101023 Statistical Learning

Mattia G.

2023-10-10

R commands notes 1

Data Structures:

- Vectors
- Matrices
- Dataframe
- Factors

```
#### "colon" operator

x <- 1:10
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
seq(1,9, by = 2)
```

```
## [1] 1 3 5 7 9
```

```
seq(8, 20, length = 6)
```

```
## [1]  8.0 10.4 12.8 15.2 17.6 20.0
```

```
x <- rep(1, 10)
x <- rep(c(1, 5), 3)
y <- c(rep(2, 3), 4, 5, rep(1,5))

a <- 1:5
a
```

```
## [1] 1 2 3 4 5
```

```
b<- 7:11

a > b
```

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

```
#if the len of a vector is the multiple of the other it's all good in the
# neighborhood, otherwise you have to be worried


#### "vectorised" functions

log(b)
```

```
## [1] 1.945910 2.079442 2.197225 2.302585 2.397895
```

```
#### some commonly used functions

x <- 22:3
length(x)
```

```
## [1] 20
```

```
max(x)
```

```
## [1] 22
```

```
min(x)
```

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

```
sum(x)
```

```
## [1] 250
```

```
prod(x)
```

```
## [1] 5.620004e+20
```

```
sort(x)
```

```
##  [1]  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
```

```
#### 3 types of variables: Numerical, Character, Logical
# character and logical vectors

y <- c("this", "is", "an", "example")
y
```

```
## [1] "this"    "is"      "an"      "example"
```

```
z <- c(5<2, 3>1, 1==0, FALSE)
z
```

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

```
#### conversion of elements to the same element

x <- c(TRUE, "example", 5)
x
```

```
## [1] "TRUE"    "example" "5"
```

```
#### R does convert all the above vector into a string
# R does convert the next vector into a numerical vector (FALSE = 0, TRUE = 1)

y <- c(TRUE, FALSE, 5)
y
```

```
## [1] 1 0 5
```

```
#### missing values

a <- c(1, NA, 2)
st <- c("this", "is", "an", "example", NA)

a
```

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

```
st
```

```
## [1] "this"    "is"      "an"      "example" NA
```

```
#### extract elements from a vector

x<- seq(0, 20, 10)
x
```

```
## [1]  0 10 20
```

```
x[1]
```

```
## [1] 0
```

```
x[3]
```

```
## [1] 20
```

```
x[1:2]
```

```
## [1]  0 10
```

```
x[c(1, 3)]
```

```
## [1]  0 20
```

```
x <- seq(1, 10, 2)
x
```

```
## [1] 1 3 5 7 9
```

```
#### Beware the negative indexes, in R by indicating a negative index it
# indicates that I dont' want the element in that spec pos to be returned

x[c(-1, -4)]
```

```
## [1] 3 5 9
```

```
#### name the elements of a vector

x <- 1:4
x
```

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

```
names(x)
```

```
## NULL
```

```
names(x) <- c("a", "b", "c", "d")

x
```

```
## a b c d
## 1 2 3 4
```

```
names(x)
```

```
## [1] "a" "b" "c" "d"
```

```
x["a"]
```

```
## a
## 1
```

```
x[c("a", "b")]
```

```
## a b
## 1 2
```

```
#### use logical values to access the elements of a vector

weight <- c(80, 70, 82, 76, 90)
height <- c(170, 168, 176, 181, 180)

height[c(3, 5)]
```

```
## [1] 176 180
```

```
weight.more.than.80 <- c(FALSE, FALSE, TRUE, FALSE, FALSE)
weight.more.than.80
```

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

```
height[weight.more.than.80]
```

```
## [1] 176
```

```
weight.more.than.80 <-weight > 80

weight.more.than.80
```

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

```
height[weight > 80]
```

```
## [1] 176 180
```

```
#### matrices

a <- matrix(1:6, nrow = 2)
a
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
b <- matrix(1:6, nrow = 2, byrow = TRUE)
b
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
```

```
x <- 3:8
matrix(x, ncol=2)
```

```
##      [,1] [,2]
## [1,]    3    6
## [2,]    4    7
## [3,]    5    8
```

```
matrix(x, ncol=2, byrow = TRUE)
```

```
##      [,1] [,2]
## [1,]    3    4
## [2,]    5    6
## [3,]    7    8
```

```
x <- c("batman", "robin", "superman", "spiderman")

matrix(x, ncol=2)
```

```
##      [,1]     [,2]
## [1,] "batman" "superman"
## [2,] "robin"  "spiderman"
```

```
matrix(c(3>=2, 3<=5, 2==2, 1<0), nrow = 2)
```

```
##      [,1]  [,2]
## [1,] TRUE  TRUE
## [2,] TRUE FALSE
```

```
#### rbind() cbind() Row/Col

a <- rbind(weight, height)
a
```

```
##        [,1] [,2] [,3] [,4] [,5]
## weight   80   70   82   76   90
## height  170  168  176  181  180
```

```
b <- cbind(weight, height)
b
```

```
##      weight height
## [1,]     80    170
## [2,]     70    168
## [3,]     82    176
## [4,]     76    181
## [5,]     90    180
```

```
#### dimension of a matrix

dim(a)
```

```
## [1] 2 5
```

```
dim(a)[1]
```

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

```
dim(a)[2]
```

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

```
b[2][2] # b[row][column]
```

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

```
a[1, ] #All columns
```

```
## [1] 80 70 82 76 90
```

```
a[, 2] #All Rows
```

```
## weight height
##     70    168
```

```
b[1:3,2]
```

```
## [1] 170 168 176
```

```
a[1, c(1, 4, 5)]
```

```
## [1] 80 76 90
```

```
b[b[,1] > 80]
```

```
## [1]  82  90 176 180
```

```
b[b[ ,2]> 175, ]
```

```
##      weight height
## [1,]     82    176
## [2,]     76    181
## [3,]     90    180
```

```
b[-2, ]
```

```
##      weight height
## [1,]     80    170
## [2,]     82    176
## [3,]     76    181
## [4,]     90    180
```

```
#### colnames and rownames

colnames(b)
```

```
## [1] "weight" "height"
```

```
rownames(b)
```

```
## NULL
```

```
rownames(b)<- c("batman", "robin", "superman", "spiderman", "ironman")
rownames(b)
```

```
## [1] "batman"    "robin"     "superman"  "spiderman" "ironman"
```

```
#### operation with matrixes

a <- matrix(0:5, nrow = 2)
b <-matrix(seq(0, 10, 2), nrow = 2)

a
```

```
##      [,1] [,2] [,3]
## [1,]    0    2    4
## [2,]    1    3    5
```

```
b
```

```
##      [,1] [,2] [,3]
## [1,]    0    4    8
## [2,]    2    6   10
```

```
a+b
```

```
##      [,1] [,2] [,3]
## [1,]    0    6   12
## [2,]    3    9   15
```

```
a*b
```

```
##      [,1] [,2] [,3]
## [1,]    0    8   32
## [2,]    2   18   50
```

```
a==5
```

```
##       [,1]  [,2]  [,3]
## [1,] FALSE FALSE FALSE
## [2,] FALSE FALSE  TRUE
```

```
a/b
```

```
##      [,1] [,2] [,3]
## [1,]  NaN  0.5  0.5
## [2,]  0.5  0.5  0.5
```

```
# in R operations between matrixes are element by element but if I want do them
# with matrix algebra:

#### matrix algebra with R

a
```

```
##      [,1] [,2] [,3]
## [1,]    0    2    4
## [2,]    1    3    5
```

```
b <- matrix(seq(0,10,2), nrow = 3)
b
```

```
##      [,1] [,2]
## [1,]    0    6
## [2,]    2    8
## [3,]    4   10
```

```
d <- a%*%b
d
```

```
##      [,1] [,2]
## [1,]   20   56
## [2,]   26   80
```

```
#### matrix inversion

solve(d)
```

```
##            [,1]       [,2]
## [1,]  0.5555556 -0.3888889
## [2,] -0.1805556  0.1388889
```

```
d%*%solve(d)
```

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

```
## [1,] 1.000000e+00 2.220446e-16
## [2,] 4.440892e-16 1.000000e+00
```

```
#### transpose

t(d)
```

```
##      [,1] [,2]
## [1,]   20   26
## [2,]   56   80
```

```
#### Dataframes

weight <- c(80, 70, 82, 76, 90)
height <- c(170, 198, 176, 181, 180)
smoker <- c("yes", "yes", "no", "no", "yes")
survey <- data.frame(weight, height, smoker)

survey
```

```
##   weight height smoker
## 1     80    170    yes
## 2     70    198    yes
## 3     82    176     no
## 4     76    181     no
## 5     90    180    yes
```

```
survey[1:3, ]
```

```
##   weight height smoker
## 1     80    170    yes
## 2     70    198    yes
## 3     82    176     no
```

```
survey[ , 2]
```

```
## [1] 170 198 176 181 180
```

```
survey[3, ]
```

```
##   weight height smoker
## 3     82    176     no
```

```
survey[survey[ , 3]=="yes", ]
```

```
##   weight height smoker
## 1     80    170    yes
## 2     70    198    yes
## 5     90    180    yes
```

```
survey[survey[ , 3]=="yes", -3]
```

```
##   weight height
## 1     80    170
## 2     70    198
## 5     90    180
```

```
#### the '$' operator (I need a dollar, dollar a dollar is what I need hey hey)

names(survey)
```

```
## [1] "weight" "height" "smoker"
```

```
survey$weight
```

```
## [1] 80 70 82 76 90
```

```
survey$weight[1:3]
```

```
## [1] 80 70 82
```

```
# Every obj has attributes but there is an attribute that gets assigned by default,
# and this is "class"

#### class of an object

M <- matrix(1:9, ncol = 3)
class(M)
```

```
## [1] "matrix" "array"
```

```
is.matrix(M)
```

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

```
is.data.frame(M)
```

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

```
class(survey)
```

```
## [1] "data.frame"
```

```
is.data.frame(survey)
```

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

```
is.matrix(survey)
```

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

```
#### vectors have no specific class

a <- 1:10
is.vector(a)
```

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

```
is.numeric(a)
```

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

```
is.character(a)
```

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

```
class(a)
```

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

```
b <- c("this", "is", "a", "string")
is.vector(b)
```

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