Data Analytics using R

SUVEN CONSULTANTS | NIRAJ SHARMA

Topics Covered

- Visualization (R-Base Class)
  - **Univariate** :
    - Plot
    - Plot.ts
    - Histogram
    - Pie Chart
  - **Bivariate** :
    - Barplot
    - Boxplot
    - Scatterplot
- Data Analytics : Case Study
  - UFO Sightings Analysis
  - IPL Deliveries Analysis
  - IPL Matches Analysis
  - Game of Thrones (Battle Analysis)

Visualization : Graphical analysis

- Graphs (Visualization) are useful for non-numerical data, such as colors, flavors, brand names, and more. When numerical measures are difficult or impossible to compute, graphs play an important role. Statistical computing is done with the aim to produce high-quality graphics.
- Various types of plots drawn in R are:
  - **Plots with single variables –** You can plot a graph for a single variable.
  - **Plots with multiple variables –** You can plot graph with multiple variables
  - **Special plots –** R has low and high-level graphics facilities.

Visualization : Graphical analysis

- *Plots for a Single Variable*
  - You may need to plot for a single variable in Graphical Data Analysis With R. For example, a plot showing daily sales values of a particular product over a period of time. You can also plot the time series for month by month sales.
  - R offers the following plotting functions for single variables:
    - **hist(y) –** Histograms to show a frequency distribution
    - **plot(y) –** Index plots to show the values of y in sequence
    - **plot.ts (y) –** Time series plots
    - **pie (x) –** Compositional plots like pie diagrams

Visualization : Graphical analysis

- Plots with Two Variables
  - The two types of variables used in the graphical data analysis with R:
    - Response variable
    - Explanatory variable
  - The **response variable** is represented on the y-axis and the **explanatory variable** is represented on the x-axis. Nature of the explanatory variable determines the kind of plot produced. When the explanatory variable is a continuous variable, such as length or weight or altitude, the appropriate plot to use is a scatterplot.

Visualization : Graphical analysis

- The types of plots available in R are:
  - **Scatterplots –** When the explanatory variable is a continuous variable.
  - **Stepped Lines –** Used to plot data distinctly and provide a clear view.
  - **Boxplots –** Boxplots show the location, spread of data and indicate skewness.
  - **Barplots –** It shows the heights of the mean values from the different treatments.

Visualization : Pie Chart

- Pie Chart
  - You can use pie charts to illustrate the proportional makeup of a sample in presentations. Here the function *pie* takes a vector of numbers and turns them into proportions. It then divides the circle on the basis of those proportions.
  - A pie-chart is a representation of values as slices of a circle with different colors. The slices are labeled and the numbers corresponding to each slice is also represented in the chart.

Visualization : Pie Chart

- Syntax

  pie(x, labels, radius, main, col, clockwise)

  Following is the description of the parameters used −

  x - is a vector containing the numeric values used in the pie chart.

  labels - is used to give description to the slices.

  radius - indicates the radius of the circle of the pie chart.(value between −1 and +1).

  main - indicates the title of the chart.

  col - indicates the color palette.

  clockwise - is a logical value indicating if the slices are drawn clockwise or anti clockwise.

Visualization : Pie Chart


```
# 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")

 (piepercent = (x / sum(x)) * 100)
 print(piepercent)
```

```
# Plot the chart.
pie(x,labels = labels, col =
      c("Red","Green","Blue","Yellow"))

legend("bottomleft",x,pch=10,legend = labels , col =
c("Red","Green","Blue","Yellow"))

 # Save the file.
 dev.off()
```

Visualization : Histogram

- Histogram
  - A histogram represents the frequencies of values of a variable bucketed into ranges. Histogram is similar to bar chat but the difference is it groups the values into continuous ranges. Each bar in histogram represents the height of the number of values present in that range.
  - Histograms display the mode, the spread, and the symmetry of a set of data.
  - The **hist()** is used to plot histograms.

Visualization : Histogram

- Syntax

  hist(v,main,xlab,xlim,ylim,breaks,col,border)

  Following is the description of the parameters used −

  v - is a vector containing numeric values used in histogram.

  main - indicates title of the chart.

  col - is used to set color of the bars.

  border - is used to set border color of each bar.

  xlab - is used to give description of x-axis.

  xlim - is used to specify the range of values on the x-axis.

  ylim - is used to specify the range of values on the y-axis.

  breaks - is used to mention the width of each bar.

Visualization : Histogram

- Code

```
# Create data for the graph.
 v <- c(47,53,31,58,36,43,22)

 # Give the chart file a name.
 png(file = "histogram.png")

 # Create the histogram.
 hist(v,ylim=c(0,5),xlab = "Weight",col = "yellow",border = "blue")

 # Save the file.
 dev.off()
```

Visualization : Plot.ts (Time Series)

- Plot.ts
  - The time series plot can be used to join the dots in an ordered set of y values when a period of time is complete. The issues arise when there are missing values in the time series (e.g., if sales values for two months are missing during the last five years), particularly groups of missing values (e.g., if sales values for two quarters are missing during the last five years) for which periods we typically know nothing about the behavior of the time series.

Visualization : Plot.ts (Time Series)

- Syntax

plot.ts(v,main,xlab,xlim,ylim,breaks,col,border)

Following is the description of the parameters used −

v - is a vector containing numeric values used in plot.

main - indicates title of the chart.

col - is used to set color of the bars.

xlab - is used to give description of x-axis.

xlim - is used to specify the range of values on the x-axis.

ylim - is used to specify the range of values on the y-axis.

Visualization : Plot.ts (Time Series)


```
# Get the data points in form of a R vector.
 rainfall <- c(799,1174.8,865.1,1334.6,635.4,918.5,685.5,998.6,784.2,985,882.8,1071)

 # Give the chart file a name.
 png(file = "timeseries.png")

 # Create the histogram.
 plot.ts(rainfall,xlab = "2012 Rainfall data",ylab = "Meters Rainfall",col = "blue",
       main = "Rainfall Plot")

 # Save the file.
 dev.off()
```

Visualization : Plot / Line Graph

- Plot (Univariate) / Line Graph (Index Plot)
  - A line chart is a graph that connects a series of points by drawing line segments between them. These points are ordered in one of their coordinate (usually the x-coordinate) value. Line charts are usually used in identifying the trends in data.
  - For plotting single samples, index plots can be used. The plot function takes a single argument. This is a continuous variable and plots values on the y-axis, with the x coordinate determined by the position of the number in the vector. Index plots are especially useful for error checking.

Visualization : Line Graph (Index Plot)

- Syntax

plot(v,type,col,xlab,ylab)

Following is the description of the parameters used −

v - is a vector containing the numeric values.

type - takes the value "p" to draw only the points, "l" to draw only the lines and "o" to draw both points and lines.

xlab - is the label for x axis.

ylab - is the label for y axis.

main - is the Title of the chart.

col - is used to give colors to both the points and lines.

Visualization : Line Graph (Index Plot)


```
# Create the data for the chart.
v <- c(7,12,28,3,41)
w <- c(9,14,24,8,50)
q <- c("V","W")

# Give the chart file a name.
png(file = "line_chart.jpg")

# Plot the Line chart.
plot(v,type = "o",col = "blue")
lines(w,type = "o",col = "red")
legend("topleft",q, pch = 15:16,col = c("blue","red"))

# Save the file.
dev.off()
```

Visualization : Bar Plot

- Bar Plot
  - Barplot used to show the heights of the mean values from the different treatments. Function *tapply* computes the heights of the bars. Thus it works out the mean values for each level of the categorical explanatory variable.
  - A bar chart represents data in rectangular bars with length of the bar proportional to the value of the variable. R uses the function **barplot()** to create bar charts. R can draw both vertical and Horizontal bars in the bar chart.

Visualization : Bar Plot

- Syntax

  barplot(H,xlab,ylab,main, names.arg,col)

  Following is the description of the parameters used −

  H - is a vector or matrix containing numeric values used in bar chart.

  xlab - is the label for x axis.

  ylab - is the label for y axis.

  main - is the title of the bar chart.

  names.arg - is a vector of names appearing under each bar.

  col - is used to give colors to the bars in the graph.

Visualization : Bar Plot


```
# Create the data for the chart.
H <- c(7,12,28,3,55)
labels <- c("A","B","C","D","E")

# Give the chart file a name.
png(file = "barchart.png")

# Plot the bar chart.
barplot(H, names.arg = labels ,
        col = c("white", "green","red", "blue",
"cyan"), ylim = c(0,60))

# Save the file.
dev.off()
```

Visualization : Box Plot / Whisker Plot

- Box Plot :
  - Boxplots are a measure of how well distributed is the data in a data set. It divides the data set into three quartiles. This graph represents the minimum, maximum, median, first quartile and third quartile in the data set. It is also useful in comparing the distribution of data across data sets by drawing boxplots for each of them.

  - A **box-and-whisker** plot is a graphical means of representing sets of numeric data using quartiles. It is based on the minimum and maximum values, and upper and lower quartiles.
    Boxplots summarizes the information available. The vertical dash lines are called the 'whiskers'. Boxplots are also excellent for spotting errors in data. The extreme outliers represents these errors.

Visualization : Box Plot / Whisker Plot

- Syntax

boxplot(x, data, notch, varwidth, names, main)

Following is the description of the parameters used −

x - is a vector or a formula.

data - is the data frame.

notch - is a logical value. Set as TRUE to draw a notch.

varwidth - is a logical value. Set as true to draw width of the box proportionate to the sample size.

names - are the group labels which will be printed under each boxplot.

main - is used to give a title to the graph.

Visualization : Box Plot / Whisker Plot


```
# Extract data from R-Studio
 data("mtcars")
 input <- mtcars[,c('mpg','cyl')]
 print(head(input))

# Give the chart file a name.
 png(file = "boxplot.png")
# Find Correlation between features
cor(mtcars)
```

```
# Plot the chart.
 boxplot(mpg ~ cyl, data = input,
        ylim = c(5,35),
        xlab = "Number of Cylinders",
        ylab = "Miles Per Gallon",
        main = "Mileage Data")

 # Save the file.
dev.off()
```

Visualization : Scatterplot

- Scatterplot :
  - Scatterplots show many points plotted in the Cartesian plane. Each point represents the values of two variables. One variable is chosen in the horizontal axis and another in the vertical axis.
  - Scatterplots displays a certain relationship between two variables. In this type of variable, the X-axis measures one variable and Y-axis measures another variable. On the one hand, if the values of both variable increase at the same time, a positive relationship exists between variables. On the other hand, if the value of one variable decreases at the time of increasing value of another variable, a negative relationship exists between variables.

Visualization : Scatterplot

- Syntax

  plot(x, y, main, xlab, ylab, xlim, ylim, axes)

  Following is the description of the parameters used −

  x - is the data set whose values are the horizontal coordinates.

  y - is the data set whose values are the vertical coordinates.

  main - is the tile of the graph.

  xlab - is the label in the horizontal axis.

  ylab - is the label in the vertical axis.

  xlim - is the limits of the values of x used for plotting.

  ylim - is the limits of the values of y used for plotting.

  axes - indicates whether both axes should be drawn on the plot.

Visualization : Scatterplot


```
# Get the input values.
 data("mtcars")
 input <- mtcars[,c('wt','mpg')]

 # Give the chart file a name.
 png(file = "scatterplot.png")

# Find Correlation between features
cor(mtcars)
```

```
# 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"
 )

 # Save the file.
dev.off()
```

Data Analytics – Case Study

Data Analytics : Case Study 3

- You have been provided with UFO sightings data. You are required to Analyze & Visualize the following: (**DATASET – complete.csv**)
  - Find Top Shape of UFO seen & described by peoples
  - Find Top Country where sightings occurred
  - Find Top State in which sightings occurred
  - Find Top city in which sightings occurred
  - Find time usually in which sightings occurred
  - Find Top Year in which sightings occurred
  - Map sightings on World Map

Data Analytics : Case Study 4

- You are employed as a Data Scientist by the cricket association and you are working on a project to analyze the Cricket Players. You are required to Analyze & Visualize the following: (**DATASET – deliveries.csv**)
  - Find Bowler name who bowled Max wide runs
  - Find No of Super over in match id 10
  - Find Batsman name whose total_run > 6
  - Find Bowler name in Match id 25 with the Max extra runs
  - Find Batsman, Bowler, Non Striker where Maximum Total Runs
  - Find Bowler name where Max Penalty Runs
  - Find Top 10 Batsman
  - Find Top 10 Bowlers
  - Find least runs given by bowler (TypeWise)

Data Analytics : Case Study 5

- You are employed as a Data Scientist by the cricket association and you are working on a project to analyze the Cricket Players. You are required to Analyze & Visualize the following: (**DATASET – matches.csv**)
  - Find Total No of matches played in Every Season
  - Find no of matches who win_by_wickets & win_by_runs
  - No of matches that were Tie
  - Distinct name of teams that played in 2016
  - Find total no of matches won by mumbai indians in 2010
  - Number of matches played in different stadiums
  - Number of matches played by each team
  - Number of Matches won by each teams
  - Does toss win has any affect on winning the match ?

Data Analytics : Case Study 6

- You are employed as a Data Scientist by the Hotstar and you are working on a project to analyze Game of Thrones Tv Series. You are required to Analyze & Visualize the following: (**DATASET – battles.csv**)
  - Which King fought Maximum Number of Battles?
  - Number of battles by region?
  - Number of battles by region but different places ?
  - Types of Battles ?
  - Types of battle and attacker king ?
  - Which pitch causes major death ?

Instructor Contact Information

nirajshar67@gmail.com /
rocky@suvenconsultants.com

9167687087 / 9892544177

10 am – 7 pm

https://www.linkedin.com/in/niraj7654/