Exploratory Data Analysis

R Pie Charts

R programming language has several libraries for creating charts and graphs.

A pie-chart is a representation of values in the form of slices of a circle with different colors.

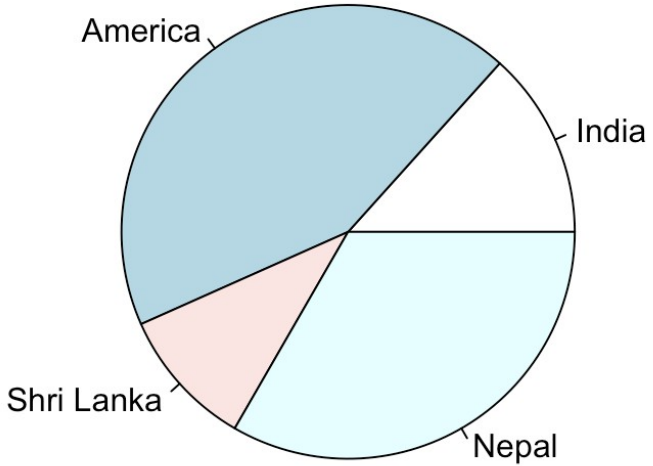
The Pie charts are created with the help of pie () function, which takes positive numbers as vector input. Additional parameters are used to control labels, colors, titles, etc.

1. pie(X, Labels, Radius, Main, Col, Clockwise)

pie(X, Labels, Radius, Main, Col, Clockwise)

- X is a vector that contains the numeric values used in the pie chart.
- Labels are used to give the description to the slices.
- Radius describes the radius of the pie chart.
- Main describes the title of the chart.
- Col defines the color palette.
- Clockwise is a logical value that indicates the clockwise or anti-clockwise direction in which slices are drawn.

```
# Creating data for the graph.

x <- c(20, 65, 15, 50)

labels <- c("India", "America", "Shri Lanka",
"Nepal")

# Giving the chart file a name.

png(file = "Country.jpg")

# Plotting the chart.

pie(x,labels)
```

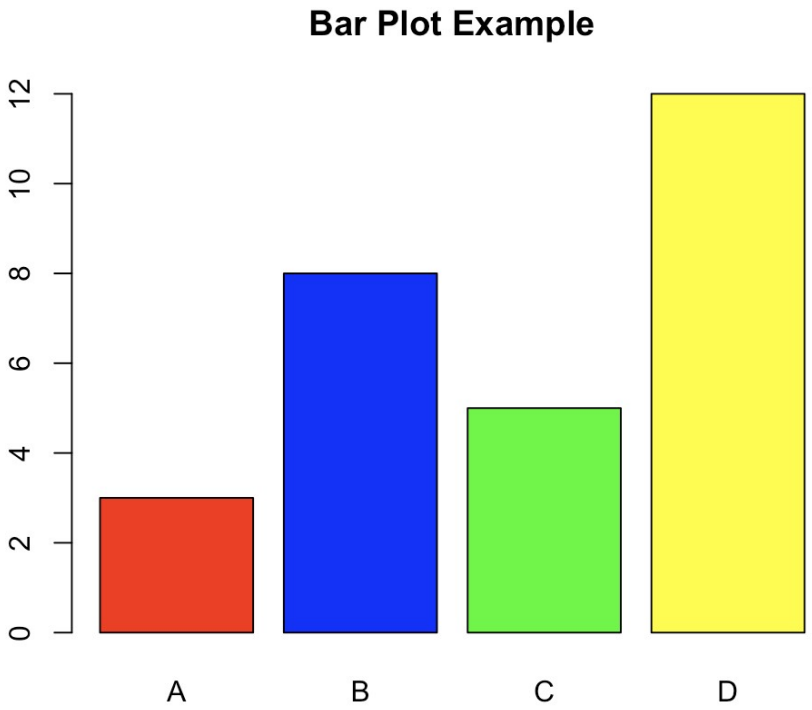


R Bar Charts

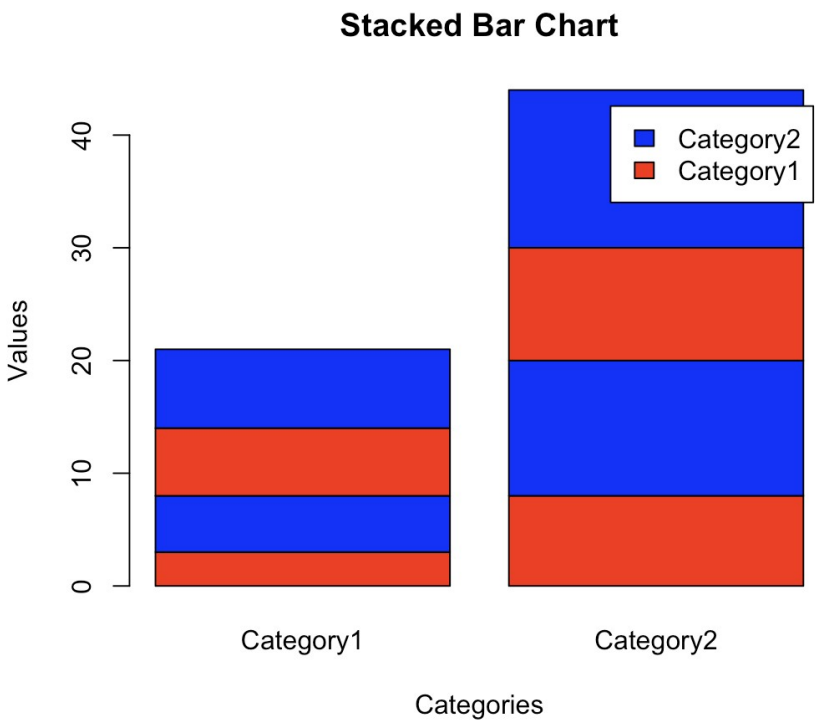
1. barplot(h,x,y,main, names.arg,col)

S.No	Parameter	Description
1.	H	A vector or matrix which contains numeric values used in the bar chart.
2.	xlab	A label for the x-axis.
3.	ylab	A label for the y-axis.
4.	main	A title of the bar chart.
5.	names.arg	A vector of names that appear under each bar.
6.	col	It is used to give colors to the bars in the graph.

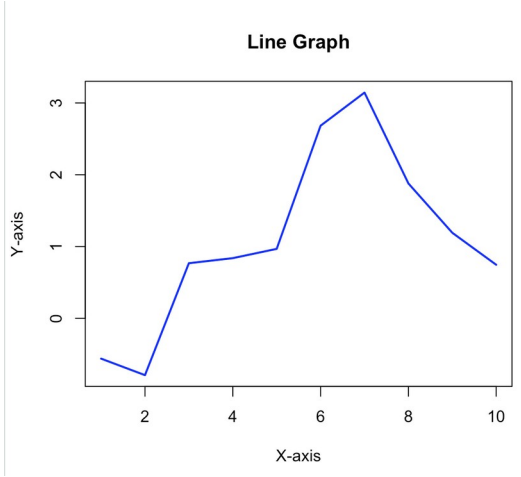
```
heights <- c(3, 8, 5, 12)

labels <- c("A", "B", "C", "D")

colors <- c("red", "blue", "green", "yellow")

# Creating a bar plot

barplot(heights, names.arg = labels, col = colors, main =
"Bar Plot Example")
```



```
data <- matrix(c(3, 8, 5, 12, 6, 10, 7, 14), ncol = 2, byrow = TRUE)

rownames(data) <- c("A", "B", "C", "D")

colnames(data) <- c("Category1", "Category2")


barplot(data, col = c("red", "blue"), main = "Stacked Bar Chart",

xlab = "Categories", ylab = "Values", legend.text = colnames(data))
```



Line Graph




```
install.packages("ggplot2")

# Example data

set.seed(123)

x <- 1:10

y <- cumsum(rnorm(10))

# Create a line graph

plot(x, y, type = "l", col = "blue", main = "Line Graph", xlab = "X-axis", ylab = "Y-axis")
```

plot(x, y, type = "l", col = "blue", main = "Line Graph", xlab = "X-axis", ylab = "Y-axis")

S.No	Parameter	Description
1.	v	It is a vector which contains the numeric values.
2.	type	This parameter takes the value ?l? to draw only the lines or ?p? to draw only the points and "o" to draw both lines and points.
3.	xlab	It is the label for the x-axis.
4.	ylab	It is the label for the y-axis.
5.	main	It is the title of the chart.
6.	col	It is used to give the color for both the points and lines

Graphical Techniques in EDA

Boxplot

Histogram

Pareto Chart

Stem-and-Leaf Plot

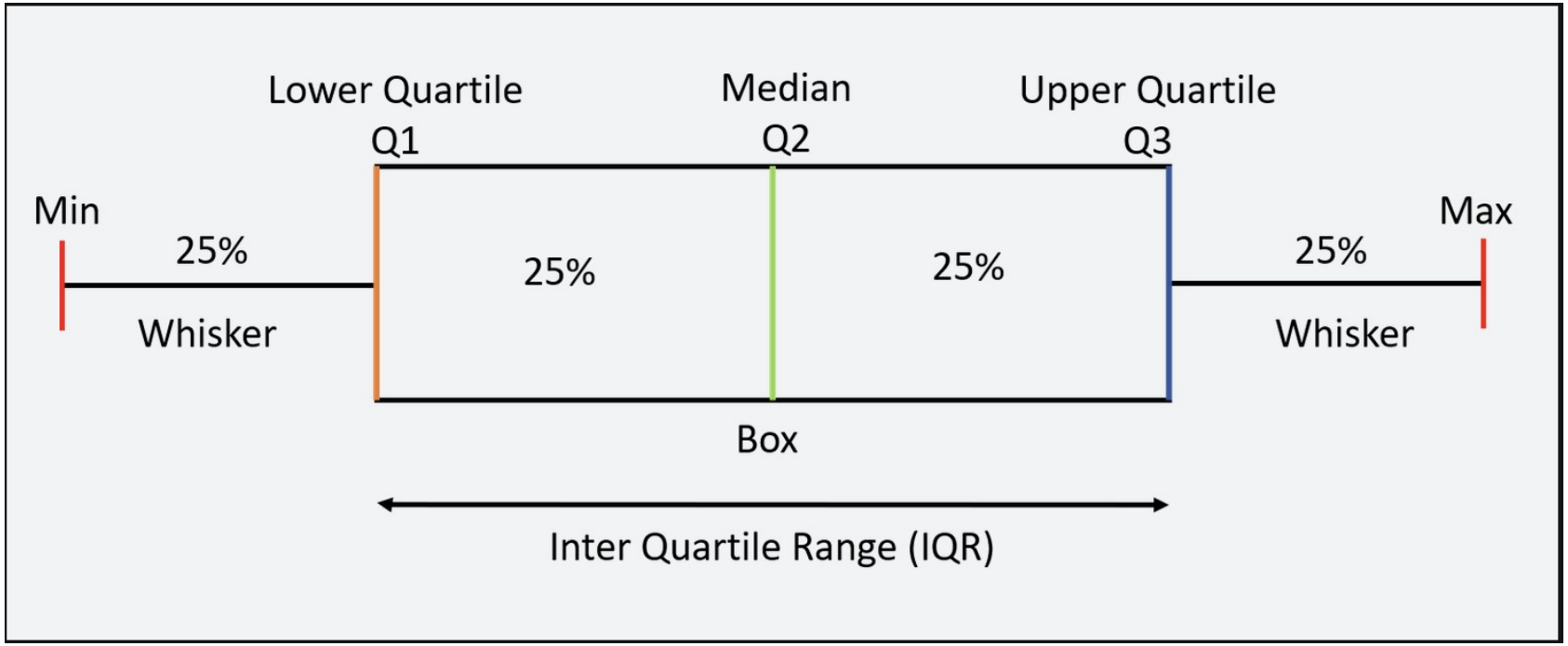
Scatter Plot

Boxplot

The boxplot is essentially a one-dimensional plot, sometimes known as the box-and-whisker plot.

To read Boxplot, first there is a line at the center, this indicates the median of all the data points. Median is the value at the center when you sort the data from the smallest to the largest.

IQR = Q3-Q1


Lower Quartile
Q1
Median
Q2
Upper Quartile
Q3
Min
Max
25%
25%
25%
25%
Whisker
Whisker
Box
Inter Quartile Range (IQR)

2) How to create a box plot

Here are the runs scored by a cricket team in a league of 12 matches – 100,120,110,150,110,140,130,170,120,220,140,110.

To draw a box plot for the given data first we need to arrange the data in ascending order and then find the minimum, first quartile, median, third quartile and the maximum.

Ascending Order -

100,110,110,110,120,120,130,140,140,150,170,220

Median (Q2) = (120+130)/2 = 125 ; Since there were even values

Q1 = (110+110)/2 = 110

Q3 = (140+150)/2 = 145

IQR = Q3-Q1 = 145-110 = 35

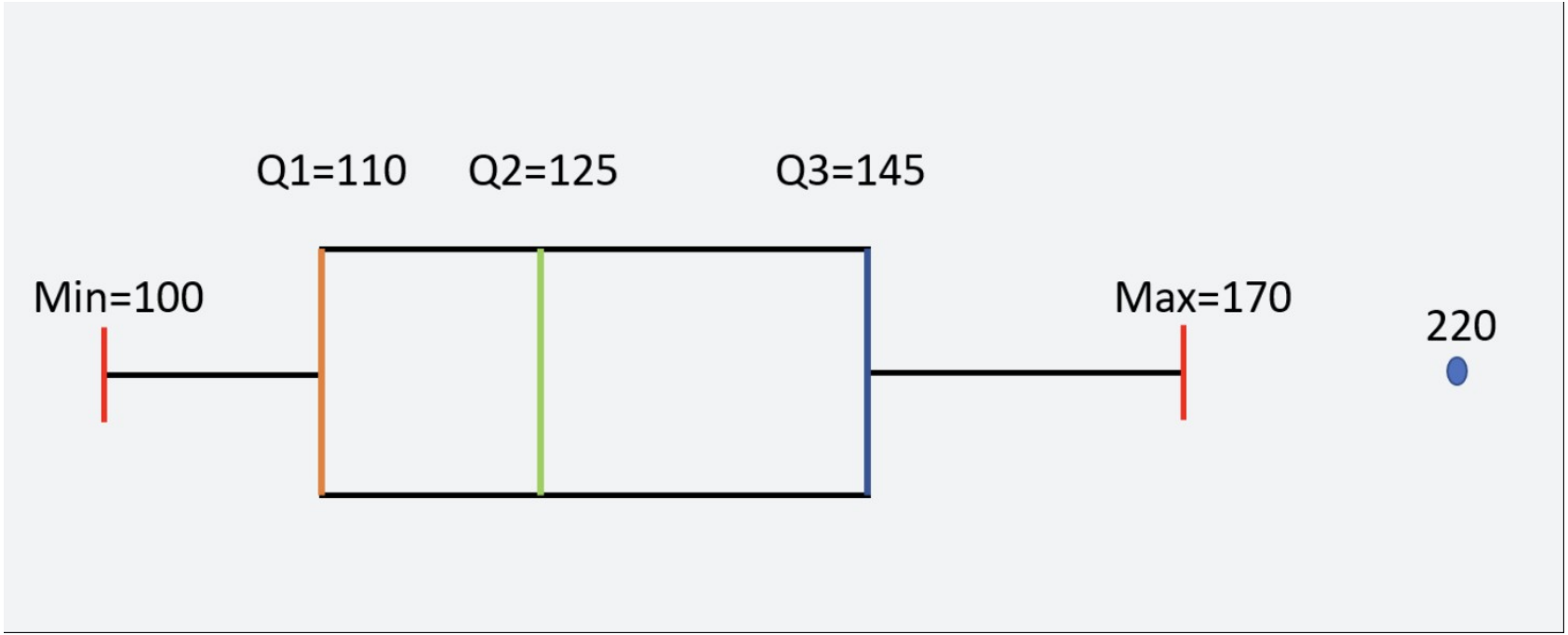
Lower Limit = Q1-1.5*IQR = 110-1.5*35 = 57.5

Upper Limit = Q3+1.5*IQR = 145+1.5*35 = 197.5

Minimum = 100

Maximum = 170

Outliers = 220


Q1=110
Q2=125
Q3=145
Min=100
Max=170
220

The basic construction of the box part of the boxplot is as follows:

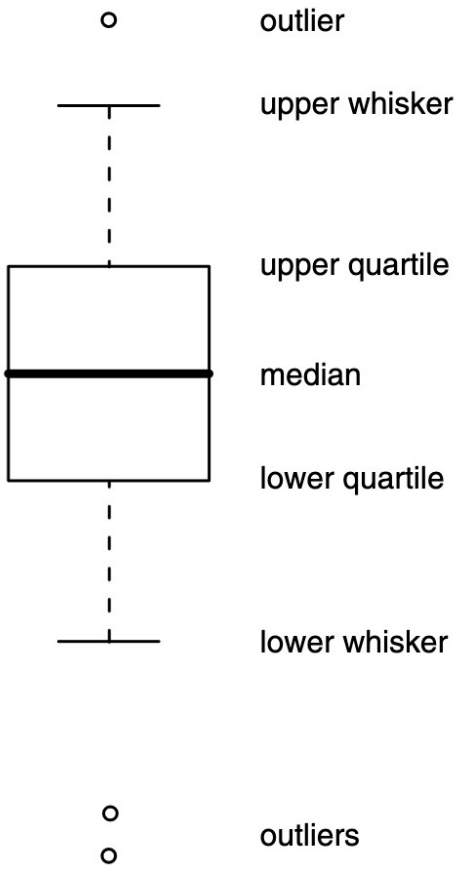
1 A horizontal line is drawn at the median.

2 Split the data into two halves, each containing the median.

3 Calculate the upper and lower quartiles as the medians of each half, and

draw horizontal lines at each of these values. Then connect the lines to

form a rectangular box.



```
> boxplot(Sepal.Length ~ Species, data = iris,
+ ylab = "Sepal length (cm)", main = "Iris measurements",
+ boxwex = 0.5)
```

compares the distributions of the sepal length measurements between the different species. Here we have used R's formula based interface to the graphics function: the syntax **Sepal.Length ~ Species** is read as **"Sepal.Length depending on Species,"** where both are columns of the data frame specified by data = iris. The boxplot() function draws separate side-by-side box plots for each species.