Aman Raj-200103020(DSA Individual Assignment)

Aman Raj

12/18/2020

## 

## Assignment and Mathematical Operations

#Creating Simple Objects and Doing Mathematical Calculation  
  
a = 5 #Value 5 is assigned to Variable a  
a

## [1] 5

b = 10  
b

## [1] 10

class(a) # DataType of variable a

## [1] "numeric"

a = "Hello"  
class(a)

## [1] "character"

a = TRUE  
class(a)

## [1] "logical"

a = FALSE  
class (a)

## [1] "logical"

#Object Assignments and Simple Calculations  
x = 10  
y = 15  
x+y #Sum of x and y

## [1] 25

x-y

## [1] -5

x\*y

## [1] 150

x/y

## [1] 0.6666667

sqrt(x)

## [1] 3.162278

x^y

## [1] 1e+15

exp(x)

## [1] 22026.47

log(x, base=exp(1))

## [1] 2.302585

log10(x)

## [1] 1

help("log") #Utilizing R Help

## starting httpd help server ... done

factorial(x)

## [1] 3628800

cos(x)

## [1] -0.8390715

abs(x)

## [1] 10

## 

## FUNCTIONS

getwd() #Get Working Directory

## [1] "C:/Users/Pankaj/Documents/IMT-G/TERM 2/R PROGRAMMING"

# Functions in R  
divider = function(x,y) {  
 result = x/y  
 print(result)  
}  
divider(50,25)

## [1] 2

divider (100,25)

## [1] 4

# Multiplication  
multiply = function(a,b){  
 return (a \* b) #Directly returns the value  
}  
multiply(23,25)

## [1] 575

multiply (19,20)

## [1] 380

#Variables Names are CASE SENSITIVE  
A=10  
a=24  
  
A #Prints value for A

## [1] 10

#CONCATENATION AND ARRAYS  
  
f <- c(1,2,3,4,5)  
f = c(1,2,3,4,5)  
f

## [1] 1 2 3 4 5

f+4 #Adds 4 to each element

## [1] 5 6 7 8 9

d = f / 4  
d

## [1] 0.25 0.50 0.75 1.00 1.25

f+d

## [1] 1.25 2.50 3.75 5.00 6.25

## 

## DATA TYPES-R

x = 10  
class(x)

## [1] "numeric"

# Numeric - Integer and Decimal - (R)- Integer (Whole Number) and Numeric (Float - Decimal)  
i = 5L # L - Integer  
class(i)

## [1] "integer"

is.integer(i)

## [1] TRUE

is.numeric(x)

## [1] TRUE

# Logical - TRUE (1) and FALSE (0)  
#R understands value of TRUE as 1 and FALSE as 0  
TRUE \* 5

## [1] 5

FALSE \* 5

## [1] 0

K = TRUE  
class(K)

## [1] "logical"

is.logical(K)

## [1] TRUE

# Date - Starting Date (1970) - Numeric Value.  
# In R - 1 Jan 1970  
# Date - mm/dd/yyyy  
# POSIXct - Date plus Time.  
  
date1 = as.Date("2012-06-28")  
# as.Date()# Auto complete # How to enter  
# ? as.Date # help  
date1

## [1] "2012-06-28"

class (date1)

## [1] "Date"

as.numeric(date1)

## [1] 15519

#POSIXct - Date and Time  
date2 = as.POSIXct("2012-06-28 17:42")  
date2

## [1] "2012-06-28 17:42:00 IST"

class(date2)

## [1] "POSIXct" "POSIXt"

as.numeric(date2)

## [1] 1340885520

## 

## VECTORS

v = c(1,2,3,4,5)  
s = v\*2  
s

## [1] 2 4 6 8 10

#Vector Operation  
d = v-2  
d

## [1] -1 0 1 2 3

f = v /2  
f

## [1] 0.5 1.0 1.5 2.0 2.5

sqrt(f)

## [1] 0.7071068 1.0000000 1.2247449 1.4142136 1.5811388

numb = c(7,8,9,3,4)  
numb

## [1] 7 8 9 3 4

length(numb) #Size of the vector

## [1] 5

numb[c(1,3)] #Access 1st and 3rd element

## [1] 7 9

numb = c(7,8,9,3,4)  
  
numb[5]=10 #Assigns 10 to 5th position  
  
numb

## [1] 7 8 9 3 10

sort(numb) #Arranges in ascending order

## [1] 3 7 8 9 10

#Give Names to Vector!  
c(One = "a", Two = "y", Last = "r") # Name-Value pair

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

w = 1:3 #numbers 1 to 3 is assigned to w  
names(w) = c("a","b","c")  
w

## a b c   
## 1 2 3

## 

## DATAFRAMES

x = 10:1  
y = -4:5  
q = c("Hockey","Football","Baseball","Curlin","Rugby","Lacrosse",  
 "Basketball","Tennis","Cricket","Soccer")  
theDF = data.frame(x,y,q) # this would create a 10x3 data.frame with x, y and q as variable names  
theDF

## x y q  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer

# Checking the dimensions of the DF.  
nrow(theDF)

## [1] 10

ncol(theDF)

## [1] 3

dim(theDF)

## [1] 10 3

names (theDF)

## [1] "x" "y" "q"

names(theDF)[3]

## [1] "q"

rownames(theDF)

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

# Head and Tail  
head(theDF)

## x y q  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse

head(theDF, n=7)

## x y q  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball

tail(theDF)

## x y q  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer

class(theDF)

## [1] "data.frame"

# Assigning Names  
theDF = data.frame (First=x, Second =y, Sport = q)  
theDF

## First Second Sport  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer

# Accessing Individual Column using $  
  
theDF$Sport # gives the third column named Sport

## [1] "Hockey" "Football" "Baseball" "Curlin" "Rugby"   
## [6] "Lacrosse" "Basketball" "Tennis" "Cricket" "Soccer"

# Accessing Specific row and column  
theDF[3,2] # 3rd row and 2nd Column

## [1] -2

theDF[3,2:3] # 3rd Row and column 2 thru 3

## Second Sport  
## 3 -2 Baseball

theDF[c(3,5), 2]# Row 3&5 from Column 2;

## [1] -2 0

# since only one column was selected, it was returned as vector and hence no column names in output.  
  
# Rows 3&5 and Columns 2 through 3  
theDF[c(3,5), 2:3]

## Second Sport  
## 3 -2 Baseball  
## 5 0 Rugby

theDF[ ,3] # Access all Rows for column 3

## [1] "Hockey" "Football" "Baseball" "Curlin" "Rugby"   
## [6] "Lacrosse" "Basketball" "Tennis" "Cricket" "Soccer"

theDF[ , 2:3]

## Second Sport  
## 1 -4 Hockey  
## 2 -3 Football  
## 3 -2 Baseball  
## 4 -1 Curlin  
## 5 0 Rugby  
## 6 1 Lacrosse  
## 7 2 Basketball  
## 8 3 Tennis  
## 9 4 Cricket  
## 10 5 Soccer

theDF[2,]# Access all columns for Row 2

## First Second Sport  
## 2 9 -3 Football

theDF[2:4,]

## First Second Sport  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin

#Another way to create data frames  
custData = data.frame(name=c("Tom", "Sally", "Sue"),  
 age=c(43, 28, 42),  
 stringsAsFactors=T)  
  
custData

## name age  
## 1 Tom 43  
## 2 Sally 28  
## 3 Sue 42

## 

## FACTORS

#Create a factor vector  
q2 = c(q,"Hockey","Lacrosse","Hockey","Water Polo","Hockey","Lacrosse")  
q2

## [1] "Hockey" "Football" "Baseball" "Curlin" "Rugby"   
## [6] "Lacrosse" "Basketball" "Tennis" "Cricket" "Soccer"   
## [11] "Hockey" "Lacrosse" "Hockey" "Water Polo" "Hockey"   
## [16] "Lacrosse"

class(q2)

## [1] "character"

as.numeric(q2)

## Warning: NAs introduced by coercion

## [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA

class(q2)

## [1] "character"

#Creating another factor vector  
direction = c("Up", "Down", "Left", "Right", "Left", "Up")  
factorDir = factor(direction)  
factorDir

## [1] Up Down Left Right Left Up   
## Levels: Down Left Right Up

is.factor(factorDir) #Checks if it is a factor

## [1] TRUE

is.factor(direction)

## [1] FALSE

factorDir #Only unique values

## [1] Up Down Left Right Left Up   
## Levels: Down Left Right Up

# A Factor object contains levels which store all possible  
# values  
levels(x=factorDir)

## [1] "Down" "Left" "Right" "Up"

# You can define your levels and their orders  
dow = c("Monday", "Tuesday", "Wednesday", "Thursday",  
 "Friday", "Saturday", "Sunday")  
  
wDays = c("Tuesday", "Thursday", "Monday")  
  
wdFact = factor(x=wDays, levels=dow, ordered=T)  
  
wdFact

## [1] Tuesday Thursday Monday   
## 7 Levels: Monday < Tuesday < Wednesday < Thursday < Friday < ... < Sunday

## 

## MATRICES

# Create a Matrix with a single column  
matrix1 = matrix(data=c(1,2,3,4))  
matrix1

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

# Create a matrix with defined rows and columns  
matrix2 = matrix(data=c(1,2,3,4), nrow=2, ncol=2)  
matrix2

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

# You can also fill by row (You can use T or TRUE)  
matrix3 = matrix(data=c(1,2,3,4), nrow=2, ncol=2, byrow=T)  
matrix3

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

# Get a Matrix dimension  
dim(matrix3)

## [1] 2 2

# A value at row, column  
matrix3[1,2]

## [1] 2

# Get a whole row  
matrix3[1,]

## [1] 1 2

# Get a whole column  
matrix3[,2]

## [1] 2 4

# Combine vectors to make a Matrix  
matrix4 = rbind(1:3, 4:6, 7:9)  
matrix4

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

# Get 2nd and 3rd row  
matrix4[2:3,]

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

# Get 2nd and 3rd row by ommitting the 1st  
matrix4[-1,]

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

# Change the first value  
matrix4[1,1] = 0  
matrix4

## [,1] [,2] [,3]  
## [1,] 0 2 3  
## [2,] 4 5 6  
## [3,] 7 8 9

# Change the 1st row  
matrix4[1,] = c(10,11,12)  
matrix4

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

#creating few more matrices  
A = matrix(1:10, nrow=5)# Create a 5x2 matrix  
B = matrix(21:30, nrow=5)#Create another 5x2 matrix   
C = matrix (21:40, nrow=2)#Create another 2x10 matrix  
D = matrix(41:45, ncol = 5) #Creates 1\*5 matrix  
D

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 41 42 43 44 45

A

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

B

## [,1] [,2]  
## [1,] 21 26  
## [2,] 22 27  
## [3,] 23 28  
## [4,] 24 29  
## [5,] 25 30

C

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]  
## [1,] 21 23 25 27 29 31 33 35 37 39  
## [2,] 22 24 26 28 30 32 34 36 38 40

nrow(A) #Gives the no. of rows of A

## [1] 5

ncol(A)

## [1] 2

dim(A) #Gives the dimensions of A

## [1] 5 2

# Add the values of the matrices A and B  
A+B

## [,1] [,2]  
## [1,] 22 32  
## [2,] 24 34  
## [3,] 26 36  
## [4,] 28 38  
## [5,] 30 40

# Multiply Them (Vector Multiplication!)  
A

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

B

## [,1] [,2]  
## [1,] 21 26  
## [2,] 22 27  
## [3,] 23 28  
## [4,] 24 29  
## [5,] 25 30

A\*B # A = 5x2 and B = 5x2

## [,1] [,2]  
## [1,] 21 156  
## [2,] 44 189  
## [3,] 69 224  
## [4,] 96 261  
## [5,] 125 300

#See if the elements are equal  
A == B

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

# Matrix Multiplication(MM. A is 5x2. B is 5x2. B-transpose is 2x5  
A %\*% t(B)

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 177 184 191 198 205  
## [2,] 224 233 242 251 260  
## [3,] 271 282 293 304 315  
## [4,] 318 331 344 357 370  
## [5,] 365 380 395 410 425

# Naming the Columns and Rows   
colnames(A)

## NULL

rownames(A)

## NULL

colnames(A)= c("Left","Right")  
rownames(A)= c("1st","2nd","3rd","4th","5th")  
colnames(B)

## NULL

rownames(B)

## NULL

colnames(B)= c("First","Second")  
rownames(B)= c("One","Two","Three","Four","Five")  
colnames(C)

## NULL

rownames(C)

## NULL

colnames(C) = LETTERS [1:10]  
rownames(C) = c("Top", "Bottom")  
  
# Matrix Multiplication. A is 5x2 and C is 2x10  
dim(A)

## [1] 5 2

dim(C)

## [1] 2 10

t(A)

## 1st 2nd 3rd 4th 5th  
## Left 1 2 3 4 5  
## Right 6 7 8 9 10

A %\*% C

## A B C D E F G H I J  
## 1st 153 167 181 195 209 223 237 251 265 279  
## 2nd 196 214 232 250 268 286 304 322 340 358  
## 3rd 239 261 283 305 327 349 371 393 415 437  
## 4th 282 308 334 360 386 412 438 464 490 516  
## 5th 325 355 385 415 445 475 505 535 565 595

## 

## ARRAYS

theArray = array(1:12, dim=c(2,3,2))# Total Elements = R x C x OD  
theArray

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

theArray [1, ,]# Accessing all elements from Row 1, all columns, all outer dimensions & build C x OD (R x C)

## [,1] [,2]  
## [1,] 1 7  
## [2,] 3 9  
## [3,] 5 11

theArray[1, ,1]# Accessing all elements from Row 1, all columns, first outer dimension

## [1] 1 3 5

theArray[, ,1]# Accessing all rows, all columns, first outer dimension

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

# Array with Four Outer Dimensions (OD)  
theArray\_4D = array(1:32, dim=c(2,4,4))  
theArray\_4D

## , , 1  
##   
## [,1] [,2] [,3] [,4]  
## [1,] 1 3 5 7  
## [2,] 2 4 6 8  
##   
## , , 2  
##   
## [,1] [,2] [,3] [,4]  
## [1,] 9 11 13 15  
## [2,] 10 12 14 16  
##   
## , , 3  
##   
## [,1] [,2] [,3] [,4]  
## [1,] 17 19 21 23  
## [2,] 18 20 22 24  
##   
## , , 4  
##   
## [,1] [,2] [,3] [,4]  
## [1,] 25 27 29 31  
## [2,] 26 28 30 32

theArray\_4D [1, ,]

## [,1] [,2] [,3] [,4]  
## [1,] 1 9 17 25  
## [2,] 3 11 19 27  
## [3,] 5 13 21 29  
## [4,] 7 15 23 31

theArray\_4D[1, ,1]

## [1] 1 3 5 7

theArray[, ,1]

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

## 

## LIST

list(1,2,3)# creates a three element list

## [[1]]  
## [1] 1  
##   
## [[2]]  
## [1] 2  
##   
## [[3]]  
## [1] 3

list(c(1,2,3))# creates a single element(vector with three elements)

## [[1]]  
## [1] 1 2 3

list3 = list(c(1,2,3), 3:7)# create two element list  
# first is three elements vector, next is five element vector.  
list3

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

# The same can be written as  
(list3 = list(c(1,2,3), 3:7))

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

# Two Element list  
# First element is data.frame and next is 10 element vector  
list(theDF, 1:10)# theDF is already created in previous exercise!

## [[1]]  
## First Second Sport  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer  
##   
## [[2]]  
## [1] 1 2 3 4 5 6 7 8 9 10

# Three element list  
list5 = list(theDF, 1:10, list3)  
list5

## [[1]]  
## First Second Sport  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer  
##   
## [[2]]  
## [1] 1 2 3 4 5 6 7 8 9 10  
##   
## [[3]]  
## [[3]][[1]]  
## [1] 1 2 3  
##   
## [[3]][[2]]  
## [1] 3 4 5 6 7

#Naming List (similar to column name in data.frame)   
names(list5)= c("data.frame", "vector","list")  
names(list5)

## [1] "data.frame" "vector" "list"

list5

## $data.frame  
## First Second Sport  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer  
##   
## $vector  
## [1] 1 2 3 4 5 6 7 8 9 10  
##   
## $list  
## $list[[1]]  
## [1] 1 2 3  
##   
## $list[[2]]  
## [1] 3 4 5 6 7

#Naming using "Name-Value" pair  
list6 = list(TheDataFrame = theDF, TheVector = 1:10, TheList = list3)  
names(list6)

## [1] "TheDataFrame" "TheVector" "TheList"

list6

## $TheDataFrame  
## First Second Sport  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer  
##   
## $TheVector  
## [1] 1 2 3 4 5 6 7 8 9 10  
##   
## $TheList  
## $TheList[[1]]  
## [1] 1 2 3  
##   
## $TheList[[2]]  
## [1] 3 4 5 6 7

# Creating an empty list  
(emptylist = vector(mode="list", length =4))

## [[1]]  
## NULL  
##   
## [[2]]  
## NULL  
##   
## [[3]]  
## NULL  
##   
## [[4]]  
## NULL

# Accessing individual element of a list - Double Square Brackets  
# specify either element number or name  
list5[[1]]

## First Second Sport  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer

list5[["data.frame"]]

## First Second Sport  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer

list5[[1]]$Sport

## [1] "Hockey" "Football" "Baseball" "Curlin" "Rugby"   
## [6] "Lacrosse" "Basketball" "Tennis" "Cricket" "Soccer"

list5[[1]][,"Second"]

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

list5[[1]][,"Second", drop = FALSE]

## Second  
## 1 -4  
## 2 -3  
## 3 -2  
## 4 -1  
## 5 0  
## 6 1  
## 7 2  
## 8 3  
## 9 4  
## 10 5

# LENGTH OF LIST  
length(list5)

## [1] 3

names(list5)

## [1] "data.frame" "vector" "list"

list5

## $data.frame  
## First Second Sport  
## 1 10 -4 Hockey  
## 2 9 -3 Football  
## 3 8 -2 Baseball  
## 4 7 -1 Curlin  
## 5 6 0 Rugby  
## 6 5 1 Lacrosse  
## 7 4 2 Basketball  
## 8 3 3 Tennis  
## 9 2 4 Cricket  
## 10 1 5 Soccer  
##   
## $vector  
## [1] 1 2 3 4 5 6 7 8 9 10  
##   
## $list  
## $list[[1]]  
## [1] 1 2 3  
##   
## $list[[2]]  
## [1] 3 4 5 6 7

## 

## READ AND WRITE FILES

myPeople = read.table("People.txt.txt",  
 header=T, sep=" ",  
 na.strings="`",  
 stringsAsFactors=F)  
myPeople

## fname lname sex  
## 1 aman raj M  
## 2 ms dhoni M  
## 3 Anushka Sharma F  
## 4 aviral saxena M  
## 5 Ayush Smith M  
## 6 Anshika Jain F  
## 7 Rahul Dravid M  
## 8 Rahul Dravid M  
## 9 Rahul Dravid M  
## 10 Rahul Dravid M  
## 11 Rahul Dravid M

#Reading the files   
  
# Add another person  
addname = data.frame(fname="Rahul",  
 lname="Dravid",  
 sex="M")  
myPeople = rbind(myPeople, addname)  
myPeople

## fname lname sex  
## 1 aman raj M  
## 2 ms dhoni M  
## 3 Anushka Sharma F  
## 4 aviral saxena M  
## 5 Ayush Smith M  
## 6 Anshika Jain F  
## 7 Rahul Dravid M  
## 8 Rahul Dravid M  
## 9 Rahul Dravid M  
## 10 Rahul Dravid M  
## 11 Rahul Dravid M  
## 12 Rahul Dravid M

# Update a record  
myPeople[5,2] = "Smith"  
myPeople

## fname lname sex  
## 1 aman raj M  
## 2 ms dhoni M  
## 3 Anushka Sharma F  
## 4 aviral saxena M  
## 5 Ayush Smith M  
## 6 Anshika Jain F  
## 7 Rahul Dravid M  
## 8 Rahul Dravid M  
## 9 Rahul Dravid M  
## 10 Rahul Dravid M  
## 11 Rahul Dravid M  
## 12 Rahul Dravid M

# Update the file by supplying the data.frame,  
# the file to write, seperator, na, whether to  
# quote strings, whether to include row numbers  
write.table(x=myPeople, "People.txt.txt",  
 sep=" ", na="`",  
 quote=F, row.names=F)  
  
# Get 1st 3 records  
head(myPeople, 3)

## fname lname sex  
## 1 aman raj M  
## 2 ms dhoni M  
## 3 Anushka Sharma F

# Get remaining records  
tail(myPeople, 3)

## fname lname sex  
## 10 Rahul Dravid M  
## 11 Rahul Dravid M  
## 12 Rahul Dravid M

## 

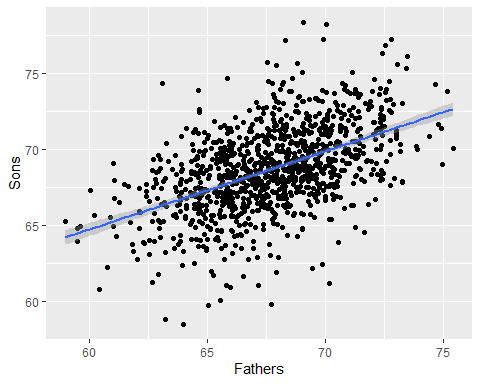
## LINEAR REGRESSION

# Simple Linear Regression (SLR)  
# Dataset: father.son.   
# Using fathers' heights to predit sons' heights using SLR.  
# Fathers height as predictor(Indep - X) and   
# Son's height as the response /Target(Dep - Y)  
require(UsingR)  
require(ggplot2)  
head(father.son)

## fheight sheight  
## 1 65.04851 59.77827  
## 2 63.25094 63.21404  
## 3 64.95532 63.34242  
## 4 65.75250 62.79238  
## 5 61.13723 64.28113  
## 6 63.02254 64.24221

ggplot(father.son, aes(x=fheight, y=sheight))+geom\_point()+  
 geom\_smooth(method="lm")+labs(x="Fathers", y="Sons")

## `geom\_smooth()` using formula 'y ~ x'



heightsLM = lm(sheight ~ fheight, data = father.son)  
heightsLM

##   
## Call:  
## lm(formula = sheight ~ fheight, data = father.son)  
##   
## Coefficients:  
## (Intercept) fheight   
## 33.8866 0.5141

summary(heightsLM)

##   
## Call:  
## lm(formula = sheight ~ fheight, data = father.son)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.8772 -1.5144 -0.0079 1.6285 8.9685   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 33.88660 1.83235 18.49 <2e-16 \*\*\*  
## fheight 0.51409 0.02705 19.01 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.437 on 1076 degrees of freedom  
## Multiple R-squared: 0.2513, Adjusted R-squared: 0.2506   
## F-statistic: 361.2 on 1 and 1076 DF, p-value: < 2.2e-16