IntroR

Boyie Chen

9/24/2019

Using R as a simple caculator

Please compile the following codes in RStudio line by line. You can either click "run" or use "Ctr+Enter" (or Command+Enter in MacOS) to compile the code.

```
##Simple Calculator
1+1
```

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

```
3*4
```

```
## [1] 12
```

```
3**4 #power
```

```
## [1] 81
```

```
sqrt(2) #square root of
```

```
## [1] 1.414214
```

```
pi*3^2
```

```
## [1] 28.27433
```

```
log(exp(1))
```

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

```
3/4
```

```
## [1] 0.75
```

Note that some operations do not give you a number.

```
-Inf/Inf #"NaN" means not a number
```

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

```
NA #"NA" means Not Available
```

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

```
5/0
```

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

```
10-Inf
```

```
## [1] -Inf
```

```
log(0)
```

```
## [1] -Inf
```

And the following operations are more like functions...

```r
3%%4 #return the remainder
```

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

```r
sin(pi/2)
```

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

```r
round(1234.5678, 1)
```

```
## [1] 1234.6
```

```r
round(1234.5678, 2)
```

```
## [1] 1234.57
```

```r
round(1234.5678, -2)
```

```
## [1] 1200
```

```r
floor(pi)
```

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

```r
ceiling(pi)
```

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

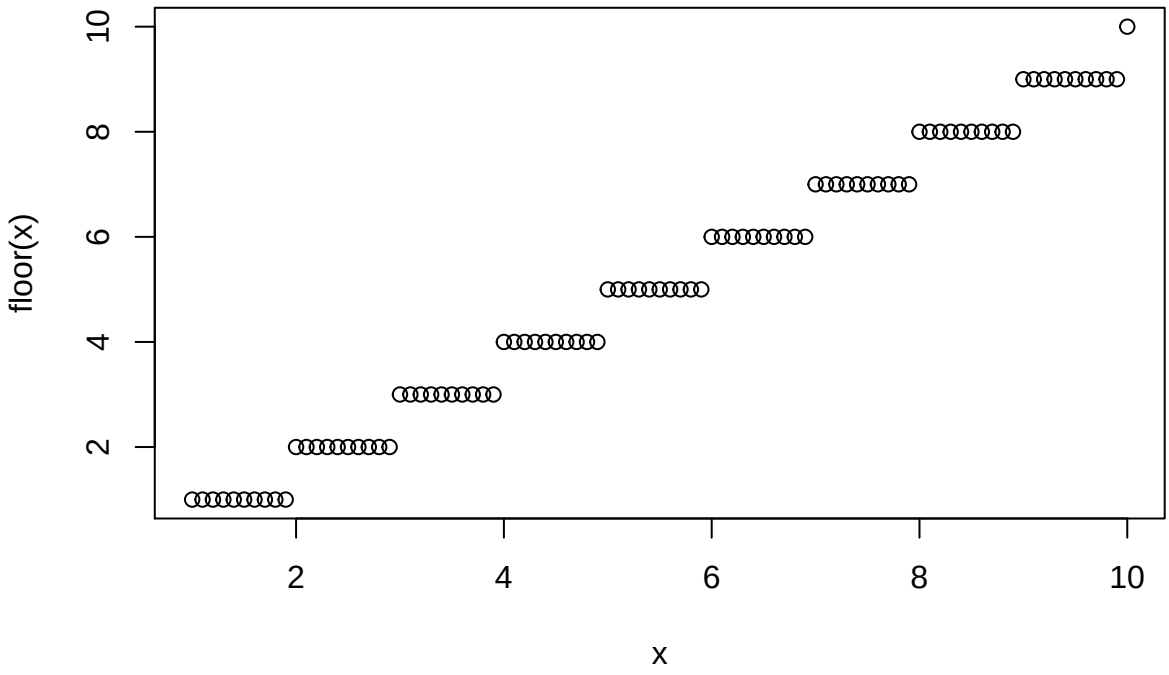
```r
trunc(pi)
```

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

```r
x = seq(1, 10, by = 0.1)
x
```

```
##  [1]  1.0  1.1  1.2  1.3  1.4  1.5  1.6  1.7  1.8  1.9  2.0  2.1  2.2  2.3
## [15]  2.4  2.5  2.6  2.7  2.8  2.9  3.0  3.1  3.2  3.3  3.4  3.5  3.6  3.7
## [29]  3.8  3.9  4.0  4.1  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.0  5.1
## [43]  5.2  5.3  5.4  5.5  5.6  5.7  5.8  5.9  6.0  6.1  6.2  6.3  6.4  6.5
## [57]  6.6  6.7  6.8  6.9  7.0  7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9
## [71]  8.0  8.1  8.2  8.3  8.4  8.5  8.6  8.7  8.8  8.9  9.0  9.1  9.2  9.3
## [85]  9.4  9.5  9.6  9.7  9.8  9.9 10.0
```

```r
plot(x, floor(x))
```



Assigning Variables

You can also embed plots, for example:

```
a = 2
b = 3
a+b
```

```
## [1] 5
```

```
print(a+b)
```

```
## [1] 5
```

```
rm(b) #remove b
rm(list = ls()) #Remove all variable
a = 2
```

Several frequent used data structure

We've mentioned values, now we can look at other data types.

```
#vectors
c1 = c(0,1,2,3,4)
c1
```

```
## [1] 0 1 2 3 4
```

```
is.vector(c1) #When you want to find out which data type it is
```

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

```
a*c1
```

```
## [1] 0 2 4 6 8
```

```
c1 = 1:5
c1 = seq(1,5, by = 1)
```

```r
c2 = c(2,4,6,8,10)
c1+c2 #addition
```

```
## [1]  3  6  9 12 15
```

```r
c1*c2 #"*" multiplication
```

```
## [1]  2  8 18 32 50
```

```r
c3 = c(c(1,2,3),
       c(4,5,6)) #still being a n*1 vector
length(c1)
```

```
## [1] 5
```

```r
c1[4]
```

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

```r
c2[4]
```

```
## [1] 8
```

Logic (Boolean)

```r
c1[2] == 2
```

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

```r
c2[4] > 8
```

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

```r
length(c3) != 5
```

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

Easily used functions:

```r
mean(c1)
```

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

```r
min(c1)
```

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

```r
max(c1)
```

```
## [1] 5
```

```r
sum(c1) # R^n -> R^1
```

```
## [1] 15
```

```r
log(c1) # R^n -> R^n
```

```
## [1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379
```

```r
factorial(c1)
```

```
## [1]   1   2   6  24 120
```

```r
z = c(1, 2, NA, 3, NA, 4)
z
```

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

```
is.na(z)
```

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

```
mean(z)
```

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

```
mean(z, na.rm = TRUE)
```

```
## [1] 2.5
```

Character

```
x1 = "I love Statistics"
x1[1]
```

```
## [1] "I love Statistics"
```

```
x1[2]
```

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

```
length(x1)
```

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

```
x2 = c("I", "love", "Statistics")
x2[3]
```

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

```
length(x2)
```

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

```
nchar(x2)
```

```
## [1]  1  4 10
```

Matrices

```
A1 = matrix(1:15, nrow = 5)
A1
```

```
##      [,1] [,2] [,3]
## [1,]    1    6   11
## [2,]    2    7   12
## [3,]    3    8   13
## [4,]    4    9   14
## [5,]    5   10   15
```

```
A2 = matrix(16:30, ncol = 3) #assign either one is enough
A2
```

```
##      [,1] [,2] [,3]
## [1,]   16   21   26
## [2,]   17   22   27
## [3,]   18   23   28
## [4,]   19   24   29
## [5,]   20   25   30
```

```r
A3 = matrix(1:15, nrow = 5, ncol = 5) #if exceed 1:15
A3
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    6   11    1    6
## [2,]    2    7   12    2    7
## [3,]    3    8   13    3    8
## [4,]    4    9   14    4    9
## [5,]    5   10   15    5   10
```

```r
A1 = matrix(1:15, nrow = 5, dimnames = list(c('r1', 'r2', 'r3', 'r4', 'r5'),
                                            c('c1', 'c2', 'c3')))
A1
```

```
##    c1 c2 c3
## r1  1  6 11
## r2  2  7 12
## r3  3  8 13
## r4  4  9 14
## r5  5 10 15
```

```r
nrow(A1)
```

```
## [1] 5
```

```r
ncol(A1)
```

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

```r
dim(A1)
```

```
## [1] 5 3
```

```r
colnames(A1)
```

```
## [1] "c1" "c2" "c3"
```

```r
rownames(A1)
```

```
## [1] "r1" "r2" "r3" "r4" "r5"
```

```r
head(A1)
```

```
##    c1 c2 c3
## r1  1  6 11
## r2  2  7 12
## r3  3  8 13
## r4  4  9 14
## r5  5 10 15
```

```r
tail(A1)
```

```
##    c1 c2 c3
## r1  1  6 11
## r2  2  7 12
## r3  3  8 13
## r4  4  9 14
## r5  5 10 15
```

```r
A1+A1
```

```
##    c1 c2 c3
```

```
## r1  2 12 22
## r2  4 14 24
## r3  6 16 26
## r4  8 18 28
## r5 10 20 30
```

```
A1*A1
```

```
##    c1  c2  c3
## r1  1  36 121
## r2  4  49 144
## r3  9  64 169
## r4 16  81 196
## r5 25 100 225
```

```
A1t = t(A1) #transpose
A1%*%A1t # "%*%" multiplication
```

```
##     r1  r2  r3  r4  r5
## r1 158 176 194 212 230
## r2 176 197 218 239 260
## r3 194 218 242 266 290
## r4 212 239 266 293 320
## r5 230 260 290 320 350
```

```
A1[3, ] #list the 3rd row
```

```
## c1 c2 c3
##  3  8 13
```

```
A1[3, 1:2] #list the 3rd and only the 1st to 2nd cols
```

```
## c1 c2
##  3  8
```

```
A1[, 2] #list the 2nd col
```

```
## r1 r2 r3 r4 r5
##  6  7  8  9 10
```

```
A1[, 2:3]
```

```
##    c2 c3
## r1  6 11
## r2  7 12
## r3  8 13
## r4  9 14
## r5 10 15
```

```
cbind(A1, A2)
```

```
##    c1 c2 c3
## r1  1  6 11 16 21 26
## r2  2  7 12 17 22 27
## r3  3  8 13 18 23 28
## r4  4  9 14 19 24 29
## r5  5 10 15 20 25 30
```

```
colnames(cbind(A1, A2))
```

```
## [1] "c1" "c2" "c3" ""   ""   ""
```

```
rbind(A1, A2)
```

```
##    c1 c2 c3
## r1  1  6 11
## r2  2  7 12
## r3  3  8 13
## r4  4  9 14
## r5  5 10 15
##    16 21 26
##    17 22 27
##    18 23 28
##    19 24 29
##    20 25 30
```