A.R

George

Sat May 13 18:28:20 2017

setwd("~/Desktop/Computational Stats/R/R")  
  
  
#does 2 equal 3?  
2 == 3

## [1] FALSE

#does 2 not equal 3  
2 != 3

## [1] TRUE

#is 2 less than 3  
2<3

## [1] TRUE

#is 2 less than or equal to 3  
2 <= 3

## [1] TRUE

#Vecotrs   
  
#creating a vector  
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)  
x

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

#operations  
x \* 3

## [1] 3 6 9 12 15 18 21 24 27 30

x + 2

## [1] 3 4 5 6 7 8 9 10 11 12

x - 3

## [1] -2 -1 0 1 2 3 4 5 6 7

x / 4

## [1] 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00 2.25 2.50

x ^ 2

## [1] 1 4 9 16 25 36 49 64 81 100

sqrt(x)

## [1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751  
## [8] 2.828427 3.000000 3.162278

#optional vector creation  
1:10

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

10:1

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

-2:3

## [1] -2 -1 0 1 2 3

5:-7

## [1] 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7

#create two vectors of equal length  
x <- 1:10  
y <- -5:4  
#operations  
x + y

## [1] -4 -2 0 2 4 6 8 10 12 14

x - y

## [1] 6 6 6 6 6 6 6 6 6 6

x\*y

## [1] -5 -8 -9 -8 -5 0 7 16 27 40

x/y

## [1] -0.2 -0.5 -1.0 -2.0 -5.0 Inf 7.0 4.0 3.0 2.5

x^y

## [1] 1.000000e+00 6.250000e-02 3.703704e-02 6.250000e-02 2.000000e-01  
## [6] 1.000000e+00 7.000000e+00 6.400000e+01 7.290000e+02 1.000000e+04

#check the length of each  
length(x)

## [1] 10

length(y)

## [1] 10

#the length of them added together should be the same  
length(x+y)

## [1] 10

x + c(1,2)

## [1] 2 4 4 6 6 8 8 10 10 12

x + c(1, 2, 3)

## Warning in x + c(1, 2, 3): longer object length is not a multiple of  
## shorter object length

## [1] 2 4 6 5 7 9 8 10 12 11

x <= 5

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

x > y

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

x < y

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

# all and any functions  
  
x <- 10:1  
y <- -4:5  
any(x<y)

## [1] TRUE

all(x<y)

## [1] FALSE

nchar(y)

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

#accessing individual elements  
  
x[1]

## [1] 10

x[x<10]

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

#provide a name for each element of an array using a name-value pair  
c(One = "a", Two = "y", Last = "r")

## One Two Last   
## "a" "y" "r"

#create the vector  
w <- 1:3  
#name elements  
names(w) <- c("a", "b", "c")  
w

## a b c   
## 1 2 3

# Factor Vectors  
q <- c("Hockey", "Football", "Baseball", "Curling", "Rugby", "Lacrosse", "Basketball", "Tennis", "Cricket", "Soccer")  
q2 <- c(q, "Hockey", "Lacrose", "Hockey", "Water Polo", "Hockey", "Lacrose")  
  
q2Factor <- as.factor(q2)  
q2Factor

## [1] Hockey Football Baseball Curling Rugby Lacrosse   
## [7] Basketball Tennis Cricket Soccer Hockey Lacrose   
## [13] Hockey Water Polo Hockey Lacrose   
## 12 Levels: Baseball Basketball Cricket Curling Football Hockey ... Water Polo

as.numeric(q2Factor)

## [1] 6 5 1 4 9 8 2 11 3 10 6 7 6 12 6 7

factor(x=c("High School", "College", "Masters", "Doctorate"), levels=c("High School", "College", "Masters", "Doctorate"), ordered=TRUE)

## [1] High School College Masters Doctorate   
## Levels: High School < College < Masters < Doctorate

#Making Matrices  
  
#column  
v <- c(1, 5, 17, 0, 11)  
w <- 1:5  
m <- cbind(w,v)  
m

## w v  
## [1,] 1 1  
## [2,] 2 5  
## [3,] 3 17  
## [4,] 4 0  
## [5,] 5 11

#bind as rows  
m<-rbind(w,v)  
m

## [,1] [,2] [,3] [,4] [,5]  
## w 1 2 3 4 5  
## v 1 5 17 0 11

#bind matrices  
rbind(m,m)

## [,1] [,2] [,3] [,4] [,5]  
## w 1 2 3 4 5  
## v 1 5 17 0 11  
## w 1 2 3 4 5  
## v 1 5 17 0 11

cbind(m,m)

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]  
## w 1 2 3 4 5 1 2 3 4 5  
## v 1 5 17 0 11 1 5 17 0 11

#matrix function   
u<- matrix(1:6, nrow=2)  
u

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

matrix(NA, nrow=3, ncol=2)

## [,1] [,2]  
## [1,] NA NA  
## [2,] NA NA  
## [3,] NA NA

u[2,3]

## [1] 6

#select subset  
u[2, 1:2]

## [1] 2 4

#all rows  
u[,1:2]

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

u == 3

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

which (u ==3)

## [1] 3

u[which(u == 3)] <-5  
u

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

#make the first and third columns of u all 1s  
u[,c(1,3)] <- rep(1,4)  
u

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

#find dimensions of matrix  
dim(u)

## [1] 2 3

ncol(u)

## [1] 3

nrow(u)

## [1] 2

#transpose matrix  
t(u)

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

t <- u + 3  
t

## [,1] [,2] [,3]  
## [1,] 4 8 4  
## [2,] 4 7 4

#Matrix Algebra  
  
u<- matrix(1:6, nrow=2)  
u

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

u[3] <- 5  
u

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

v <- 1:3  
v

## [1] 1 2 3

#multiplying matrices  
  
u %\*% v

## [,1]  
## [1,] 26  
## [2,] 28

u

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

t(u)

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

u %\*% t(u)

## [,1] [,2]  
## [1,] 51 52  
## [2,] 52 56