for print

"hello wolrd"

For Graph

plot(1:10)

for Variables

name <- "komal" print(name)

Assign one value to multiple variables

var1 <- var2 <- var3 <- "Orange"

Print variable values

var1 var2 var3

Data type

x <- 10.5 class(x)

y <- 55L class(y)

convert from integer to numeric:

a <- as.numeric(y) class(a)x

Built-in Math Functions

max(5, 10, 15)

min(5, 10, 15)

sqrt(4)

abs(-4.9)

ceiling(1.1)

floor(1.3)

String

str <- "Lorem ipsum dolor sit amet consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua." str

break line

cat(str)

string length

str <- "Hello World!" nchar(str)

characters are present in a string

str <- "Hello World!"

grepl("H", str) grepl("Hello", str) grepl("X", str)

Combine Two Strings

str1 <- "Hello" str2 <- "World"

paste(str1, str2)

Escape Characters in string

str <- "We are the so-called \"Vikings\", from the north."

str cat(str)

while loop

dice <- 1 while (dice <= 6) { if (dice < 6) { print("No Yahtzee") } else { print("Yahtzee!") } dice <- dice + 1 }

for loop

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

for (x in dice) { print(x) }

nested loop

adj <- list("red", "big", "tasty")

fruits <- list("apple", "banana", "cherry") for (x in adj) { for (y in fruits) { print(paste(x, y)) } }

Function

my_function <- function() { print("Hello World!") } my_function()

my_function <- function(fname) { paste(fname, "Griffin") } my_function("Peter") my_function("Lois") my_function("Stewie")

vector

A vector is simply a list of items that are of the same type. To combine the list of items to a vector, use the c() function and separate the items by a comma.

fruits <- c("banana", "apple", "orange")#String Vector fruits num <- c(1,2,3,4,5,6,7)#number vector num

Vector with numerical values in a sequence

numbers <- 1:10 numbers x<- 1.5:4.5 x

vector Length

length(x)

sorting Length

sort(fruits)

Access the first and third item

num[c(1, 6)]

remove element from particular position

num[c(-3)]

replace

fruits <- c("banana", "apple", "orange", "mango", "lemon")

replace "banana" to "pear"

fruits[1] <- "pear" fruits

Repeat each element

x<- rep(c(16,12,97), each =3) x

Repeat element independently

x<- rep(c(16,12,97), time= c(4,5,3)) x

Generating Sequence

numbers <- 1:10 numbers

Generating Sequence bigger to smaller

num <- seq(from=0, to =100, by = 20) num

List

vari<- list("komal",2,3,TRUE,"singh") vari

Access List element through it's index position

vari[6]

replace

vari[4]<- FALSE vari

check element present in the list or not

"komal" %in% vari 6 %in% vari

adding element

vari<- list("komal",2,3,TRUE,"singh") append(vari,"aman") append(vari,45) vari

vari<- list("komal",2,3,TRUE,"singh") append(vari, "rani", after = 2) vari

remove element

thislist <- list("apple", "banana", "cherry") newlist <- thislist[-1] newlist

thislist <- list("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango") (thislist)[2:5]

join List

list1 <- list("a", "b", "c") list2 <- list(1,2,3) list3 <- c(list1,list2) list3

Matrices

matri<- matrix(c(1,2,3,4,5,6,7,8,9,10),nrow =2, ncol = 5) matri matri1 <- matrix(c("apple", "banana", "cherry", "orange","grape", "pineapple", "pear", "melon", "fig"), nrow = 3, ncol = 3) matri1

Access Matrix Items

matri[2,3] matri[2,] matri1[c(1,2),]

Add column

matri2 <- cbind(matri, c(11,12)) matri2

Add row

matri3 <- rbind(matri, c(11,13,15,17,19)) matri3

Remove Rows and Columns

matri2 <- matri2[-c(1),-c(1)] matri2

Number of Rows and Columns

dim(matri3) dim(matri)

Matrix Length or total element present in matrix

length(matri) length(matri3)

Matrix1 <- matrix(c("apple", "banana", "cherry", "grape"), nrow = 2, ncol = 2) Matrix2 <- matrix(c("orange", "mango", "pineapple", "watermelon"), nrow = 2, ncol = 2) Matrix1 Matrix2 com1 <- rbind(Matrix1,Matrix2) com1 com <- cbind(Matrix1,Matrix2) com

array

thisarray <- c(1:24) thisarray multiarray <- array(thisarray, dim = c(4, 3, 2)) multiarray multiarray[2, 3, 2]

data frame

data1<- data.frame(x =c(1,2,3,4,5), y =c("abc","asd","xcv","qww","poi")) data1

add new column

newdata<- cbind(data1, z =c(12,13,14,15,16)) newdata

add row

newdata1<- rbind(newdata, z =c(6,"dfg",17)) newdata1

plot

plot(4:10)

plot with multiple point

plot(c(1, 2, 3, 4, 5), c(3, 7, 8, 9, 12))

Plot Labels

plot(1:10, main="My Graph", xlab="The x-axis", ylab="The y axis")

Colors: change the color of the points

plot(1:10, col="blue")

Size: change the size of the points

plot(1:10, cex=2)

Point Shape

pch parameter ranges from 0 to 25

plot(1:10, pch=11, cex=2)

draw a line

plot(1:10, type="l")

plot(1:10, type="l", lwd=5, lty=3)

Line Width, color, style

plot(1:10, type="l", col="blue", lwd=5, lty=3)

line1 <- c(1,2,3,4,5,10) line2 <- c(2,5,7,8,9,10)

plot(line1, type = "l", col = "blue") lines(line2, type="l", col = "red")

Scatter Plot

x1 <- c(5,7,8,7,2,2,9,4,11,12,9,6) y1 <- c(99,86,87,88,111,103,87,94,78,77,85,86)

plot(x1,y1)

day two, the age and speed of 15 cars:

x2 <- c(2,2,8,1,15,8,12,9,7,3,11,4,7,14,12) y2 <- c(100,105,84,105,90,99,90,95,94,100,79,112,91,80,85)

plot(x1, y1, main="Observation of Cars", xlab="Car age", ylab="Car speed", col="red", cex=2) points(x2, y2, col="blue", cex=2,pch=11)

pie plot

x <- c(10,20,30,40) pie(x)

Labels, color and Header of pie

x <- c(10,20,30,40)

Create a vector of labels

mylabel <- c("Apples", "Bananas", "Cherries", "Dates")

Display the pie chart with labels

pie(x, label = mylabel, main = "Fruits")

colors <- c("blue", "yellow", "green", "black")

Display the pie chart with colors

pie(x, label = mylabel, main = "Fruits", col = colors)

Create a vector of labels

Display the explanation box

legend("bottomright", mylabel, fill = colors)

bar plot

x <- c("A", "B", "C", "D")

y-axis values

y <- c(2, 4, 6, 8)

barplot(y, names.arg = x)

color, Horizontal and texture

x <- c("A", "B", "C", "D") y <- c(2, 4, 6, 8) barplot(y, names.arg = x, density = 10,col = "red", horiz = TRUE)

inbuilt data set

x<-mtcars dim(x)

about data set

?mtcars

get the name of each row in the first column

rownames(x)

max and min

Data_Cars <- mtcars

max(Data_Cars$hp) min(Data_Cars$hp)

rownames(Data_Cars)[which.max(Data_Cars$hp)] rownames(Data_Cars)[which.min(Data_Cars$hp)]

mean

mean(Data_Cars$wt)

median

median(Data_Cars$wt)

mode

names(sort(-table(Data_Cars$wt)))[1]

Percentiles

quantile(Data_Cars$wt, c(0.75))

quantile(Data_Cars$wt)