

Learning Outcomes

At the end of the session, you will be able to:

- Using sample datasets from library
- Visualizing basic plots- Pie Charts, Box plots, Histogram, Scatterplots, Bar chart, Line Graphs

Activity

1. Using sample datasets from library

- 1.1. To see the list of pre-loaded data, type the function data()
 - Write and run the following in R. Make your conclusion about the code:

```
data()
```

- 1.2. Loading the dataset. Write and run the following in R:

```
# Load whole dataset
data(mtcars)
# Print the first 10 rows
print(head(mtcars, 10))

# Store only 2 columns in a variable
input <- mtcars[,c('mpg','cyl')]
# Print the first 5 rows
print(head(input))
```

2. Visualizing basic plots

2.1. Demonstration of the image-like graphics built-ins of R:

- Write and run the following in R. Make your conclusion about the code:

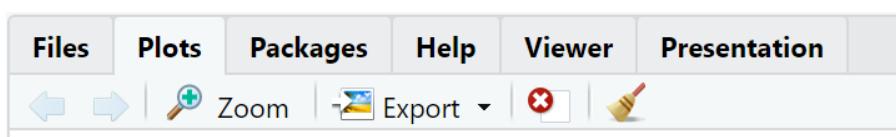
```
demo(graphics)
demo(image)
# press 'Enter' key in console to see next graphic/image
```

2.2. Creating Pie Charts

- Write and run the following in R. Make your conclusion about the code:

```
# Create data for the graph.
x <- c(21, 62, 10, 53)
labels <- c("London", "New York", "Singapore", "Mumbai")

# Plot the chart. You can view on the 'Plots' tab
pie(x,labels)
```



OR

```
Give the chart file a name.  
png(file = "city.png")  
# Plot the chart.  
pie(x,labels)  
# Save the file.  
# The image can be found in the same working directory  
dev.off()
```

2.3. Creating Pie Chart Title and Colors

- Write and run the following in R. Make your conclusion about the code:

```
# Plot the chart with title and rainbow color pallet.  
pie(x, labels, main = "City pie chart", col = rainbow(length(x)))
```

2.4. Creating Slice Percentages and Chart Legend

- Write and run the following in R. Make your conclusion about the code:

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

2.5. Creating 3D Pie Chart

- Write and run the following in R. Make your conclusion about the code:

```
# Get the library.  
library(plotrix)  
# Plot the chart.  
pie3D(x,labels = lbl,explode = 0.1, main = "Pie Chart of Countries  
)
```

2.6. Creating Box Plots

- Write and run the following in R. Make your conclusion about the code:

```
# Plot the chart.  
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders",  
ylab = "Miles Per Gallon", main = "Mileage Data")
```

2.7. Creating Box Plot with Notch

- Write and run the following in R. Make your conclusion about the code:

```
# 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")
)
```

2.8. Creating Histogram

- Write and run the following in R. Make your conclusion about the code:

```
# Create data for the graph.
v <- c(9,13,21,8,36,22,12,41,31,33,19)
# Create the histogram.
hist(v,xlab = "Weight",col = "yellow",border = "blue")
```

2.9. Creating Histogram Range of X and Y values

- Write and run the following in R. Make your conclusion about the code:

```
# Create data for the graph.
v <- c(9,13,21,8,36,22,12,41,31,33,19)
# Create the histogram.
hist(v,xlab = "Weight",col = "green",border = "red", xlim = c(0,40),
ylim = c(0,5),n breaks = 5)
```

2.10. Creating Scatterplots

- Write and run the following in R. Make your conclusion about the code:

```
# Get the input values.
input <- mtcars[,c('wt','mpg')]
# Plot the chart
plot(x = input$wt,y = input$mpg,
      xlab = "Weight",
      ylab = "Milage",
      xlim = c(2.5,5),
      ylim = c(15,30),
      main = "Weight vs Milage")
```

2.11. Creating Scatterplots Matrices

- Write and run the following in R. Make your conclusion about the code:

```
# Plot the matrices between 4 variables giving 12 plots.
# One variable with 3 others and total 4 variables.
pairs(~wt+mpg+disp+cyl,data = mtcars,
      main = "Scatterplot Matrix")
```

2.12. Creating Bar charts

- Write and run the following in R. Make your conclusion about the code:

```
# Create the data for the chart
H <- c(7,12,28,3,41)
# Plot the bar chart
barplot(H)
```

TEB2164 Introduction to Data Science
Lab 8a- Built-ins datasets & Data Visualization in R

2.13. Creating Bar Chart Labels, Title and Colors

- Write and run the following in R. Make your conclusion about the code:

```
# Create the data for the chart
H <- c(7,12,28,3,41)
M <- c("Mar","Apr","May","Jun","Jul")
# Plot the bar chart
barplot(H,names.arg=M,xlab="Month",ylab="Revenue",col="blue",
main="Revenue chart",border="red")
```

2.14. Creating Group Bar Chart and Stacked Bar Chart

- Write and run the following in R. Make your conclusion about the code:

```
# 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)
# 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)
```

2.15. Creating Line Graphs

- Write and run the following in R. Make your conclusion about the code:

```
# Create the data for the chart.
v <- c(7,12,28,3,41)
# Plot the chart.
plot(v,type = "o")
```

2.16. Creating Line Chart Title, Color and Labels

- Write and run the following in R. Make your conclusion about the code:

```
# Plot the bar chart.
plot(v,type = "o", col = "red", xlab = "Month", ylab = "Rain fall",
main = "Rain fall chart")
```

2.17. Creating Multiple Lines in a Line Chart

- Write and run the following in R. Make your conclusion about the code:

```
# Add more data for the chart.
t <- c(14,7,6,19,3)
# 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")
```