



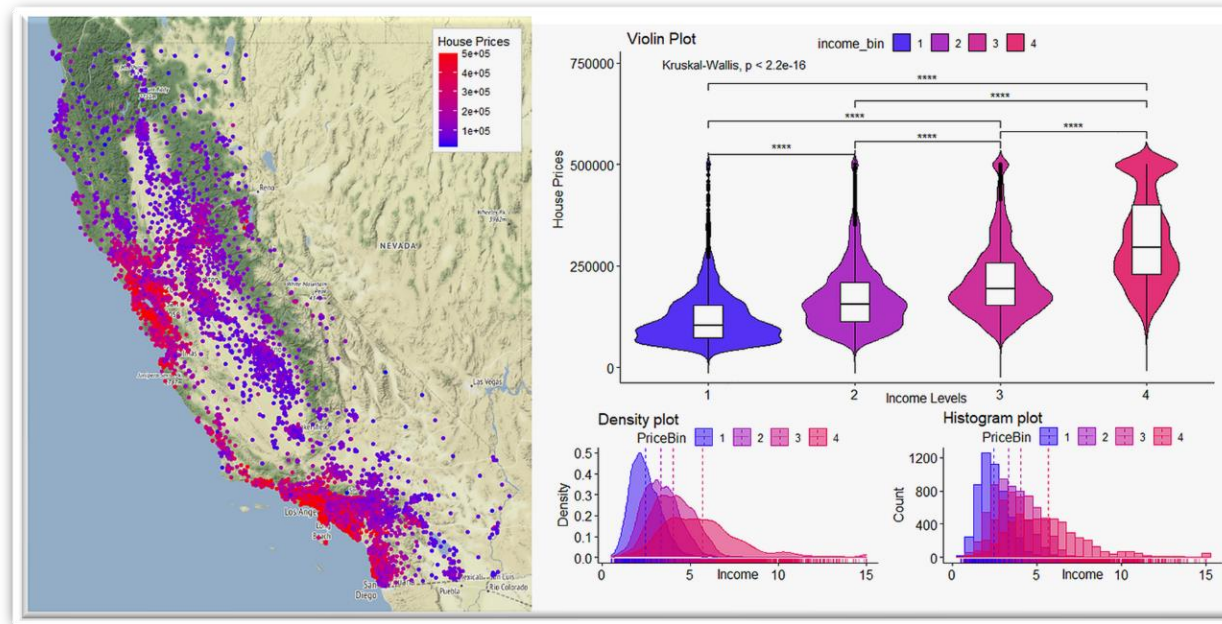
R Graphics

Part of Future Connect Media's IT
Course

By Abdullah Hashmi



Graphs in R language is a preferred feature which is used to create various types of graphs and charts for visualizations. R language supports a rich set of packages and functionalities to create the graphs using the input data set for data analytics.



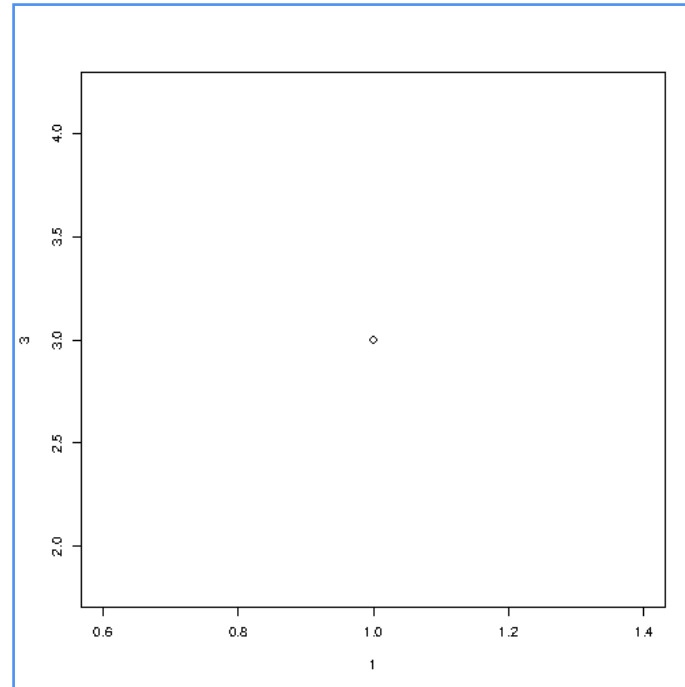
R Plotting

- The `plot()` function is used to draw points (markers) in a diagram.
- The function takes parameters for specifying points in the diagram.
- Parameter 1 specifies points on the **x-axis**.
- Parameter 2 specifies points on the **y-axis**.
- At its simplest, you can use the `plot()` function to plot two numbers against each other.

Example

Draw one point in the diagram, at position (1) and position (3):

- `plot(1, 3)`
- Result:



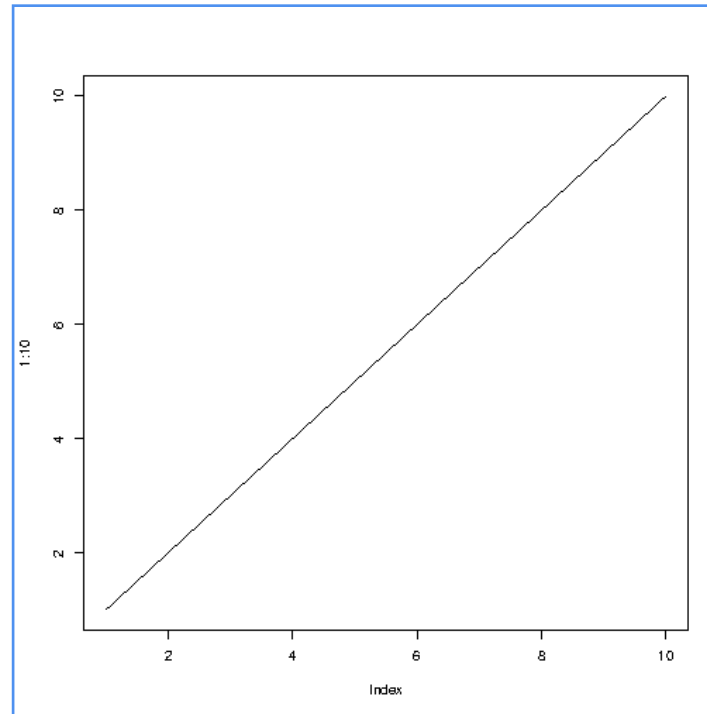
Line Graphs

A line graph has a line that connects all the points in a diagram.

To create a line, use the `plot()` function and add the `type` parameter with a value of `"l"`:

Example:

```
plot(1:10, type="l")
```



Multiple Lines

To display more than one line in a graph, use the `plot()` function together with the `lines()` function

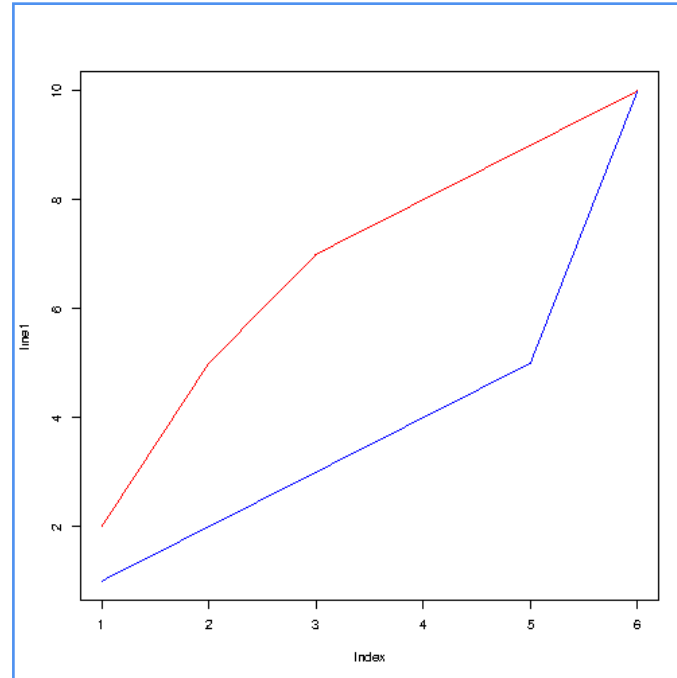
Example:

```
line1 <- c(1,2,3,4,5,10)
```

```
line2 <- c(2,5,7,8,9,10)
```

```
plot(line1, type = "l", col = "blue")
```

```
lines(line2, type="l", col = "red")
```



Scatter Plots

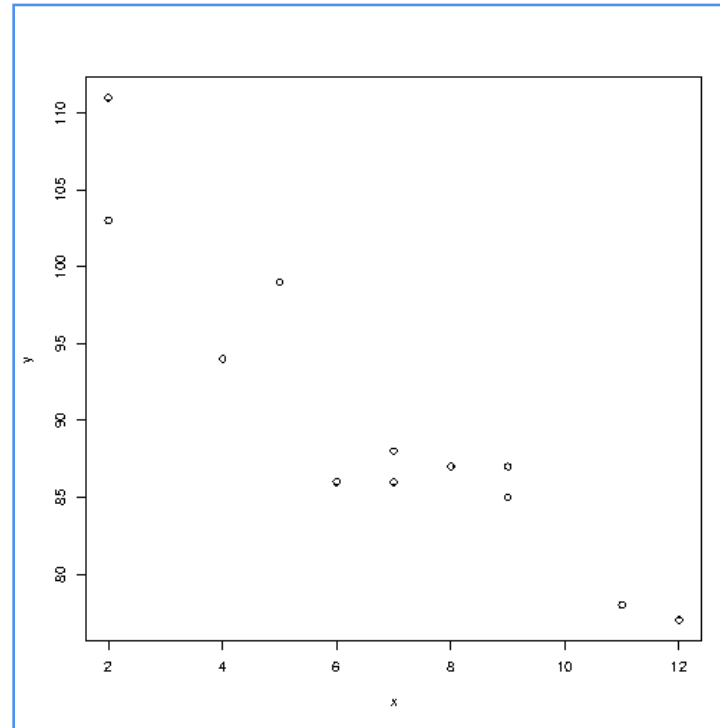
- You learned from the Plot chapter that the `plot()` function is used to plot numbers against each other.
- A "scatter plot" is a type of plot used to display the relationship between two numerical variables, and plots one dot for each observation.
- It needs two vectors of same length, one for the x-axis (horizontal) and one for the y-axis (vertical):

Example

```
x <- c(5,7,8,7,2,2,9,4,11,12,9,6)
```

```
y <- c(99,86,87,88,111,103,87,94,78,77,85,86)
```

```
plot(x, y)
```



Pie Charts

A pie chart is a circular graphical view of data.

Use the `pie()` function to draw pie charts:

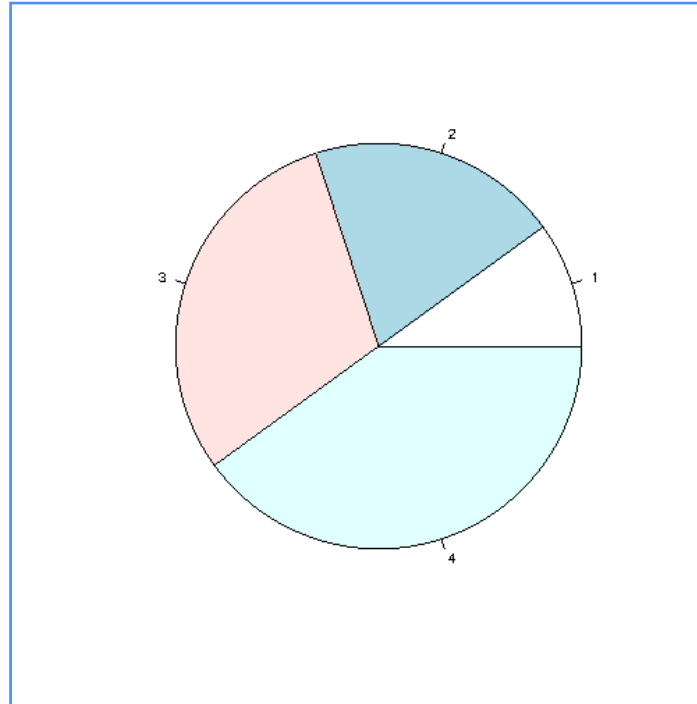
Example:

```
# Create a vector of pies
```

```
x <- c(10,20,30,40)
```

```
# Display the pie chart
```

```
pie(x)
```



Labels & Header

- Use the **label** parameter to add a label to the pie chart, use the **main** parameter to add a header.

Example:

Create a vector of pies

```
x <- c(10,20,30,40)
```

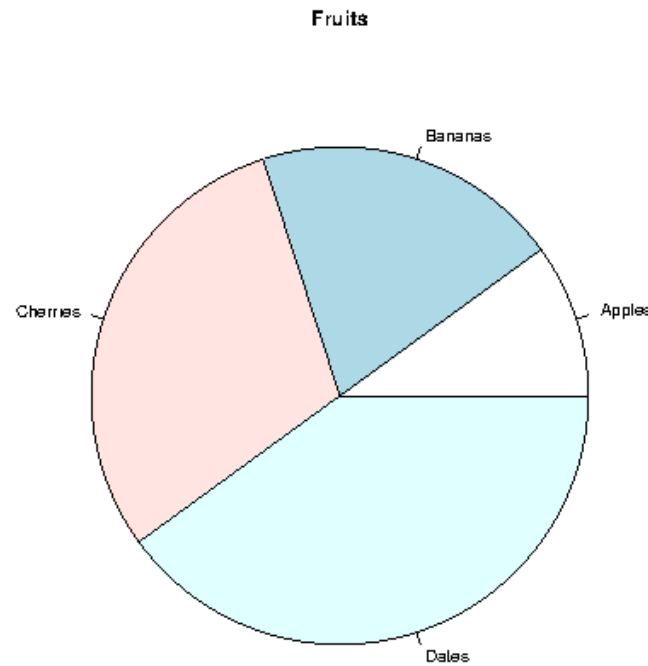
Create a vector of labels

```
mylabel <-
```

```
c("Apples", "Bananas", "Cherries", "Dates")
```

Display the pie chart with labels

```
pie(x, label = mylabel, main = "Fruits")
```



Colors

You can add a color to each pie with the `col` parameter:

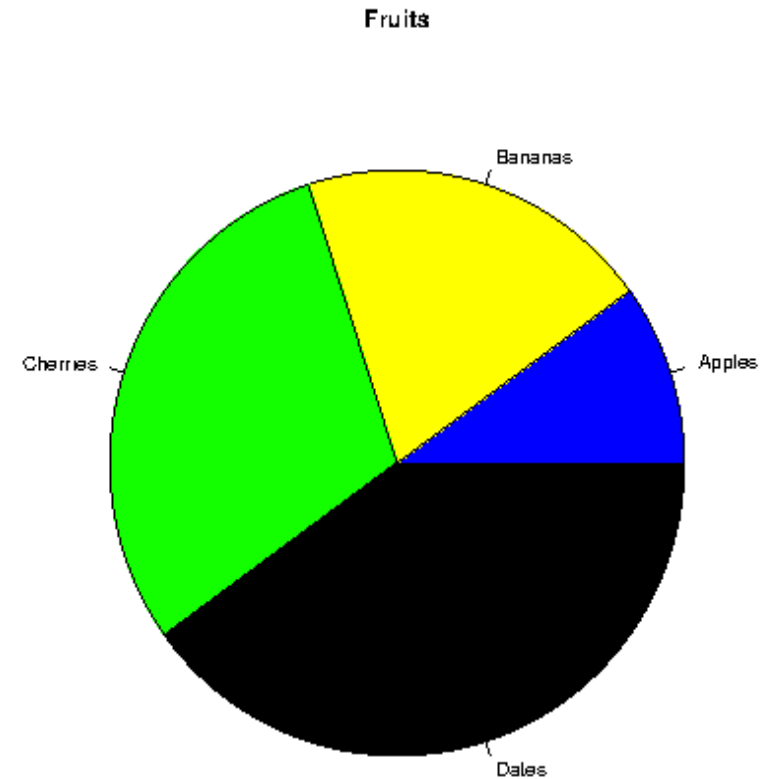
Example:

Create a vector of colors

```
colors <- c("blue", "yellow", "green", "black")
```

Display the pie chart with colors

```
pie(x, label = mylabel, main = "Fruits", col = colors)
```



Bar Charts

- A bar chart uses rectangular bars to visualize data. Bar charts can be displayed horizontally or vertically. The height or length of the bars are proportional to the values they represent.
- Use the `barplot()` function to draw a vertical bar chart:

Example:

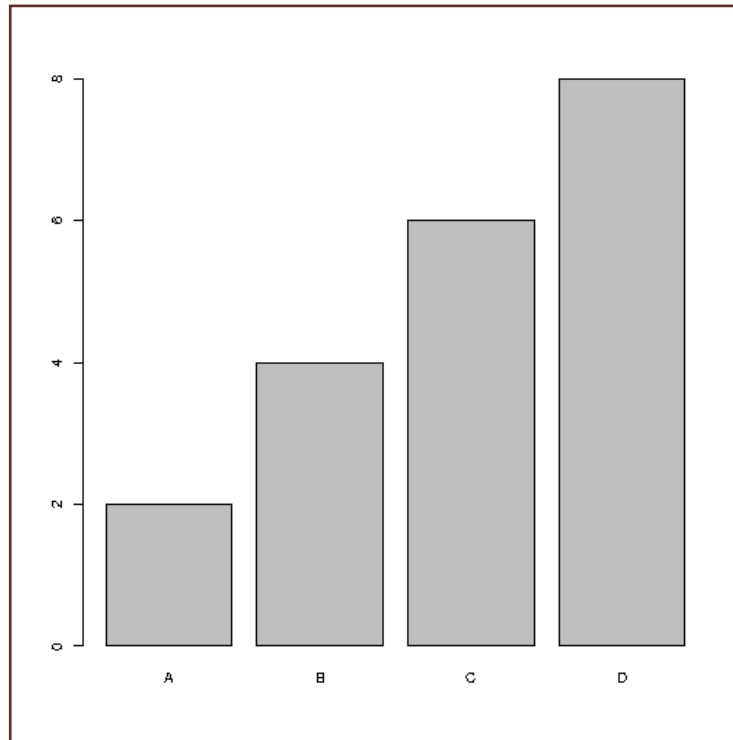
x-axis values

```
x <- c("A", "B", "C", "D")
```

y-axis values

```
y <- c(2, 4, 6, 8)
```

```
barplot(y, names.arg = x)
```



Density & Bar Texture

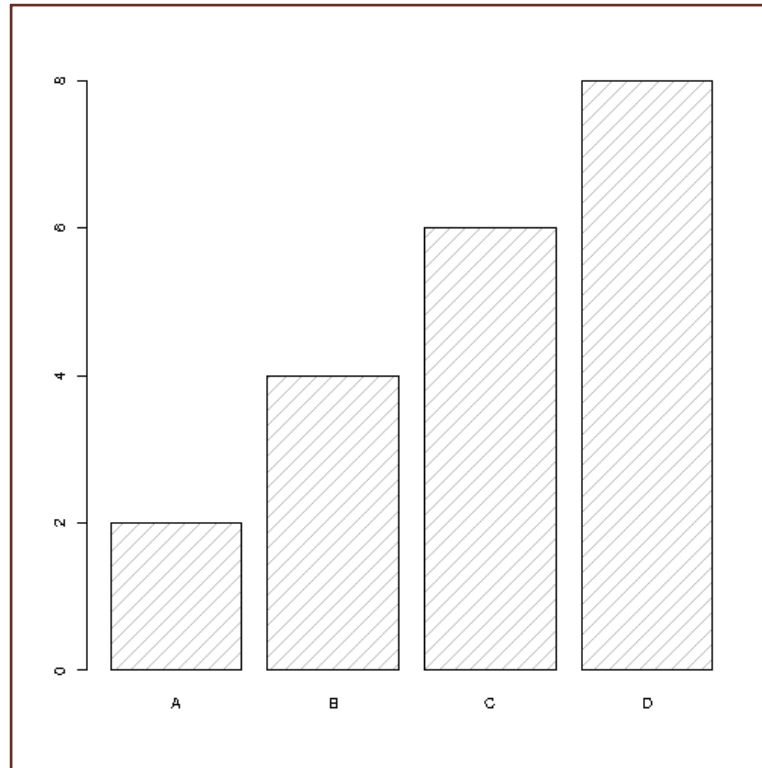
To change the bar texture, use the **density** parameter:

Example:

```
x <- c("A", "B", "C", "D")
```

```
y <- c(2, 4, 6, 8)
```

```
barplot(y, names.arg = x, density = 10)
```



The background features several decorative elements: a large blue circle on the left containing the text 'Thank you'; a purple circle in the top left; an orange L-shaped line in the top right; a green circle on the far left; and a blue circle with green curved lines in the bottom center.

Thank you

Future Connect Training Institute

Website: <https://www.fctraining.co.uk/>