Brigit Harvey

Primer of Ecology with R Appendix B homework

> #

> these are how to take notes

Error: unexpected symbol in "these are"

> # this is how you take notes

> # this will calculate the mean of 10 random standard normal variables

> mean( rnorm(10))

[1] -0.5075613

> mean(rnorm(10))

[1] 0.1289094

> '?'(mean)

> help("mean",htmlhelp = FALSE)

Error in help("mean", htmlhelp = FALSE) :

unused argument (htmlhelp = FALSE)

> options(htmlhelp=FALSE)

> options(htmlhelp=FALSE)

>

> help(mean)

> help("mean", htmlhelp= FALSE)

Error in help("mean", htmlhelp = FALSE) :

unused argument (htmlhelp = FALSE)

> options(htmlhelp = FALSE)

> help.search("mean")

> apropos("mean")

[1] ".colMeans" ".rowMeans" "colMeans"

[4] "kmeans" "mean" "mean.Date"

[7] "mean.default" "mean.difftime" "mean.POSIXct"

[10] "mean.POSIXlt" "rowMeans" "weighted.mean"

> #these are different commands related to the keyword mean

> #help for functions that you have not downloaded yet

> RSiteSearch("violin")

A search query has been submitted to http://search.r-project.org

The results page should open in your browser shortly

> #success

> RSiteSearch("violin", restrict = c("functions"))

A search query has been submitted to http://search.r-project.org

The results page should open in your browser shortly

> #success

> help(RSiteSearch)

> #success

> #B.2 Assignment

> a <- 2 + 3

> a

[1] 5

> b <- a + a

> a + a; a + b

[1] 10

[1] 15

> #can perform two actions on one line with a semicolon, sometimes the semicolon is referred to as an "end of line" operator

> #B.3 Data Structures

> #scalar is a single real number

> # a vector may be a column of scalars, vector is a group of numbers arranged in a row or column)

> Y<- c(8.3, 8.6, 10.7, 10.8, 11, 11, 11.1, 11.2, 11.3, 11.4)

> #Data appeared in Global Environment

> #can also use an '=' instead of '<-'

> #Sequences

> 1:4

[1] 1 2 3 4

> 4:1

[1] 4 3 2 1

> -1:3

[1] -1 0 1 2 3

>

> -(1:3)

[1] -1 -2 -3

> seq(from = 1, to = 3, by = 0.2)

[1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0

> seq(1, 3, by = 0.2)

[1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0

> seq(1, 3, 0.2)

[1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0

> # can use 'by' or just add an additional comma

> seq(1, 3, length = 7)

[1] 1.000000 1.333333 1.666667 2.000000 2.333333

[6] 2.666667 3.000000

> # length is 7 units

> rep(1,3)

[1] 1 1 1

> rep(1:3,2)

[1] 1 2 3 1 2 3

> rep(1:3, each = 2)

[1] 1 1 2 2 3 3

> #Getting information about vectors

> sum(Y)

[1] 105.4

> mean(Y)

[1] 10.54

> max(Y)

[1] 11.4

> length(Y)

[1] 10

> summary(Y)

Min. 1st Qu. Median Mean 3rd Qu. Max.

8.30 10.72 11.00 10.54 11.18 11.40

> #vector can be a character or logical

> Names <- c("Sarah", "Yunluan")

> Names

[1] "Sarah" "Yunluan"

> b <- c(TRUE, FALSE)

> b

[1] TRUE FALSE

> #vectors can also be dates, complex numbers, real numbers, integers, factors- can also what classes of data it belongs to

> class(Y)

[1] "numeric"

> class(b)

[1] "logical"

> #when we test an object we get a logical vector back that tells us for each element whether the condition was true or false

> Y > 10

[1] FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE

[9] TRUE TRUE

> Y > mean(Y)

[1] FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE

[9] TRUE TRUE

> Y == 11

[1] FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE

[9] FALSE FALSE

> a test of equal to

Error: unexpected symbol in "a test"

> # a test of not equal to

> Y != 11

[1] TRUE TRUE TRUE TRUE FALSE FALSE TRUE TRUE

[9] TRUE TRUE

> #Algebra with vectors

> #we can add, subtract, multiply and divide vectors

> a <- 1:3

> b <- 4:6

> a + b

[1] 5 7 9

> a \* b

[1] 4 10 18

> a/b

[1] 0.25 0.40 0.50

> a + 1

[1] 2 3 4

> a \* 2

[1] 2 4 6

> 1/a

[1] 1.0000000 0.5000000 0.3333333

> # R is recycling the scalar as many times as it needs to in order to match the length of the vector. If we try to multiply vectors of unequal length, R will perform the operation but may or may not give a warning- if multiply a vector of length 3 by vector of length 2, R gives warning

> a \* 1:2

[1] 1 4 3

Warning message:

In a \* 1:2 :

longer object length is not a multiple of shorter object length

> #R recycles the shorter vector just enough to match the length of the longer vector

> a \* c(1, 2, 1)

[1] 1 4 3

> #if multiply vectors of length 4 and 2, no error recieved

> 1:4 \* 1:2

[1] 1 4 3 8

> 1:4 \* c(1, 2, 1, 2)

[1] 1 4 3 8

> # Extraction and missing values

> #can extract or subset elements by identifying the rows or columns needed or providing a logical vector (TRUE or FALSE) of same length as vector I am subsetting

> Y[1]

[1] 8.3

> Y[1:3]

[1] 8.3 8.6 10.7

> Y > mean(Y)

[1] FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE

[9] TRUE TRUE

> Y [Y > mean(Y)]

[1] 10.7 10.8 11.0 11.0 11.1 11.2 11.3 11.4

> #missing data are NA, test which elements are missing with is.na

> a <- c(5, 3, 6, NA)

> a

[1] 5 3 6 NA

> is.na(a)

[1] FALSE FALSE FALSE TRUE

> !is.na(a)

[1] TRUE TRUE TRUE FALSE

> a[!is.na(a)]

[1] 5 3 6

> na.exclude(a)

[1] 5 3 6

attr(,"na.action")

[1] 4

attr(,"class")

[1] "exclude"

> mean(a)

[1] NA

> mean(a, na.rm = TRUE)

[1] 4.666667

> d <- na.exclude(a)

> mean(d)

[1] 4.666667

> #mean removing the NA data

> #Matrices

> #a matrix is a two dimensional set of elements, for which all elements are of the same type

> matrix(letters[1:4], ncol = 2)

[,1] [,2]

[1,] "a" "c"

[2,] "b" "d"

> M <- matrix(1:4, nrow = 2)

> M

[,1] [,2]

[1,] 1 3

[2,] 2 4

> #matrix is filled by columns or column major order, can also do it by rows

> M2 <- matrix(1:4, nrow = 2, byrow = TRUE)

> M2

[,1] [,2]

[1,] 1 2

[2,] 3 4

> I <- diag(1, nrow = 2)

> I

[,1] [,2]

[1,] 1 0

[2,] 0 1

> #Identity matrix plays a role, equivalent to the scalar 1. inverse of matrix M is M^-1 which is the matrix that also satisfies MM^-1 = I where I is the identiy matrix

> Minv <- solve(M)

> M %\*% Minv

[,1] [,2]

[1,] 1 0

[2,] 0 1

> #Extraction of Matrices

> M[1,2]

[1] 3

> M[1,1:2]

[1] 1 3

> M[,2]

[1] 3 4

> #leaving either of the rows or the columns blank, R gives all rows or columns

> M[,]

[,1] [,2]

[1,] 1 3

[2,] 2 4

> #Data Frames

> #data frames are 2D, like spreadsheets and matrices, all columns have same number of rows, each column can have a different data type- a data frame can have names of species, experimental treatment, dimensions of species traits, as character, factor, and numeric variables

> dat <-data.frame(species = c("S.altissima", "S.rugosa", 'E.graminifolia", "A. pilosus"), treatment = factor(c("Control", "Water", "Control", "Water")), height = c(1.1, 0.8, 0.9, 1) width = c(1, 1.7, 0.6, 0.2))

+

+ s

+ stop()

+ =

+ help

+ stp[]

+ == 45

+ #lksjfslkdjfalskjf

+ ghh

+ {}

+ f;dsljhl

+ df;shiose94p5er

+ stop()

+ stop( )

+

> #use esc button to get rid of perpetual plus sign

> dat <- data.frame(species = c("S.altissima", "S.rugosa", "E.graminifolia", "A.pilosus"), treatment = factor(c("Control", "Water", "Control", "Water")), height = c(1.1, 0.8, 0.9, 1) width = c(1, 1.7, 0.6, 0.2))

Error: unexpected symbol in "dat <- data.frame(species = c("S.altissima", "S.rugosa", "E.graminifolia", "A.pilosus"), treatment = factor(c("Control", "Water", "Control", "Water")), height = c(1.1, 0.8, 0.9, 1) width"

> dat<-data.frame(species=c("S.altissima", "S.rugosa", "E.graminifolia", "A.pilosus"),treatment=factor(c("Control", "Water","Control","Water")),height=c(1.1,0.8,0.9,1),width=c(1,1.7,0.6,0.2))

> dat

species treatment height width

1 S.altissima Control 1.1 1.0

2 S.rugosa Water 0.8 1.7

3 E.graminifolia Control 0.9 0.6

4 A.pilosus Water 1.0 0.2

> dat[2,]

species treatment height width

2 S.rugosa Water 0.8 1.7

> dat[3, 4]

[1] 0.6

> dat[,2] == "Water"

[1] FALSE TRUE FALSE TRUE

> dat[dat[, 2] == "Water", ]

species treatment height width

2 S.rugosa Water 0.8 1.7

4 A.pilosus Water 1.0 0.2

> subset(dat, treatment == "Water")

species treatment height width

2 S.rugosa Water 0.8 1.7

4 A.pilosus Water 1.0 0.2

> View(dat)

> #Done with 341-352