#alt+- assigned to (<- )

#c() - concatenate

#control+enter

#Data types (numeric, factor, integer,

#complex, character/string, logical/boolean)

#1. Numeric

numerics <- 10.5

numerics\_many <- c(10,11,12.5, 100)

#2. integer

integers <- c(1L, 10L)

#3. complex (imaginary part represented by i)

complexs <- c(9+3i)

#4. character/string, use speech marks (" ")

characters <- c("hello", "world", "in")

#for you to know the data type in R use the function class()

class(numerics\_many)

#5. logical/boolean

#this is the TRUE, FALSE

a <- 10

b <- 9

c <- 10

a>b

a<b

a == b

a == c

as.numeric() # converting data type to numeric

as.integer() #converting data type to integer

#we use ? to ask for help in R

#simple maths

#addition +

#subtraction -

#division /

#square root sqrt()

#summarise data summary()

#exponent ^

#multiplication \*

#modulus/absolute %%

#integer division %/%

x <- 1+2

y <- 1/3

sqrt(9)

summary(numerics\_many)

summary(x)

summary(y)

summary(c(1,2,3,4,5))

mean(numerics\_many)

median(numerics\_many)

#exponent

2^2

2^10

#multiplication

2\*8

#modulus

3 %% 16

3 %/% 16

#Data structures (vectors, lists, matrices, arrays, data frames)

#1. Vectors- list of items/variables that are of the same data type

# we use c(), separate items with ,

x1 <- c("apple", "oranges")

x2 <- c(1, 2, 3, 4, 5)

x3 <- 1:5

x4 <- 1.5:6.5 #R will use the default step as 1

x5 <- 1.5:6.2 #R will use the default step as 1

x6 <- seq(1.5, 6.2, by = 0.2) # we are able to define the step we want

#to know the length of a vector we use length()

length(x6)

# we use sort to arrange our items numerically/alphabetically

sort(x6)

x7 <- c(2,20,5,30,1,6,5,3,1,7,8)

sort(x7, decreasing = TRUE) # descending order

sort(x7, decreasing = FALSE) #ascending order

#access an item in vector we use []

x6[5]

x8 <- 65:1000

x8[500]

#lists - contain different data types

#we use list() to create

x9 <- list("apple", "banana", "kiwi")

x9

#to know the data structure we use class()

class(x9)

#to know the length of a list we use length()

length(x9)

#to access an item in a list we use []

x9[2]

#to check if an item is present we use %in%

"apple" %in% x9

"mango" %in% x9

#to add item(s) in lists we use append()

x10 <- append(x9, "mango", after = 1)

x11 <- append(x9, "mango")

#joining two lists or more we use c()

#join x9 and x10

x12 <- c(x9, x10)

#matrices - two dimensional data that has rows (horizontal)

#and columns (vertical), we use matrix(), we specify the rows with nrow

#and the columns with ncol

x13 <- matrix(c(1,2,3,4,5,6), nrow = 3, ncol = 2)

x14 <- matrix(c(1,2,3,4,5,6))

x15 <- matrix(c("apple", "banana", "cherry",

"kiwi", "oranges", "mangoes"), nrow = 3, ncol = 2)

#we can create matrices from vectors, but the vectors

#have to be of the same length

x16 <- c(1,2,3,4,5,6)

x17 <- c("apple", "banana", "cherry",

"kiwi", "oranges", "mangoes")

#combining using columns, use cbind()

x18 <- cbind(x16, x17)

#combining using rows, we use rbind()

x19 <- rbind(x16, x17)

#use ncol

x20 <- matrix(c(x18,x19),ncol = 3)

#check if we have an item in a matrix, we use %in%

5 %in% x14

#check the number of rows and colums of a matrix we use dim()

dim(x14)

dim(x13)

#to check the dimension of a matrix we use length()

length(x14)

length(x13)

#Data frame - data is displayed in form of a table

#it can contain different data types

#each column should have the same data type

#we use data.frame() to create

x21 <- data.frame(

fruits = c("apple", "banana", "cherry"),

counts = c(1,2,3),

duration = c(40,50,60)

)

#do some computation in dataframes

summary(x21)

length(x21) #this is for counting number of columns in a dataframe

table(x21$fruits, x21$counts)

#access an item in dataframe, we use $, [], [[]]

x21$fruits

x21$counts

c(x21$fruits, x21$counts)

#accessing the specific column

x21[3]

#accessing items in a specific column

x21[["fruits"]]

x21[["counts"]]

#Factors - used to categorize data, eg. demographic(male, female)

#we use factor() to create and as.factor() to convert a data structure to a facotr

x22 <- factor(c("male", "female"), levels = c("male", "female"))

#the number of items in a factor

length(x22)

#access an item in a factor

x22[1]