BVV Sangha, Bagalkot
AMRUTA INSTITUTE OF ENGINEERING & MANAGEMENT SCIENCES
Approved by AICTE, New Delhi
Recognized by Government of Karnataka & Affiliated to VTU, Belagavi
AIEMS BENGALURU

# DATA ANALYTICS WITH R ( BDS306C )

**AI/ML – 3rd Semester**

**AMRUTA INSTITUTE OF ENGINEERING AND MANAGEMENT SCIENCES**

**ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

**Academic Year : 2025-26**

**Faculty Name : Dr. Sanjana Prasad**

**MODULE 4**

**GRAPHICS USING R**

**TOPICS**

- **Exploratory Data Analysis**
- **Main graphical packages**
- **Pie charts using R**
- **Scatter plots using R**
- **Line plots using R**
- **Histograms using R**
- **Box plots using R**
- **Bar plots using R**
- **Other existing graphical packages**

## 1. EXPLORATORY DATA ANALYSIS:

- Exploratory Data Analysis (EDA) is a visual based method used to analyse data sets and to summarize their main characteristics.

- EDA shows how to use visualisation and transformation to explore data in a systematic way.
- EDA is an iterative cycle of the below steps:

***1) Generate questions about data.***

***2) Search for answers by visualising, transforming, and modelling data.***

***3) Use what is learnt to refine questions and/or generate new questions.***

Exploratory Data Analysis (EDA) is an approach for data analysis that employs a variety of techniques (mostly graphical) to:

***1) Maximize insight into a data set***

***2) Uncover underlying structure***

***3) Extract important variables***

***4) Detect outliers and anomalies***

***5) Test underlying assumptions***

***6) Develop parsimonious models***

***7) Determine optimal factor settings.***

**PRACTICE QUESTIONS**

**Q1 (L2 – Understand)**

1. Explain the concept of Exploratory Data Analysis (EDA). Why is EDA considered an essential step before applying machine learning algorithms?

**Q2 (L4 – Analyze)**

2. Given a real-world dataset of your choice, analyze how EDA techniques help in identifying patterns, outliers, and data quality issues. Support your answer with

## 2. MAIN GRAPHICAL PACKAGES

- **Base graphics system** – Used for drawing basic graphs ; some limitations

- **Grid graphics system** - allows to plot the points or lines in the place where desired; does not allow us to draw a scatter plot.
- **Lattice graphics system** - Results of a plot can be saved. Scatter plots can contain multiple panels in which we can draw multiple graphs and compare them to each other.
- **ggplot2 graphics system ("grammar of graphics")** - breaks down the graphs into many parts or chunks.

**Q3 (L2 – Understand)**

Describe the major graphical packages available in R. Explain the purpose of base graphics, lattice graphics, and ggplot2.

**Q4 (L5 – Evaluate)**

Evaluate the suitability of base graphics, lattice, and ggplot2 for visualizing large datasets. Justify your choice with reasons and examples.

**TYPES OF CHARTS**

- Pie Charts
- Scatter Plots
- Line Plots
- Histograms
- Box Plots
- Bar Plots
- Other Graphical Packages

## 3. PIE CHARTS

- Pie chart is created using the pie() function .

- Takes positive numbers as vector input.
- Additional parameters - control labels, colour, title etc.

**Syntax for creating Pie charts :**

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

*x – numeric vector*

*labels – description of the slices*

*radius – values between [-1 to +1]*

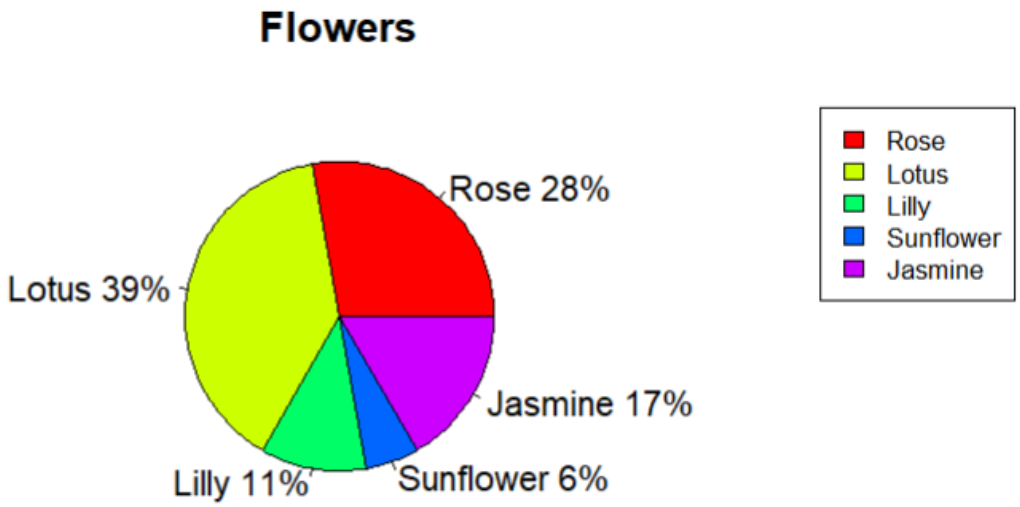
*main – title of the chart*

*col – colour palette*

*clockwise – logical value – TRUE (Clockwise), FALSE (Anti Clockwise)*

**Program :**

```
> x <- c(25, 35, 10, 5, 15)
>
> labels <- c("Rose", "Lotus", "Lilly", "Sunflower", "Jasmine")
>
> # Calculate percentage labels
> percent <- round(x / sum(x) * 100)
>
> # Create pie chart
> pie(x,
+     labels = paste0(labels, " ", percent, "%"),
+     main = "Flowers",
+     col = rainbow(length(x))
+ )
>
> # Add legend
> legend("topright",
+        legend = labels,
+        fill = rainbow(length(x)),
+        cex = 0.8
+ )
```




**Q5 (L3 – Apply)**

Write an R program to construct a pie chart representing category-wise data and enhance it by adding colors and percentage labels.

**Q6 (L6 – Create)**

Design an informative pie chart for a real-world scenario (such as market share or survey results) and interpret the insights obtained from the visualization.

## 4. SCATTER PLOTS

- Used for exploring the relationship between the **two continuous variables**.
- Consider the dataset "cars" that lists the "Speed and Stopping Distances of Cars".

- The basic scatter plot in the base graphics system can be obtained by using the plot() function as in Fig. 4.3.
- The below example compares if the speed of a car has effect on its stopping distance using the plot.
- **Basic Scatter Plot :**

```
> colnames(cars)
[1] "speed" "dist"
> plot(cars$speed, cars$dist)
```

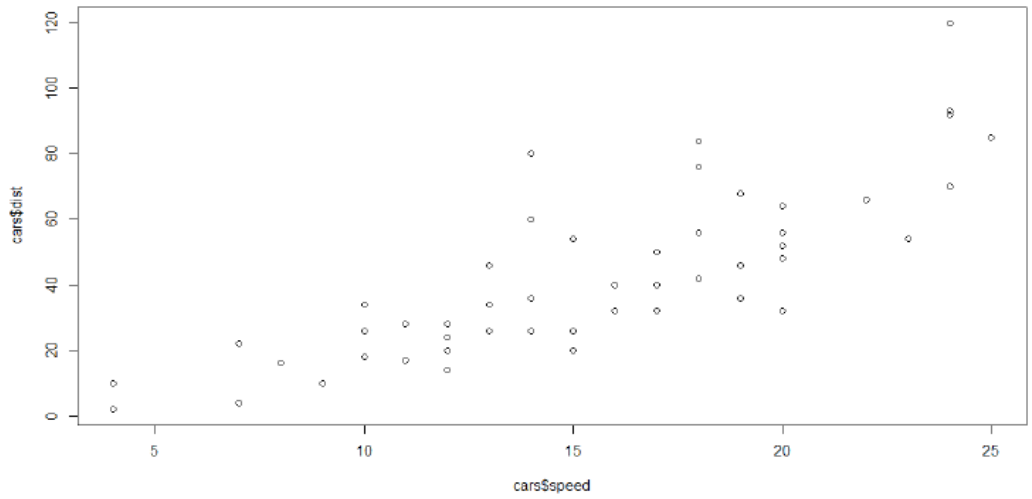



**Figure 4.3** Basic Scatter Plot of Car Speed Vs Distance

**To add colour and change the plotting character in Scatter Plots .**

- Arguments col and pch (can take the values between 1 and 25) in the plot() function are used.
- Thus the plot in Fig. 4.4 shows that there is a strong positive correlation between the speed of a car and its stopping distance.

- `pch = 0` for a square
- `pch = 1` for a circle
- `pch = 2` for a triangle pointing up
- `pch = 3` for a plus sign
- `pch = 4` for a cross
- `pch = 19` for a solid circle
- `pch = 21` for a filled circle (with separate border and fill colors)

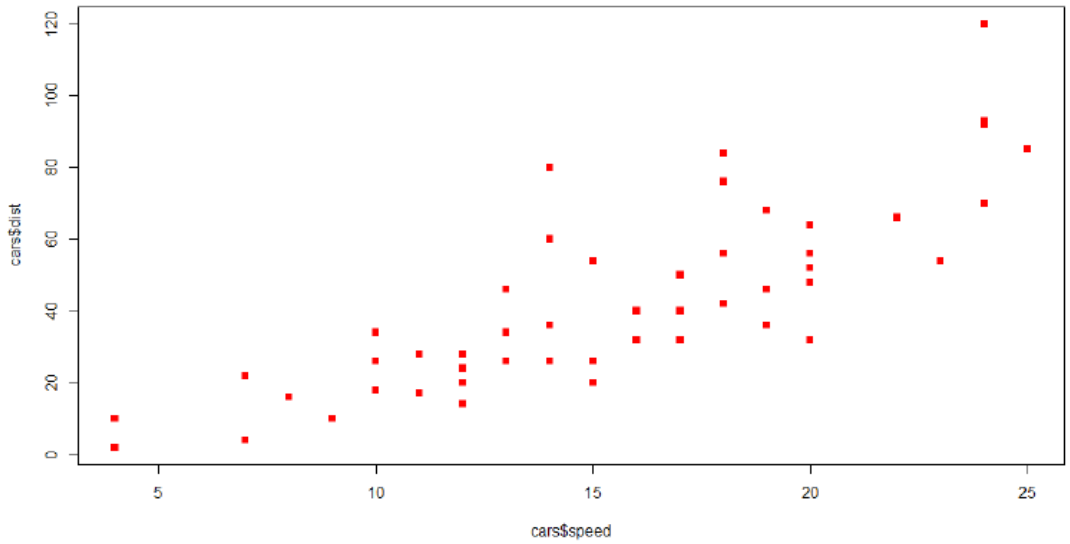



**Figure 4.4** Coloured Scatter Plot of Car Speed Vs Distance

- The **layout() function** is used to control the layout of multiple plots in the matrix.
- Thus in the example below multiple related plots are placed in a single figure as in Fig. 4.5.

**To draw multiple scatter plots in the same layout**

```
> data(mtcars)
> layout(matrix(c(1,2,3,4), 2, 2, byrow = TRUE))
> plot(mtcars$wt, mtcars$mpg, col = "blue", pch = 17)
> plot(mtcars$wt, mtcars$disp, col = "red", pch = 15)
> plot(mtcars$mpg, mtcars$disp, col = "dark green", pch = 10)
> plot(mtcars$mpg, mtcars$hp, col = "violet", pch = 7)
```

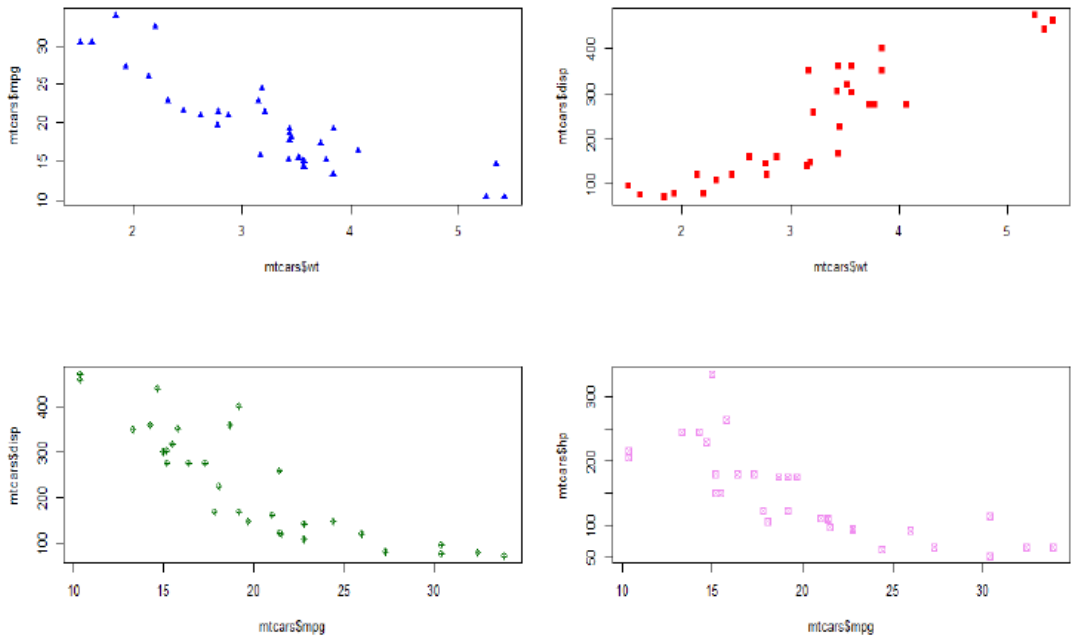



**Figure 4.5** Layout of Multiple Scatter Plots

**Scatter plot matrix:**

- Scatter plot matrix is used when there are **more than two variables** and we need to find the correlation between one variable versus the remaining ones
- A scatterplot matrix shows **multiple scatterplots in one grid**, comparing each variable with every other variable.

- **pairs() function** to create matrices of scatter plots.
- **Syntax for creating scatter plot matrices in R** :

**pairs(formula, data)**

**> pairs(~wt+mpg+disp+cyl,data = mtcars, main = "Scatterplot Matrix")**

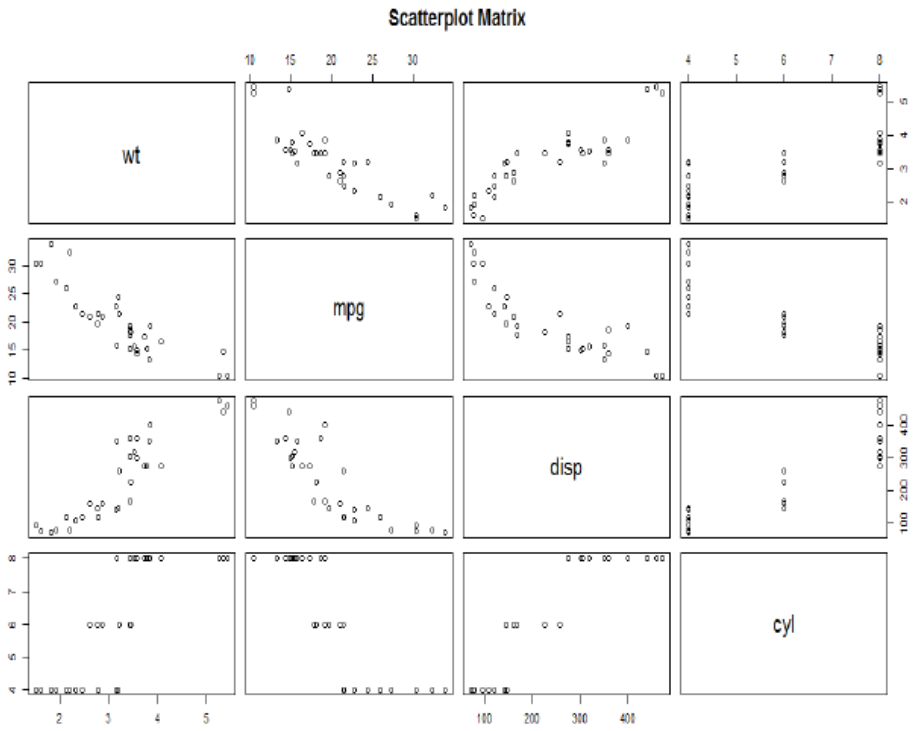



**Figure 4.6** Scatter Plot Matrix Using pairs()

- The **lattice graphics system** has equivalent of plot() function and it is **xyplot().**
- This function uses a formula to specify the x and y variables **(yvar ~ xvar) and a data frame argument.**
- To use this function, it is required to include the lattice package.

**> library(lattice)**

**> xyplot(mtcars$mpg ~ mtcars$disp, mtcars, col = "purple", pch = 7)**

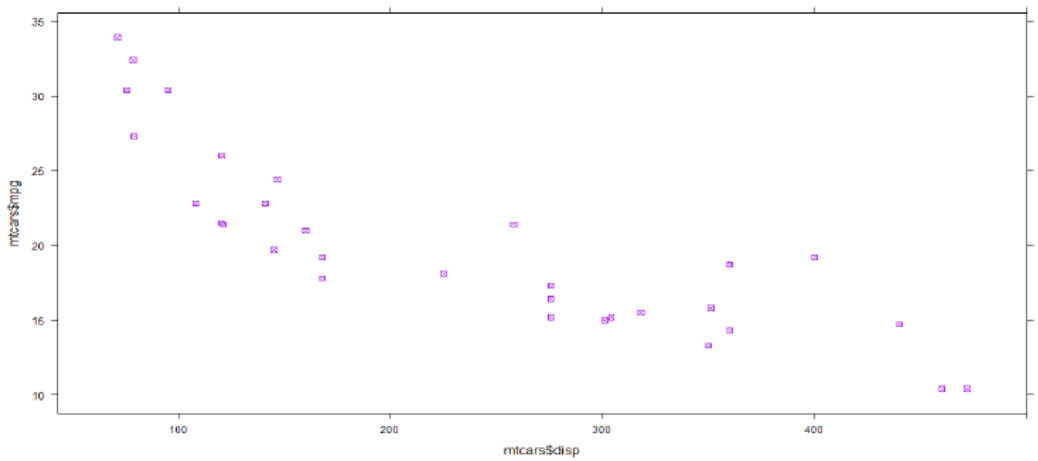



**Figure 4.7** Scatter Plot Matrix Using xyplot()

- **Axis scales** can be specified in the **xyplot()** using the scales argument and this argument must be a list.
- This list consists of the **name = value** pairs.
- If we mention **log = TRUE**, the log scales for the x and y axis are set as in Fig. 4.8.
- The **scales list** can take other arguments also like **the x and y that sets the x and y axes respectively.**

**> xyplot(mtcars$mpg ~ mtcars$disp, mtcars, scales = list(log = TRUE), col = "red", pch = 11)**

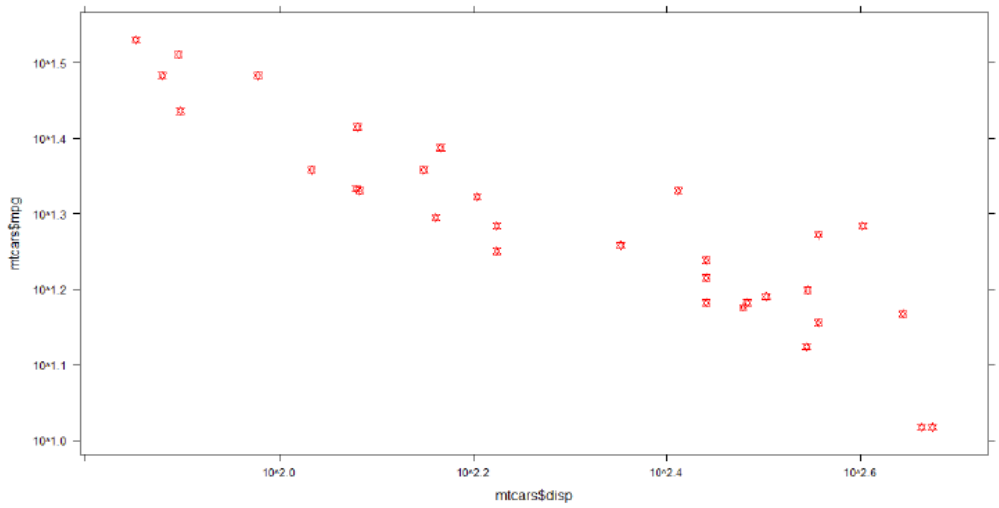



**Figure 4.8** Scatter Plot Matrix with Axis Scales Using xyplot()

- The data in the graph can be split based on one of the columns in the dataset namely mtcars$carb.
- This can be done by appending the pipe symbol (|) along with the column name used for splitting.
- The argument relation = "same" means that each panel shares the same axes. If the argument alternating = TRUE, axis ticks for each panel is drawn on alternating sides of the plot as in Fig. 4.9.

**> xyplot(mtcars$mpg ~ mtcars$disp | mtcars$carb, mtcars, scales = list(log = TRUE, relation = "same", alternating = FALSE), layout = c(3, 2), col = "blue", pch = 14)**

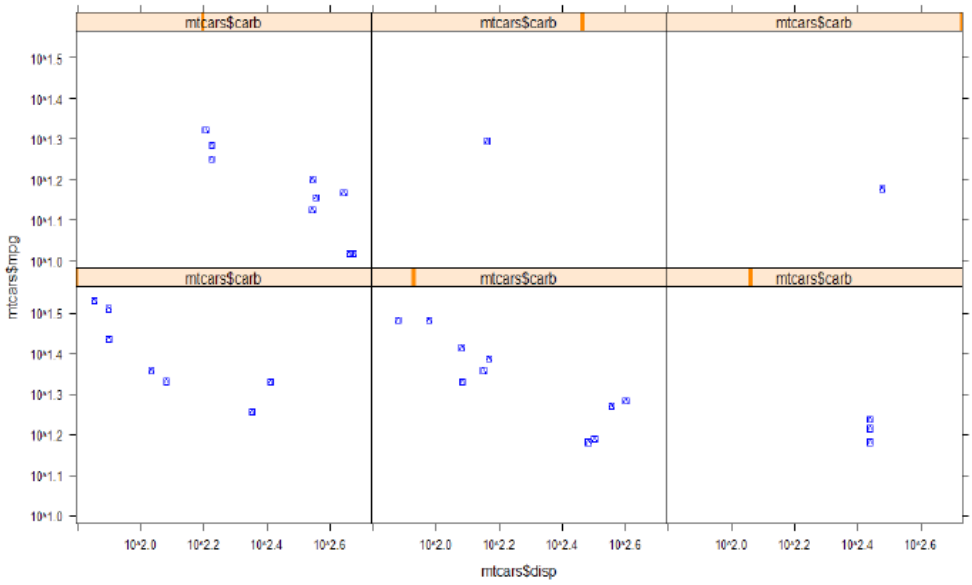



**Figure 4.9** Scatter Plot Split on a Column

## UPDATION OF LATTICE PLOTS :

- The lattice plots can be stored in variables and hence they can be further updated using the function update as below.

**> graph1 <- xyplot(mtcars$mpg ~ mtcars$disp | mtcars$carb, mtcars, scales = list(log = TRUE, relation = "same", alternating = FALSE), layout = c(3, 2), col = "blue", pch = 14)**

**> graph2 <- update(graph1, col = "yellow", pch = 6)**

- In the ggplot2 graphics, each plot is drawn with a call to the ggplot() function as in Fig. 4.10.
- This function takes a data frame as its first argument.
- The passing of data frame columns to the x and y axis is done using the **aes() function** which is used within the ggplot() function.

**Adding aesthetics to the graph**

The other aesthetics to the graph are then added using the **geom() function** appended with a **"+"** symbol to the **ggplot() function**.

**> library(ggplot2)**

**> ggplot(mtcars, aes(mpg, disp)) + geom_point(color = "purple", shape = 16, cex = 2.5)**

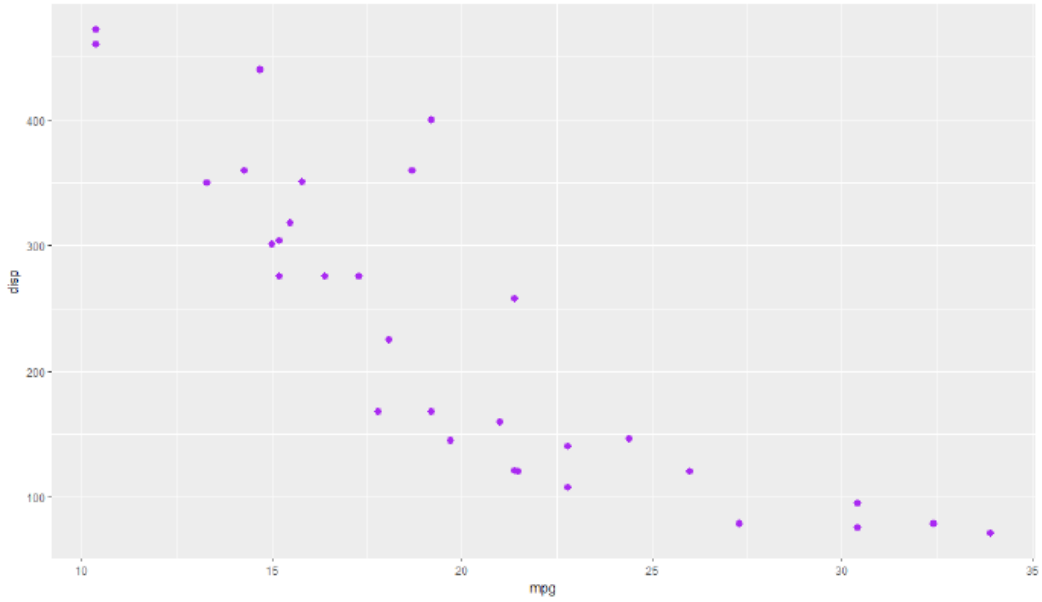



**Figure 4.10** Scatter Plot Using ggplot()

**SPLITTING OF GGPLOTS INTO SEVERAL PANELS:**

- The function **facet_wrap()** which takes a formula of the column used for splitting.
- The function **theme()** is used to specify the orientation of the axis readings.
- The functions **facet_wrap() and theme()** are appended to the **ggplot()** function using the **"+" symbol**.
- The ggplots can be stored in a variable like the lattice plots and as usual wrapping the expression in parentheses makes it to auto print.

**> (graph1 <- ggplot(mtcars, aes(mpg, disp)) + geom_point(color = "dark green", shape = 15, cex = 3))**

**> (graph2 <- graph1 + facet_wrap(~mtcars$cyl, ncol = 3) + theme(axis.text.x = element_text(angle = 90, hjust = 1)**

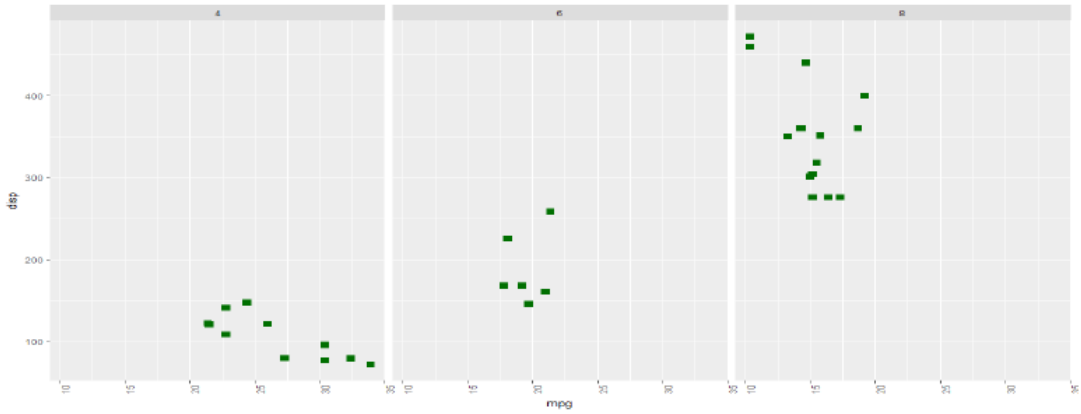



**Figure 4.11** – Scatter Plot Split into Panels Using ggplot()

4. Scatter Plots Using R

Q7 (L3 – Apply)

Develop an R program to create a scatter plot using an appropriate dataset and customize it using color and point symbols.

Q8 (L4 – Analyze)

Analyze how scatter plots help in identifying relationships and trends between two variables. Illustrate your answer using pairs() or ggplot().

## 4.5. Line Plots

- A line chart / line plot is a graph that connects a series of points by drawing line segments between them.
- Line charts are usually used in identifying the trends in data.
- The plot() function in R is used to create the line graph in base graphics as in Fig. 4.12.

- This function takes a vector of numbers as input together with few more parameters listed below.

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

***v – numeric vector***

***type - takes value "p" (only points), or "l" (only lines) or "o" (both points and lines)***

***xlab – label of x-axis***

***ylab – label of y-axis***

***main - title of the chart***

***col - colour palette***

**LINE PLOT USING BASE GRAPHICS**

```
> male <- c(1000, 2000, 1500, 4000, 800)
> female <- c(700, 300, 600, 1200, 800)
> child <- c(1000, 1200, 1500, 800, 2000)
> wages <- c("Male", "Female", "Children")
> color = c("red", "blue", "green")
> plot(male, type = "o", col = "red", xlab = "Month", ylab = "Wages",
main = "Monthly Wages", ylim = c(0, 5000))
> lines(female, type = "o", col = "blue")
> lines(child, type = "o", col = "green")
> legend("topleft", wages, cex = 0.8, fill = color)
```


**Figure 4.12** Line Plot Using Basic Graphics

## LINE PLOTS USING LATTICE GRAPHICS

- Line plots in the lattice graphics uses the **xyplot() function.**
- Multiple lines can be created using the **"+" symbol** in the formula where the **x and the y axes** are mentioned.
- The argument **type = "l"** is used to mention that it is a **continuous line.**

**> xyplot(economics$pop + economics$unemploy ~ economics$date, economics, type = "l")**

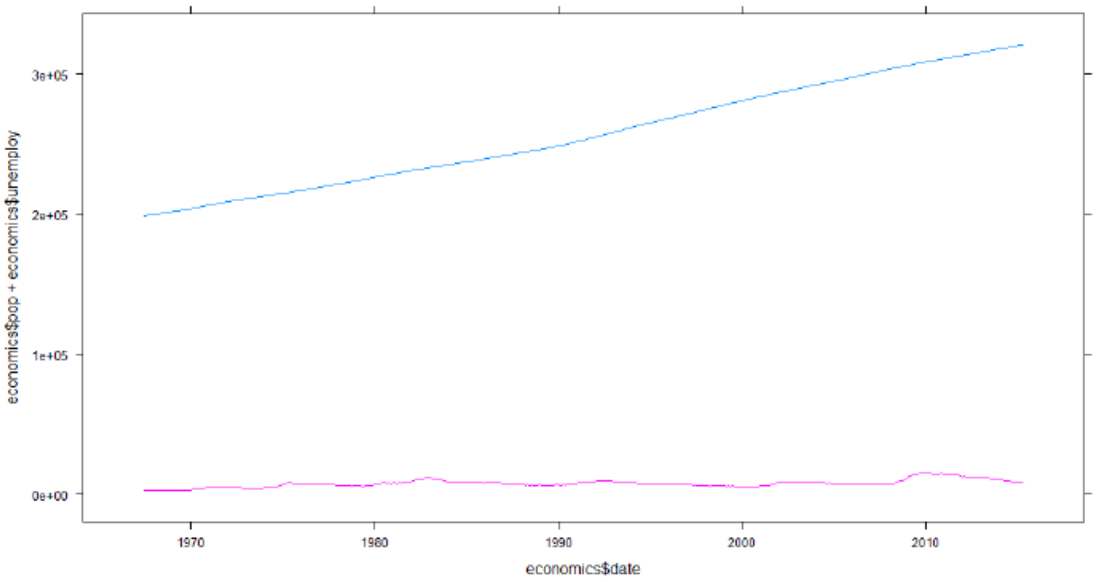



**Figure 4.13** Line Plot Using Lattice Graphics

## LINE PLOT USING ggplot2 GRAPHICS :

- Instead of geom_plot() function , geom_line() function is used .
- Multiple geom_line() functions are needed for drawing multiple lines to be drawn in the graph.

**> ggplot(economics, aes(economics$date)) + geom_line(aes(y =economics$pop)) + geom_line(aes(y = economics$unemploy))**

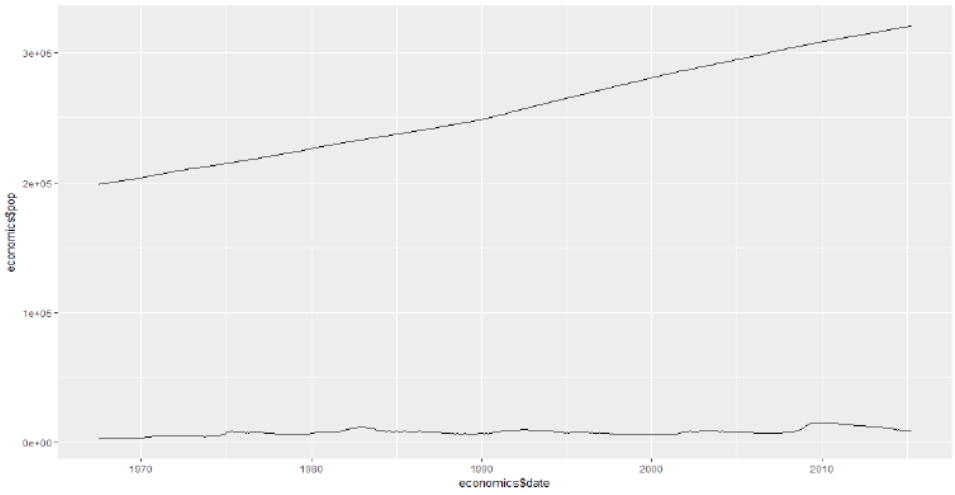



**Figure 4.14** Line Plot Using ggplot2 Graphics

## LINE PLOT USING geom_ribbon() FUNCTION

- The plot can be drawn without using multiple geom_line() functions also.
- This is possible using the function geom_ribbon() as mentioned below.
- This function plots not only the two lines, along with the contents in between the two lines.

**> ggplot(economics, aes(economics$date, ymin = economics$unemploy, ymax = economics$pop)) + geom_ribbon(color = "blue", fill = "white")**

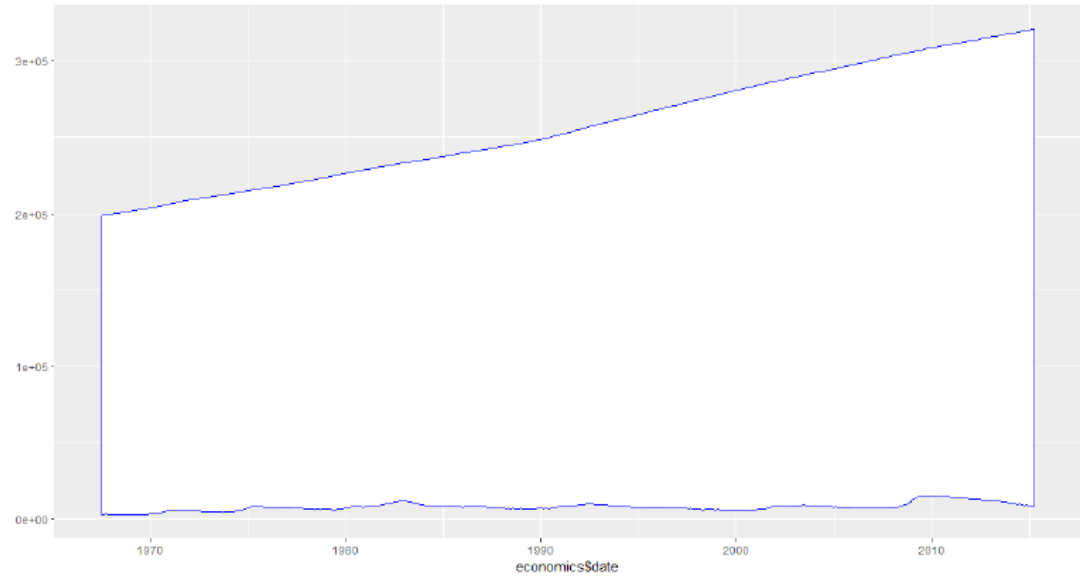



**Figure 4.15** Line Plot Using geom_ribbon()

**5. Line Plots Using R**

**Q9 (L2 – Understand)**

Define a line plot. Explain its importance in representing time-series data with suitable examples.

**Q10 (L5 – Evaluate)**

Compare line plots created using base graphics, lattice graphics, and ggplot2. Evaluate which method is more effective for trend analysis and justify your answer.

## 6.Histograms

- Histograms represents the **variable values frequencies**, that are split into **ranges.**
- It is similar to **bar charts**.
- Histograms group values into **continuous ranges**.
- In R histograms in the **base graphics** are drawn using the **function hist()** as in the Fig. 4.16, that takes a **vector of numbers as input** together with few more parameters listed below.

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

*v – numeric vector ; main - title of the chart*

*col - colour palette ; border – border colour*

*xlab – label of x-axis ;  xlim – range of x-axis*

*ylim – range of y-axis ; breaks – width of each bar*

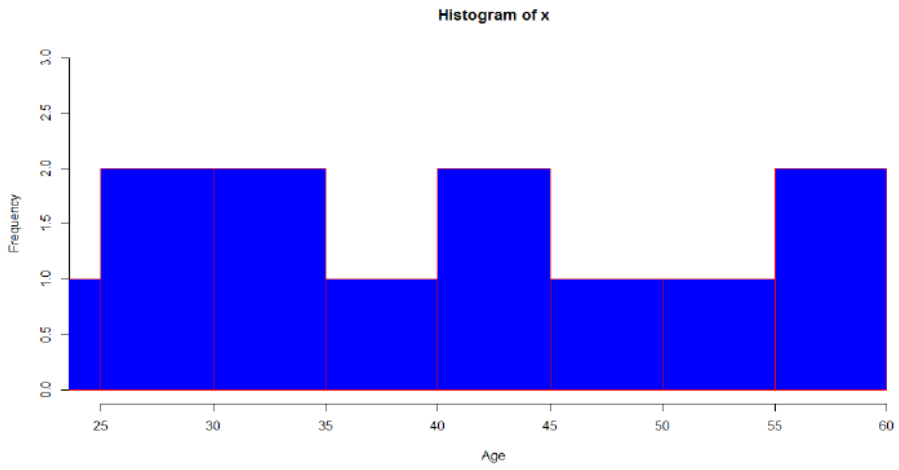



**Figure 4.16** Histogram Using Base Graphics

## HISTOGRAMS USING BASE GRAPHICS

**> x <- c(45, 33, 31, 23, 58, 47, 39, 58, 28, 55, 42, 27)**

**> hist(x, xlab = "Age", col = "blue", border = "red", xlim = c(25, 60), ylim = c(0, 3), breaks = 5)**

## HISTOGRAM USING LATTICE GRAPHICS :

- The lattice histogram is drawn using the function **histogram()**
- Similar to the behavior of **base ones**.
- Allows **easy splitting of data into panels** and **saving plots as variables**.
- The **breaks argument** behaves the same way as with **hist()**.
- The lattice histograms support **counts, probability densities, and percentage** y-axes via the type argument, which **takes the string "count", "density", or "percent"**.

**Code : > histogram(~ mtcars$mpg, mtcars, breaks = 10)**

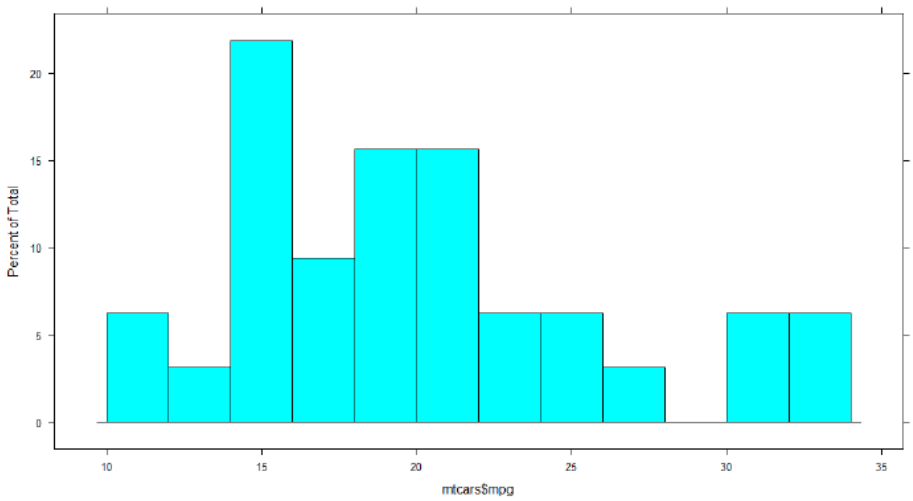



**Figure 4.17** Histogram Using Lattice Graphics

## HISTOGRAMS USING ggplot() FUNCTION

- The ggplot histograms are created by adding the function **geom_histogram()** to the ggplot() function as in Fig. 4.18.

- Bin specification is simple here, we just need to pass a **numeric bin width to geom_histogram()** function.
- It is possible to **choose between counts and densities** by passing the special names ..count.. or ..density.. to the y-aesthetic.

**> ggplot(mtcars, aes(mtcars$mpg, ..density..)) + geom_histogram(binwidth = 5)**

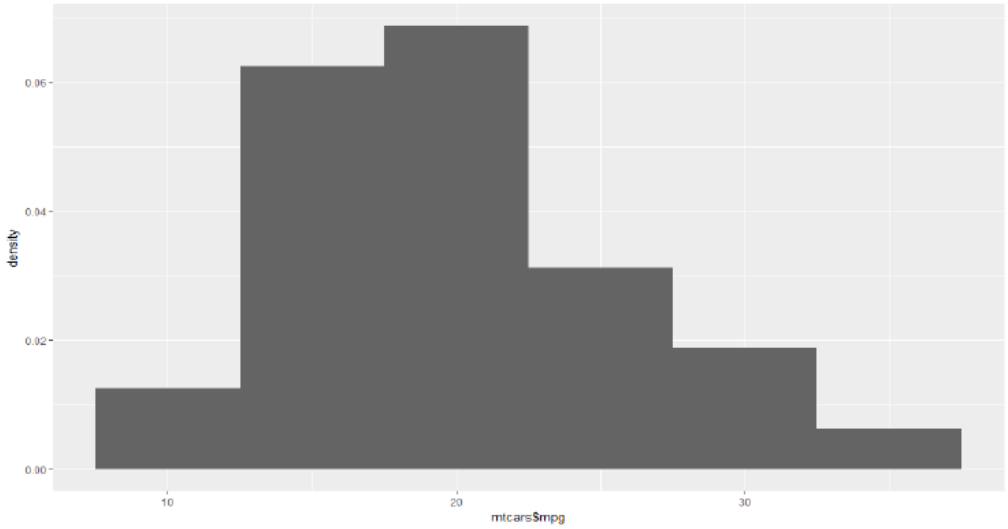



**Figure 4.18** Histogram Using ggplot2 Graphics

**6. Histograms Using R**

**Q11 (L3 – Apply)**

Write an R program to generate a histogram and modify its appearance using appropriate arguments such as color, labels, and axis limits.

**Q12 (L4 – Analyze)**

Analyze how histograms help in understanding the distribution of data. Compare the output obtained using base graphics and lattice graphics.

**7. Box Plots**

- The box plot divides the data into **three quartiles**.
- This graph represents **the minimum, maximum, median, first quartile and third quartile** in the data.
- This **shows the data distribution by drawing the box plots**.

**BOX PLOT USING LATTICE GRAPHICS - boxplot() function**

- In R base graphics the box plot is created using the **boxplot() function** as in Fig. 4.19, which takes the following parameters.
- The parameters are used to **give the data as a data frame, a vector or a formula, a logical value to draw a notch, a logical value to draw a box** as per the width of the sample, give title of the chart, labels for the boxes.
- Syntax for creating a box-plot

**Syntax :**

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

***x – vector or a formula***

***data – data frame***

***notch – logical value (TRUE – draw a notch)***

***varwidth – logical value (TRUE – box width proportionate to sample size***

***names – labels printed under the boxes***

***main – title of the chart***

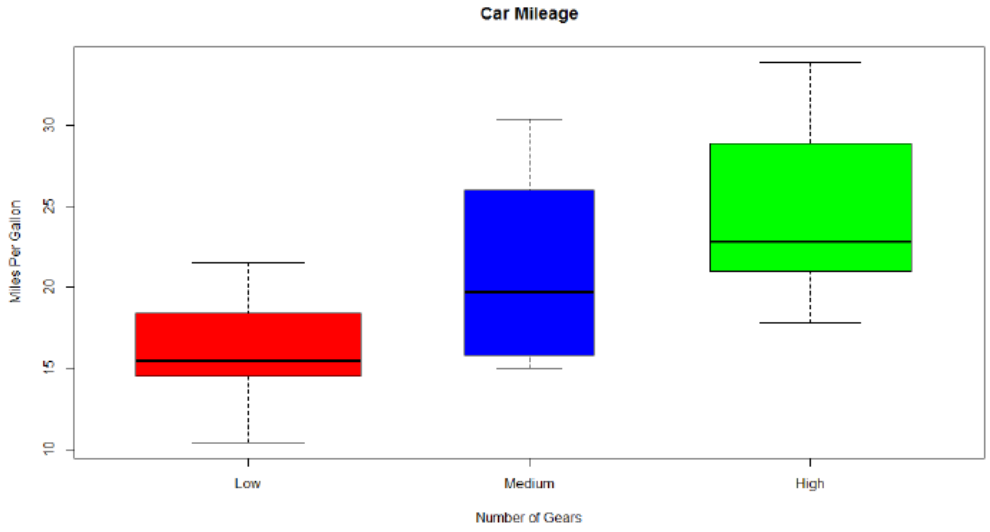



**Figure 4.19** Box Plots Using Base Graphics

## REORDERING THE BOX PLOTS FROM SMALLEST TO LARGEST

- **reorder() function** changes the order of a factor's levels, based upon some numeric score.

**> boxplot(mpg ~ reorder(gear, mpg, median), data = mtcars, xlab = "Number of Gears", ylab = "Miles Per Gallon", main = "Car Mileage", varwidth = TRUE, col = c("red","blue", "green"), names = c("Low", "Medium", "High"))**

## BOX PLOT USING LATTICE GRAPHICS - bwplot() function

In the lattice graphics the box plot is drawn using the function bwplot() as in Fig. 4.20.

**> bwplot(mpg ~ reorder(gear, mpg, median), data = mtcars,**
**xlab = "Number of Gears", ylab = "Miles Per Gallon",**
**main = "Car Mileage", varwidth = TRUE,**
**col = c("red","blue", "green"), names = c("Low", "Medium", "High"))**

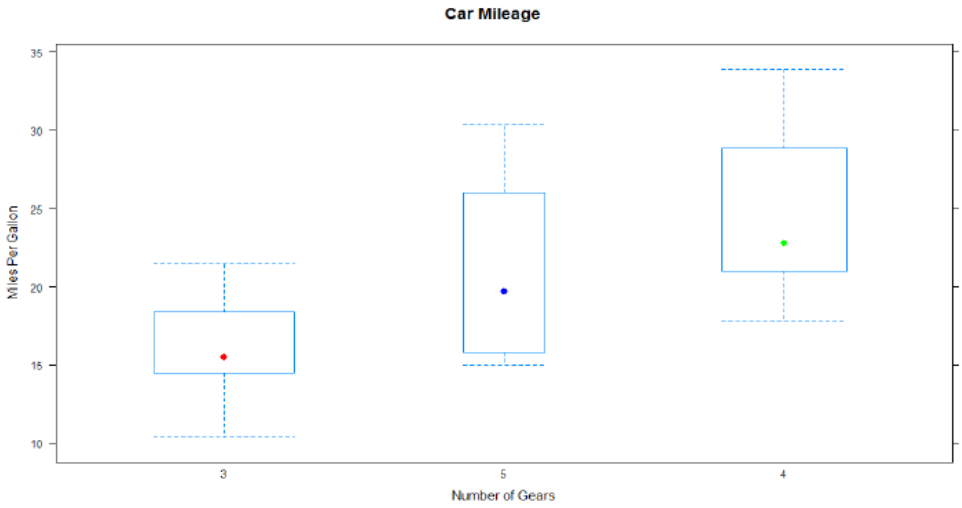



**Figure 4.20** Box Plots Using Lattice Graphics

## BOX PLOTS USING ggplot2 graphics

In the ggplot2 graphics the box plot is drawn by adding the function geom_ boxplot() to the function ggplot().

**> ggplot(mtcars, aes(reorder(gear, mpg, median), mpg)) + geom_boxplot()**

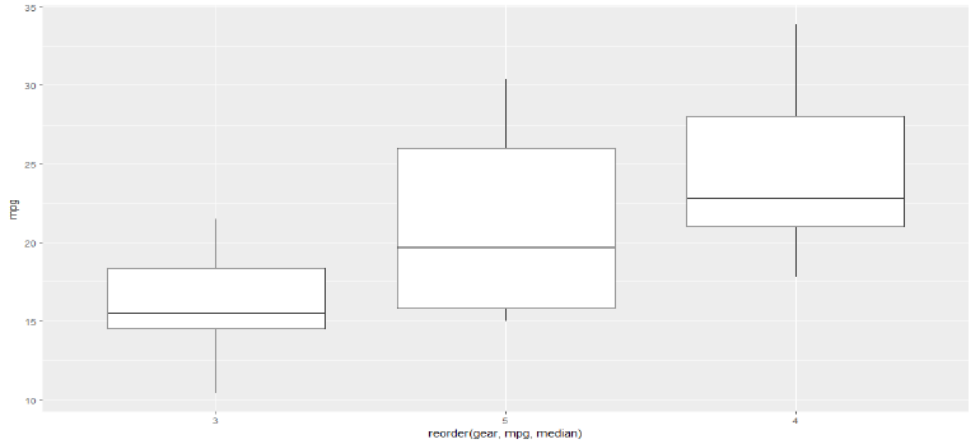



**Figure 4.21** Box Plots Using ggplot2 Graphics

**7. Box Plots Using R**

**Q13 (L2 – Understand)**

**Explain the significance of box plots in data analysis. Describe the information conveyed by median, quartiles, and outliers.**

**Q14 (L4 – Analyze)**

**Using a suitable dataset, analyze variations in data using box plots drawn with lattice graphics or ggplot2.**

### 4.8. Bar Plots

- Bar charts used to display **numeric variables split by a categorical variable**.
- In R base graphics, the bar chart is created using the **barplot() function** as in Fig. 4.22, which takes a matrix or a vector of numeric values.
- The additional parameters are used to give labels to the **X-axis, Y-axis, give title of the chart, labels for the bars and colours**.

**SYNTAX FOR CREATING A BAR-CHART**

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

***H – numeric vector or matrix***

***x-lab – label of x-axis***

***y-lab – label of y-axis***

***main - title of the chart***

***names.arg – vector of labels under each bar***

***col – colour palette***

**EXAMPLE PROGRAM :**

```
> x <- matrix(c(1000, 900, 1500, 4400, 800, 2100, 1700, 2900, 3800),
nrow = 3, ncol = 3)
> years <- c("2011", "2012", "2013")
```

**> city <- c(“Chennai”, “Mumbai”, “Kolkata”)**
**> color <- c(“red”, “blue”, “green”)**
**> barplot(x, main = “Yearly Sales”, names.arg = years, xlab = “Year”, ylab = “Sales”, col = color)**
**> legend(“topleft”, city, cex = 0.8, fill = color)**

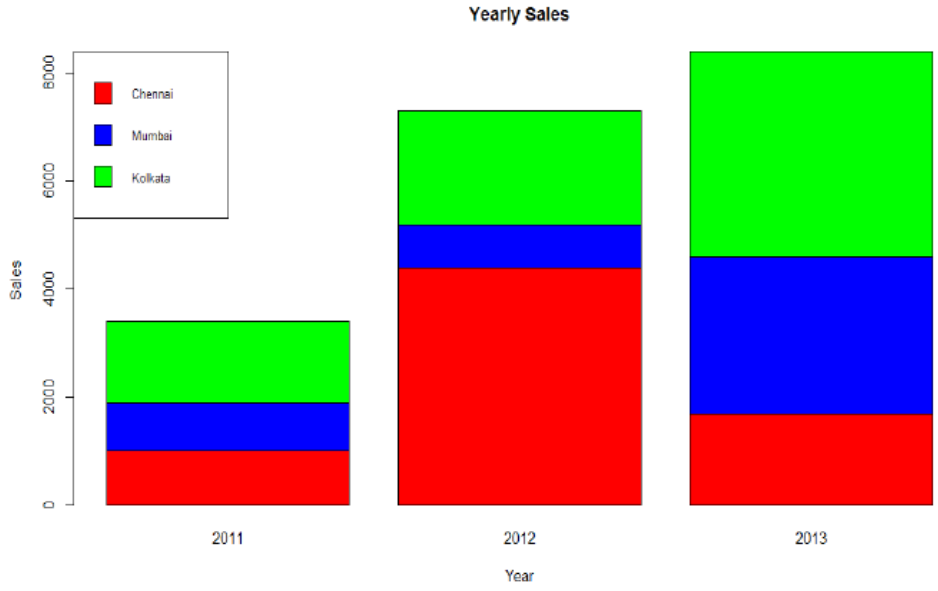



**Figure 4.22** Vertical Bar Plot Using Base Graphics

## HORIZONTAL BAR PLOTS USING BASE GRAPHICS :

- The parameter **horiz = TRUE** is used
- Fiddling with the plot parameters, via the **par() function**.
- To **Controls the labels** should be horizontal, vertical, parallel, or perpendicular to the axes – use the **las parameter**
- Plots are usually more readable if you set **las = 1**, for horizontal.
- The **mar parameter** is a numeric vector of length 4, giving the width of the **plot margins at the bottom/left/ top/right of the plot**.

## EXAMPLE PROGRAM :

**> x <- matrix(c(1000, 900, 1500, 4400, 800, 2100, 1700, 2900, 3800), nrow = 3, ncol = 3)**
**> years <- c(“2011”, “2012”, “2013”)**
**> city <- c(“Chennai”, “Mumbai”, “Kolkata”)**
**> color <- c(“red”, “blue”, “green”)**

```
> par(las = 1, mar = c(3, 9, 1, 1))
> barplot(x, main = "Yearly Sales", names.arg = years,
xlab = "Year", ylab = "Sales", col = color, horiz = TRUE)
> legend("bottomright", city, cex = 0.8, fill = color)
```

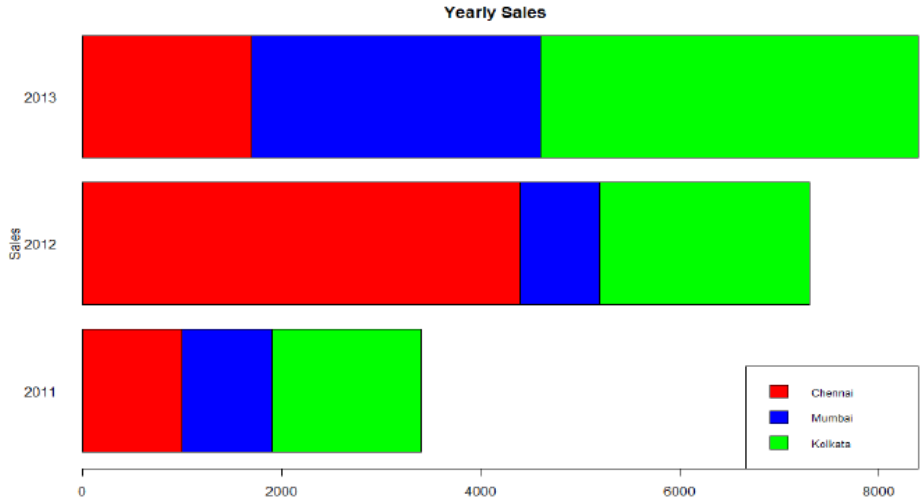



**Figure 4.23** Horizontal Bar Plot Using Base Graphics

## HORIZONTAL BAR PLOT USING LATTICE GRAPHICS

- Lattice equivalent of the function barplot(), is the function **barchart()**
- The formula interface is the same as those we saw with scatter plots,

**yvar ~ xvar**

```
> barchart(mtcars$mpg ~ mtcars$disp, mtcars)
```

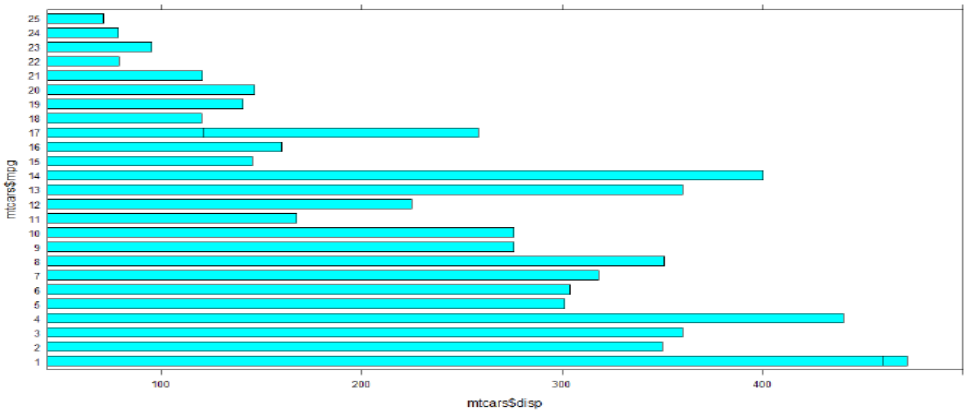



**Figure 4.24** Horizontal Bar Plot Using Lattice Graphics

## EXTENDING TO MULTIPLE VARIABLES

- Extending this to multiple variables just requires a tweak to the formula, and passing stack = TRUE to make a stacked plot as in Fig. 4.25.

**> barchart(mtcars$mpg ~ mtcars$disp + mtcars$qsec + mtcars$hp, mtcars, stack = TRUE**

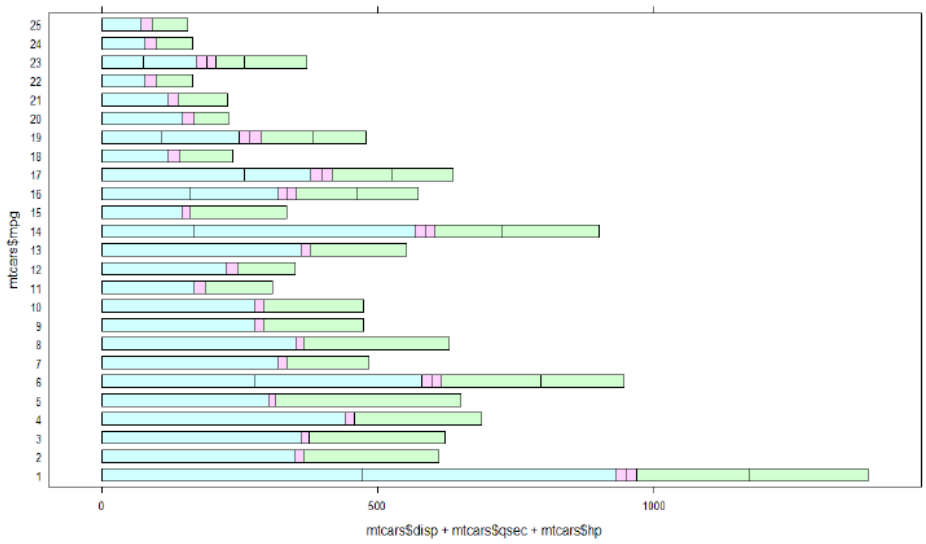



**Figure 4.25** Horizontal Stacked Bar Plot Using Lattice Graphics

## HORIZONTAL BAR PLOT USING ggplot2 GRAPHICS

- **geom_bar() function** is added to the **ggplot() function**.
- **Like base,** ggplot2 defaults to vertical bars; adding the function **coord_flip()** swaps this.
- The **argument stat= "identity"** to the function geom_bar().

**> ggplot(mtcars, aes(mtcars$mpg, mtcars$disp)) + geom_bar(stat = "identity") + coord_flip()**

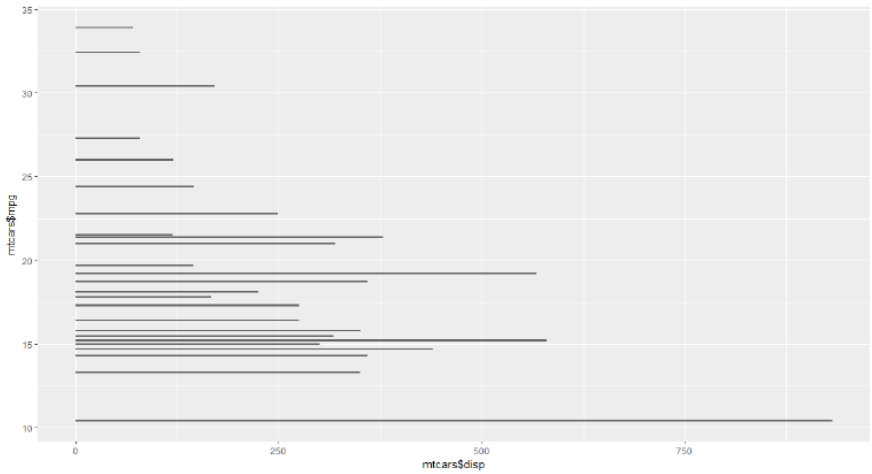



**Figure 4.26** Horizontal Bar Plot Using ggplot2 Graphics

## 8. Bar Plots Using R

**Q15 (L3 – Apply)**

**Write an R program to draw a bar plot for categorical data and customize it by changing orientation, colors, and labels.**

**Q16 (L6 – Create)**

**Create a grouped or stacked bar plot for a real-world dataset and interpret the insights obtained from the visualization.**

## 4.9. OTHER GRAPHICAL PACKAGES

| Package Name | Description |
|---|---|
| Vcd | Plots for visualizing categorical data, such as mosaic plots and association plots |
| Plotrix | Loads of extra plot types |
| latticeExtra | Extends the *lattice* package |
| GGally | Extend the *ggplot2* package |
| Grid | Provide access to the underlying framework of lattice and ggplot2 packages |

| Package Name | Description |
|---|---|
| gridSVG | Write grid-based plots to SVG files |
| Playwith | Allows pointing and clicking to interact with base or lattice plots |
| Iplots | Provides a whole extra system of plots with more interactivity |
| googleVis | Provides an R wrapper around Google Chart Tools, creating plots that can be displayed in a browser. |
| Rggobi | Provides an interface to GGobi package (for visualizing high-dimensional data) |
| GGobi | Open source visualization program for exploring high-dimensional data. |
| Rgl | Provides an interface to OpenGL for interactive 3D plots |
| Animation | Lets to make animated GIFs or SWF animations |
| rCharts | Provides wrappers to half a dozen JavaScript plotting libraries using lattice syntax. |

**9. Other Existing Graphical Packages in R**

**Q17 (L2 – Understand)**

Identify other graphical packages available in R apart from base, lattice, and ggplot2. Briefly explain their applications.

**Q18 (L5 – Evaluate)**

Evaluate the role of advanced graphical packages such as plotly, leaflet, or shiny in developing interactive data visualizations.

**HIGHLIGHTS**

- Exploratory Data Analysis (EDA) shows how to use visualisation and transformation to explore data in a systematic way.
- The main graphical packages are base, lattice and ggplot2.
- In R the pie chart is created using the pie() function.
- A 3D Pie Chart can be drawn using the package plotrix which uses the function pie3D().
- The basic scatter plot in the base graphics system can be obtained by using the plot() function.
- We use the arguments col and pch (values between 1 and 25) in the plot() function to specify colour and plot pattern.
- The layout() function is used to control the layout of multiple plots in the matrix.
- We use pairs() function to create matrices of scatter plots.
- The lattice graphics system has equivalent of plot() function and it is xyplot().
- In the ggplot2 graphics, each plot is drawn with a call to the ggplot() function.

- The ggplots can also be split into several panels like the lattice plots and this is done using the function facet_wrap().
- The plot() function in R is used to create the line graph in base graphics and the argument type = “l” is used to mention that it is a line.
- Line plots in the lattice graphics uses the xyplot() function and the argument type = “l” is used to mention that it is a line.
- The ggplot2 scatter plot are created by adding the function geom_line() to the ggplot() function.
- In R histograms in the base graphics are drawn using the function hist().
- The lattice histogram is drawn using the function histogram().
- The ggplot2 histograms are created by adding the function geom_ histogram() to the ggplot() function.
- In R base graphics the box plot is created using the boxplot() function.
- In the lattice graphics the box plot is drawn using the function bwplot().
- In the ggplot2 graphics the box plot is drawn by adding the function geom_boxplot() to the function ggplot().
- In R base graphics the bar chart is created using the barplot() function.
- The lattice equivalent of the function barplot(), is the function barchart().
- In the ggplot2 graphics the bar chart is drawn by adding the function geom_bar() to the function ggplot().