R-intro notes

Install.packages(“”) ------ download the package from the Internet

Library() ----- load the downloaded package

Vector:

X <- c(1,2,3)

Y <- c(1,2,3,4,5)

V <- 2\*x +y

-The warning message will appear but it’s ok, the x will keep iterating.

-If the vectors are of different lengths, R 'recycles' the shorter vector until it is the same length as the longer vector.

my\_char <- c("My", "name", "is")

paste(my\_char, collapse = " ") --what you want add into the strings(single string)

paste("Hello","world!", sep=" ") – two strings add up

Sequence:

Seq(0,10,by = 0.5)

Seq(5,10,length = 30)

Rep( c(0,1,2), times = 10)

Rep ( c(0,1,2) , each = 10)

Logcial:

A | B (union) A and B at least one

A & B (intersection) A and B

TRUE & c(TRUE, FALSE, FALSE) == c (True, True, True) & c(TRUE, FALSE, FALSE) -- iterating the True

TRUE && c(TRUE, FALSE, FALSE) only evaluate the first one

TRUE FALSE FALSE

All AND operators are evaluated before OR operators

Function:

Identical(a,b) -- like the “is” in python

isTRUE(x) – return True is x is true

xor (a,b) only return TRUE when a , b == TRUE AND FALSE

which() take in a logical vector return the indices of the vector are True

any() take in a logical vector, if there is one evaluate True, then True

all() take in a logical vector, if all evaluate True, then True

11.7

Missing value:

NA – not available, missing value

NaN – not a number (0/0 )/ ( Inf – Inf)

is.na(x) give you a logic table

to tell how many NA just need to sum up the is.na table

because True == 1 False == 0

select element:

four types:

1. logical vectors
2. vectors of positive integers (same as python)
3. vectors of negative integers (exclude the negative value)
4. vectors of character strings

operation:

x[is.na(x)] get all the NA

x [! is.na(x)] get all the None- NA

y[y>0] get all y>0

x [! is.na(x) & x>0]

x [c (-2, -10)] exclude the positon 2 and 10

x [-c (2,10)] easy way to put it

when want to extract a sequence , must use c() function to get it

vect <- c(foo=11, bar=2 ,norf = NA) making the name

names(vect)

vect2 <- c (11,2, NA)

names(vect2) <- c("foo","bar","norf")

vect["bar"]

Dataframe and Matrix

Dim() :

1. to find the dimension of the data (if have) / ( if don’t, return Null) (same as attributes())
2. dim(vector) <- c(4,5) assign a dimension to data

matrix : everything should at the same type – the reason why we don’t like it , because as the name is the string, the number will become string

matrix ( data = , nrow = , ncol = ) make the matrix

cbind(), rbind() :

to get one direction of title to the matrix, but it may change the type type

dataframe :

using rownames(dataframe) <- c()

colnames(dataframe) <- c()

to give name of it

Function:

%% -- take reminder

evaluate <- function(func, dat){

func(dat)

}

#function in function

evaluate(function(x){x+1}, 6)

#anonymous function

"unpack" arguments from an ellipses

> add\_two <- function(...){

+ args <- list(...)

+ a <- args[["alpha"]]

+ b <- args[["beta"]]

+ a + b

+ }

add\_two(alpha = 3, beta = 4) # don’t use alpha <- 3 , it doesn’t work

7

Binary operators

"%mult\_add\_one%" <- function(left, right){ # Notice the quotation marks!

left \* right + 1

}

4 %mult\_add\_one% 5

21

Tricks (short-cut):

Using up arrow to find the previous operation

Type the first two letters and then hit the Tab to find the store created variables.

Command + Return : run the current line ( in R script )

Operations

Function: rnorm() normal distribution

Rep(x, n) – repeat x n times

Sample ( x ,n) from x get n point

Class() to find the type: numeric, character