[12:43 PM, 1/19/2019] Saket Vikas Sir Madrid: vector1 <- c(5,9,3)

vector2 <- c(10,11,12,13,14,15)

# Take these vectors as input to the array.

result <- array(c(vector1,vector2),dim = c(3,3,2))

print(result)

[12:43 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create two vectors of different lengths.

vector1 <- c(5,9,3)

vector2 <- c(10,11,12,13,14,15)

column.names <- c("COL1","COL2","COL3")

row.names <- c("ROW1","ROW2","ROW3")

matrix.names <- c("Matrix1","Matrix2")

# Take these vectors as input to the array.

result <- array(c(vector1,vector2),dim = c(3,3,2),dimnames = list(row.names,column.names,

matrix.names))

print(result)

[12:43 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create two vectors of different lengths.

vector1 <- c(5,9,3)

vector2 <- c(10,11,12,13,14,15)

column.names <- c("COL1","COL2","COL3")

row.names <- c("ROW1","ROW2","ROW3")

matrix.names <- c("Matrix1","Matrix2")

# Take these vectors as input to the array.

result <- array(c(vector1,vector2),dim = c(3,3,2),dimnames = list(row.names,

column.names, matrix.names))

# Print the third row of the second matrix of the array.

print(result[3,,2])

# Print the element in the 1st row and 3rd column of the 1st matrix.

print(result[1,3,1])

# Print the 2nd Matrix.

print(result[,,2])

[12:43 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create two vectors of different lengths.

vector1 <- c(5,9,3)

vector2 <- c(10,11,12,13,14,15)

# Take these vectors as input to the array.

array1 <- array(c(vector1,vector2),dim = c(3,3,2))

# Create two vectors of different lengths.

vector3 <- c(9,1,0)

vector4 <- c(6,0,11,3,14,1,2,6,9)

array2 <- array(c(vector1,vector2),dim = c(3,3,2))

# create matrices from these arrays.

matrix1 <- array1[,,2]

matrix2 <- array2[,,2]

# Add the matrices.

result <- matrix1+matrix2

print(result)

[12:43 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create two vectors of different lengths.

vector1 <- c(5,9,3)

vector2 <- c(10,11,12,13,14,15)

# Take these vectors as input to the array.

new.array <- array(c(vector1,vector2),dim = c(3,3,2))

print(new.array)

# Use apply to calculate the sum of the rows across all the matrices.

result <- apply(new.array, c(1), sum)

print(result)

[12:45 PM, 1/19/2019] Saket Vikas Sir Madrid: this is a list practice set

[12:45 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create a list containing strings, numbers, vectors and a logical

# values.

list\_data <- list("Red", "Green", c(21,32,11), TRUE, 51.23, 119.1)

print(list\_data)

[12:45 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create a list containing a vector, a matrix and a list.

list\_data <- list(c("Jan","Feb","Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),

list("green",12.3))

# Give names to the elements in the list.

names(list\_data) <- c("1st Quarter", "A\_Matrix", "A Inner list")

# Show the list.

print(list\_data)

[12:45 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create a list containing a vector, a matrix and a list.

list\_data <- list(c("Jan","Feb","Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),

list("green",12.3))

# Give names to the elements in the list.

names(list\_data) <- c("1st Quarter", "A\_Matrix", "A Inner list")

# Access the first element of the list.

print(list\_data[1])

# Access the thrid element. As it is also a list, all its elements will be printed.

print(list\_data[3])

# Access the list element using the name of the element.

print(list\_data$A\_Matrix)

[12:46 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create a list containing a vector, a matrix and a list.

list\_data <- list(c("Jan","Feb","Mar"), matrix(c(3,9,5,1,-2,8), nrow = 2),

list("green",12.3))

# Give names to the elements in the list.

names(list\_data) <- c("1st Quarter", "A\_Matrix", "A Inner list")

# Add element at the end of the list.

list\_data[4] <- "New element"

print(list\_data[4])

# Remove the last element.

list\_data[4] <- NULL

# Print the 4th Element.

print(list\_data[4])

# Update the 3rd Element.

list\_data[3] <- "updated element"

print(list\_data[3])

[12:46 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create two lists.

list1 <- list(1,2,3)

list2 <- list("Sun","Mon","Tue")

# Merge the two lists.

merged.list <- c(list1,list2)

# Print the merged list.

print(merged.list)

[12:46 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create lists.

list1 <- list(1:5)

print(list1)

list2 <-list(10:14)

print(list2)

# Convert the lists to vectors.

v1 <- unlist(list1)

v2 <- unlist(list2)

print(v1)

print(v2)

# Now add the vectors

result <- v1+v2

print(result)

[12:47 PM, 1/19/2019] Saket Vikas Sir Madrid: this is matrix example

[12:47 PM, 1/19/2019] Saket Vikas Sir Madrid: ...............................................................................................................................................................................................................................

[12:47 PM, 1/19/2019] Saket Vikas Sir Madrid: # Elements are arranged sequentially by row.

M <- matrix(c(3:14), nrow = 4, byrow = TRUE)

print(M)

# Elements are arranged sequentially by column.

N <- matrix(c(3:14), nrow = 4, byrow = FALSE)

print(N)

# Define the column and row names.

rownames = c("row1", "row2", "row3", "row4")

colnames = c("col1", "col2", "col3")

P <- matrix(c(3:14), nrow = 4, byrow = TRUE, dimnames = list(rownames, colnames))

print(P)

[12:47 PM, 1/19/2019] Saket Vikas Sir Madrid: # Define the column and row names.

rownames = c("row1", "row2", "row3", "row4")

colnames = c("col1", "col2", "col3")

# Create the matrix.

P <- matrix(c(3:14), nrow = 4, byrow = TRUE, dimnames = list(rownames, colnames))

# Access the element at 3rd column and 1st row.

print(P[1,3])

# Access the element at 2nd column and 4th row.

print(P[4,2])

# Access only the 2nd row.

print(P[2,])

# Access only the 3rd column.

print(P[,3])

[12:47 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create two 2x3 matrices.

matrix1 <- matrix(c(3, 9, -1, 4, 2, 6), nrow = 2)

print(matrix1)

matrix2 <- matrix(c(5, 2, 0, 9, 3, 4), nrow = 2)

print(matrix2)

# Add the matrices.

result <- matrix1 + matrix2

cat("Result of addition","\n")

print(result)

# Subtract the matrices

result <- matrix1 - matrix2

cat("Result of subtraction","\n")

print(result)

[12:47 PM, 1/19/2019] Saket Vikas Sir Madrid: # Create two 2x3 matrices.

matrix1 <- matrix(c(3, 9, -1, 4, 2, 6), nrow = 2)

print(matrix1)

matrix2 <- matrix(c(5, 2, 0, 9, 3, 4), nrow = 2)

print(matrix2)

# Multiply the matrices.

result <- matrix1 \* matrix2

cat("Result of multiplication","\n")

print(result)

# Divide the matrices

result <- matrix1 / matrix2

cat("Result of division","\n")

print(result)

[1:00 PM, 1/19/2019] Saket Vikas Sir Madrid: Create Data Frame

[1:00 PM, 1/19/2019] Saket Vikas Sir Madrid: emp\_id = c (1:5),

emp\_name = c("Rick","Dan","Michelle","Ryan","Gary"),

salary = c(623.3,515.2,611.0,729.0,843.25),

start\_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",

"2015-03-27"))

[1:00 PM, 1/19/2019] Saket Vikas Sir Madrid: Get the Structure of the Data Frame

[1:01 PM, 1/19/2019] Saket Vikas Sir Madrid: str(emp.data)

[1:01 PM, 1/19/2019] Saket Vikas Sir Madrid: emp.data <- data.frame(

emp\_id = c (1:5),

emp\_name = c("Rick","Dan","Michelle","Ryan","Gary"),

salary = c(623.3,515.2,611.0,729.0,843.25),

start\_date = as.Date(c("2012-01-01","2013-09-23","2014-11-15","2014-05-11",

"2015-03-27")),

stringsAsFactors = FALSE

)

# Extract Specific columns.

result <- data.frame(emp.data$emp\_name,emp.data$salary)

print(result)

[1:03 PM, 1/19/2019] Saket Vikas Sir Madrid: emp.data <- data.frame(

emp\_id = c (1:5),

emp\_name = c("Rick","Dan","Michelle","Ryan","Gary"),

salary = c(623.3,515.2,611.0,729.0,843.25),

start\_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",

"2015-03-27")),

stringsAsFactors = FALSE

)

# Extract first two rows.

result <- emp.data[1:2,]

print(result)

[1:04 PM, 1/19/2019] Saket Vikas Sir Madrid: Extract 3rd and 5th row with 2nd and 4th column

Live Demo

# Create the data frame.

emp.data <- data.frame(

emp\_id = c (1:5),

emp\_name = c("Rick","Dan","Michelle","Ryan","Gary"),

salary = c(623.3,515.2,611.0,729.0,843.25),

start\_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",

"2015-03-27")),

stringsAsFactors = FALSE

)

# Extract 3rd and 5th row with 2nd and 4th column.

result <- emp.data[c(3,5),c(2,4)]

print(result)

[1:04 PM, 1/19/2019] Saket Vikas Sir Madrid: Add Row

To add more rows permanently to an existing data frame, we need to bring in the new rows in the same structure as the existing data frame and use the rbind() function.

In the example below we create a data frame with new rows and merge it with the existing data frame to create the final data frame.

Live Demo

# Create the first data frame.

emp.data <- data.frame(

emp\_id = c (1:5),

emp\_name = c("Rick","Dan","Michelle","Ryan","Gary"),

salary = c(623.3,515.2,611.0,729.0,843.25),

start\_date = as.Date(c("2012-01-01", "2013-09-23", "2014-11-15", "2014-05-11",

"2015-03-27")),

dept = c("IT","Operations","IT","HR","Finance"),

stringsAsFactors = FALSE

)

# Create the second data frame

emp.newdata <- data.frame(

emp\_id = c (6:8),

emp\_name = c("Rasmi","Pranab","Tusar"),

salary = c(578.0,722.5,632.8),

start\_date = as.Date(c("2013-05-21","2013-07-30","2014-06-17")),

dept = c("IT","Operations","Fianance"),

stringsAsFactors = FALSE

)

# Bind the two data frames.

emp.finaldata <- rbind(emp.data,emp.newdata)

print(emp.finaldata)

[12:48 PM, 1/20/2019] hansee316: Hi

[12:49 PM, 1/20/2019] Saket Vikas Sir Madrid: # Create data for the graph.

x <- c(21, 62, 10, 53)

labels <- c("London", "New York", "Singapore", "Mumbai")

# Give the chart file a name.

png(file = "city.jpg")

# Plot the chart.

pie(x,labels)

[12:49 PM, 1/20/2019] Saket Vikas Sir Madrid: Example

The below script will create and save the pie chart in the current R working directory.

Live Demo

# Create data for the graph.

x <- c(21, 62, 10, 53)

labels <- c("London", "New York", "Singapore", "Mumbai")

# Give the chart file a name.

png(file = "city\_title\_colours.jpg")

# Plot the chart with title and rainbow color pallet.

pie(x, labels, main = "City pie chart", col = rainbow(length(x)))

[12:49 PM, 1/20/2019] Saket Vikas Sir Madrid: Slice Percentages and Chart Legend

We can add slice percentage and a chart legend by creating additional chart variables.

Live Demo

# Create data for the graph.

x <- c(21, 62, 10,53)

labels <- c("London","New York","Singapore","Mumbai")

piepercent<- round(100\*x/sum(x), 1)

# Give the chart file a name.

png(file = "city\_percentage\_legends.jpg")

# Plot the chart.

pie(x, labels = piepercent, main = "City pie chart",col = rainbow(length(x)))

legend("topright", c("London","New York","Singapore","Mumbai"), cex = 0.8,

fill = rainbow(length(x)))

[12:50 PM, 1/20/2019] Saket Vikas Sir Madrid: bar char...................................................................................................................................................................................................................................

[12:56 PM, 1/20/2019] Saket Vikas Sir Madrid: H <- c(7,12,28,3,41)

# Give the chart file a name

png(file = "barchart.png")

# Plot the bar chart

barplot(H)

[12:57 PM, 1/20/2019] Saket Vikas Sir Madrid: # Create the data for the chart

H <- c(7,12,28,3,41)

M <- c("Mar","Apr","May","Jun","Jul")

# Give the chart file a name

png(file = "barchart\_months\_revenue.png")

# Plot the bar chart

barplot(H,names.arg=M,xlab="Month",ylab="Revenue",col="blue",

main="Revenue chart",border="red")

[12:57 PM, 1/20/2019] Saket Vikas Sir Madrid: # Create the input vectors.

colors = c("green","orange","brown")

months <- c("Mar","Apr","May","Jun","Jul")

regions <- c("East","West","North")

# Create the matrix of the values.

Values <- matrix(c(2,9,3,11,9,4,8,7,3,12,5,2,8,10,11), nrow = 3, ncol = 5, byrow = TRUE)

# Give the chart file a name

png(file = "barchart\_stacked.png")

# Create the bar chart

barplot(Values, main = "total revenue", names.arg = months, xlab = "month", ylab = "revenue", col = colors)

# Add the legend to the chart

legend("topleft", regions, cex = 1.3, fill = colors)

[12:58 PM, 1/20/2019] Saket Vikas Sir Madrid: box plot..........................................................................................................................................................

[12:58 PM, 1/20/2019] Saket Vikas Sir Madrid: Boxplot with Notch

We can draw boxplot with notch to find out how the medians of different data groups match with each other.

The below script will create a boxplot graph with notch for each of the data group.

Live Demo

# Give the chart file a name.

png(file = "boxplot\_with\_notch.png")

# Plot the chart.

boxplot(mpg ~ cyl, data = mtcars,

xlab = "Number of Cylinders",

ylab = "Miles Per Gallon",

main = "Mileage Data",

notch = TRUE,

varwidth = TRUE,

col = c("green","yellow","purple"),

names = c("High","Medium","Low")

)

[12:59 PM, 1/20/2019] Saket Vikas Sir Madrid: Histogram graph.......................................................................................................................................................

[12:59 PM, 1/20/2019] Saket Vikas Sir Madrid: # Create data for the graph.

v <- c(9,13,21,8,36,22,12,41,31,33,19)

# Give the chart file a name.

png(file = "histogram.png")

# Create the histogram.

hist(v,xlab = "Weight",col = "yellow",border = "blue")

[12:59 PM, 1/20/2019] Saket Vikas Sir Madrid: # Create data for the graph.

v <- c(9,13,21,8,36,22,12,41,31,33,19)

# Give the chart file a name.

png(file = "histogram\_lim\_breaks.png")

# Create the histogram.

hist(v,xlab = "Weight",col = "green",border = "red", xlim = c(0,40), ylim = c(0,5),

breaks = 5)

[1:00 PM, 1/20/2019] Saket Vikas Sir Madrid: Line Graph......................................................................................................................................................................

[1:00 PM, 1/20/2019] Saket Vikas Sir Madrid: A simple line chart is created using the input vector and the type parameter as "O". The below script will create and save a line chart in the current R working directory.

Live Demo

# Create the data for the chart.

v <- c(7,12,28,3,41)

# Give the chart file a name.

png(file = "line\_chart.jpg")

# Plot the bar chart.

plot(v,type = "o")

[1:00 PM, 1/20/2019] Saket Vikas Sir Madrid: # Create the data for the chart.

v <- c(7,12,28,3,41)

# Give the chart file a name.

png(file = "line\_chart\_label\_colored.jpg")

# Plot the bar chart.

plot(v,type = "o", col = "red", xlab = "Month", ylab = "Rain fall",

main = "Rain fall chart")

[1:01 PM, 1/20/2019] Saket Vikas Sir Madrid: Multiple Lines in a Line Chart

More than one line can be drawn on the same chart by using the lines()function.

After the first line is plotted, the lines() function can use an additional vector as input to draw the second line in the chart,

Live Demo

# Create the data for the chart.

v <- c(7,12,28,3,41)

t <- c(14,7,6,19,3)

# Give the chart file a name.

png(file = "line\_chart\_2\_lines.jpg")

# Plot the bar chart.

plot(v,type = "o",col = "red", xlab = "Month", ylab = "Rain fall",

main = "Rain fall chart")

lines(t, type = "o", col = "blue")

[1:01 PM, 1/20/2019] Saket Vikas Sir Madrid: Scatter Plot......................................................................................................................................

[1:01 PM, 1/20/2019] Saket Vikas Sir Madrid: Creating the Scatterplot

The below script will create a scatterplot graph for the relation between wt(weight) and mpg(miles per gallon).

Live Demo

# Get the input values.

input <- mtcars[,c('wt','mpg')]

# Give the chart file a name.

png(file = "scatterplot.png")

# Plot the chart for cars with weight between 2.5 to 5 and mileage between 15 and 30.

plot(x = input$wt,y = input$mpg,

xlab = "Weight",

ylab = "Milage",

xlim = c(2.5,5),

ylim = c(15,30),

main = "Weight vs Milage"

)

[1:02 PM, 1/20/2019] Saket Vikas Sir Madrid: pairs(~wt+mpg+disp+cyl,data = mtcars,

main = "Scatterplot Matrix")

[1:42 PM, 1/20/2019] Saket Vikas Sir Madrid: install.packages("XML")

[1:44 PM, 1/20/2019] Saket Vikas Sir Madrid: install.packages("rjson")

[1:45 PM, 1/20/2019] Saket Vikas Sir Madrid: install.packages("stringr")

install.packages("plyr")

[1:48 PM, 1/20/2019] Saket Vikas Sir Madrid: data <- read.xlsx("input.xlsx", sheetIndex = 1)