character

```
x<-'1'  #assign a character '1' to x
x
```

```
## [1] "1"
```

```r
class(x)
```

```
## [1] "character"
```

Complex

```r
x<-complex(real=1,imaginary = 1)
x
```

```
## [1] 1+1i
```

```r
class(x)
```

```
## [1] "complex"
```

Logical

```r
x<-TRUE
x
```

```
## [1] TRUE
```

```r
class(x)
```

```
## [1] "logical"
```

3. Basic Type of R objects: vector, Lists

A vector can only contain objects of the same class

```r
a <- c(1,2,5.3,6,-2,4) # numeric vector
b <- c("one","two","three") # character vector
c <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE) #logical vector
```

```r
a
```

```
## [1]  1.0  2.0  5.3  6.0 -2.0  4.0
```

```r
b
```

```
## [1] "one"   "two"   "three"
```

```r
c
```

```
## [1]  TRUE  TRUE  TRUE FALSE  TRUE FALSE
```

Implicity Coercion by R

```
y<-c(1.7,'a')
y
```

```
## [1] "1.7" "a"
```

```
class(y)
```

```
## [1] "character"
```

Objects can be explicity Coercion from one class to another class

```
as.character(a)
```

```
## [1] "1"   "2"   "5.3" "6"   "-2"  "4"
```

```
as.logical(a)
```

```
## [1] TRUE TRUE TRUE TRUE TRUE TRUE
```

```
as.numeric(b) #R cannot figure out how to coerce the object, and result in NA
```

```
## Warning: NAs introduced by coercion
```

```
## [1] NA NA NA
```

A list is a generic vector containing other objects.

```
n<-c(1,2,3)
s<-c('a','b')
l<-c(TRUE,FALSE)
x<-list(n,s,l)
x
```

```
## [[1]]
## [1] 1 2 3
##
## [[2]]
## [1] "a" "b"
##
## [[3]]
## [1]  TRUE FALSE
```

Parsing a list: the location index of a list starts from 1

```
x<-list(1,TRUE,'a')
x[1]
```

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

```r
x[2]
```

```
## [[1]]
## [1] TRUE
```

```r
x[3]
```

```
## [[1]]
## [1] "a"
```

Nested list:

```r
x<-list(c(1,2),c(TRUE,FALSE,TRUE),'a')
x[[1]] #get the first element c(1,2)
```

```
## [1] 1 2
```

```r
x[[1]][1] #get 1 in c(1,2)
```

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

```r
x[[2]]  #get the second element c(TRUE,FALSE,TRUE)
```

```
## [1]  TRUE FALSE  TRUE
```

```r
x[[2]][2] #get the FALSE in c(TRUE,FALSE,TRUE)
```

```
## [1] FALSE
```

```r
x[[3]][2] #exceed limit, return NA
```

```
## [1] NA
```

4. factors: represent categorical data

```r
x <- factor(c("yes", "yes", "no", "yes", "no"))
x
```

```
## [1] yes yes no  yes no
## Levels: no yes
```

5. missing values: is.na() or is.nan()

```
x<-NA
is.na(x)
```

```
## [1] TRUE
```

6. matrices, data frames

matrics and data frames are a special list where every element in the list has the same length **matrices must have every element the same class(e.g. All integers, all numeric, all character)**

```
m<-matrix(1,nrow = 3,ncol = 4)
m
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    1    1    1    1
## [3,]    1    1    1    1
```

add column name and row name to m

```
colnames(m)<-c('a','b','c','d')
rownames(m)<-c('e','f','g')
m
```

```
##   a b c d
## e 1 1 1 1
## f 1 1 1 1
## g 1 1 1 1
```

dataframe can have different type class of entries

```
df<-data.frame(foo = 1:4, bar = c(T, T, F, F))
df
```

```
##   foo   bar
## 1   1  TRUE
## 2   2  TRUE
## 3   3 FALSE
## 4   4 FALSE
```

```
dim(df)
```

```
## [1] 4 2
```

reset names of a data frame

```
names(df)<-c('column1','column2') #reset column names
row.names(df)<-c('row1','row2','row3','row4')
df
```

```
##      column1 column2
## row1       1    TRUE
## row2       2    TRUE
## row3       3   FALSE
## row4       4   FALSE
```

Summary.

In fact, everything in R is an object. An object is a data structure having some attributes and methods which act on its attributes.

THis introduction workshop used examples in the following website: https://www.statmethods.net/input/datatypes.html and https://bookdown.org/rdpeng/RProgDA/

There are many other online resources. Stackover flow is a good place to ask and get answers for your questions.