

Charts and Graphs in R

Tushar B. Kute,
<http://tusharkute.com>

Charts and Graphs supported

- Pie chart
- Bar chart
- Box plots
- Histograms
- Line graphs
- Scatter plots

Pie charts

- R Programming language has numerous libraries to create charts and graphs.
- 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.
- In R the pie chart is created using the `pie()` function which takes positive numbers as a vector input.
- The additional parameters are used to control labels, color, title etc.

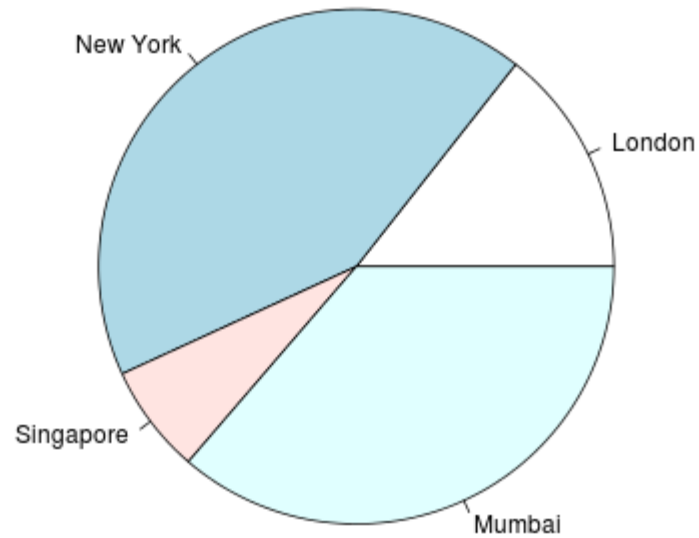
Pie charts – Syntax

- The basic syntax for creating a pie-chart using the R is –
`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.

Pie charts – Example

```
# 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.png")  
  
# Plot the chart.  
pie(x, labels)  
  
# Save the file.  
dev.off()
```

Pie charts – Example

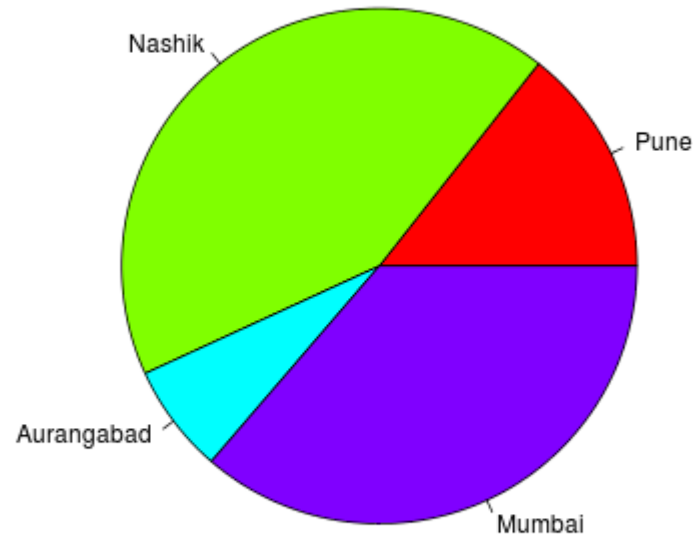


Pie chart example with colors

```
# Create data for the graph.  
x <- c(21, 62, 10, 53)  
labels <- c("Pune", "Nashik", "Aurangabad", "Mumbai")  
  
# Give the chart file a name.  
png(file = "city_title_colours.png")  
  
# Plot the chart with title and rainbow color pallet.  
pie(x, labels, main = "City pie chart", col =  
rainbow(length(x)))  
  
# Save the file.  
dev.off()
```

Pie chart example with colors

City pie chart



Pie chart with colors and labels

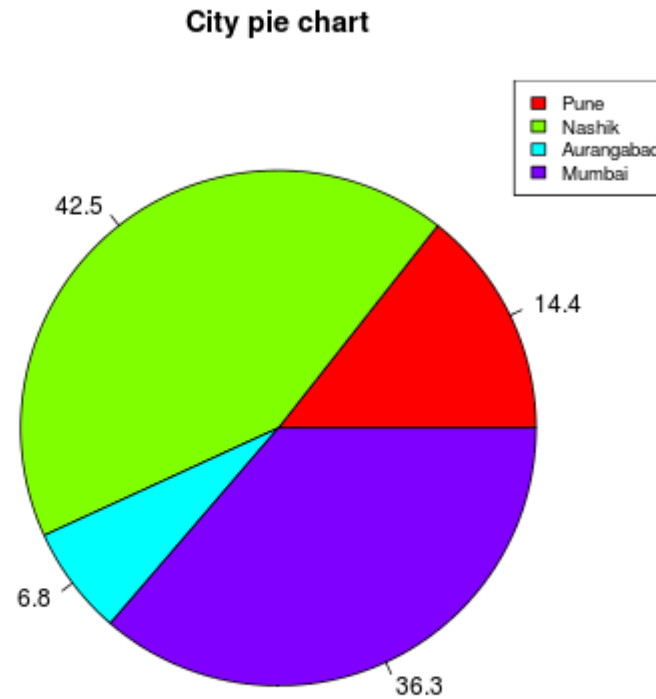
```
# Create data for the graph.
x <- c(21, 62, 10, 53)
labels <- c("London", "New York", "Singapore", "Mumbai")

piepercent<- round(100*x/sum(x), 1)
png(file = "city_percentage_legends.png")

# Plot the chart.
pie(x, labels = piepercent, main = "City pie chart", col =
rainbow(length(x)))
legend("topright", c("Pune", "Nashik", "Aurangabad", "Mumbai"),
cex = 0.8, fill = rainbow(length(x)))

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

Pie chart with colors and labels



3D Pie Chart

```
# Get the library.
library(plotrix)

# Create data for the graph.
x <- c(21, 62, 10, 53)
lbl <- c("Nashik", "Aurangabad", "Navi Mumbai", "Nagpur")

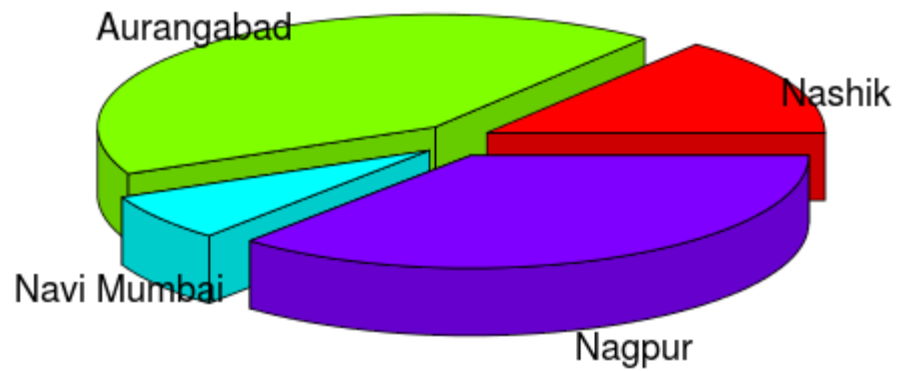
png(file = "3d_pie_chart.png")

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

dev.off()
```

3D Pie Chart

Pie Chart of Countries



Bar charts

- 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.
- In bar chart each of the bars can be given different colors.

Bar charts – Syntax

- The basic syntax to create a bar-chart in R is –
`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.

Bar charts – Example

```
# Create the data for the chart.
```

```
H <- c(7,12,28,3,41)
```

```
# Give the chart file a name.
```

```
png(file = "barchart.png")
```

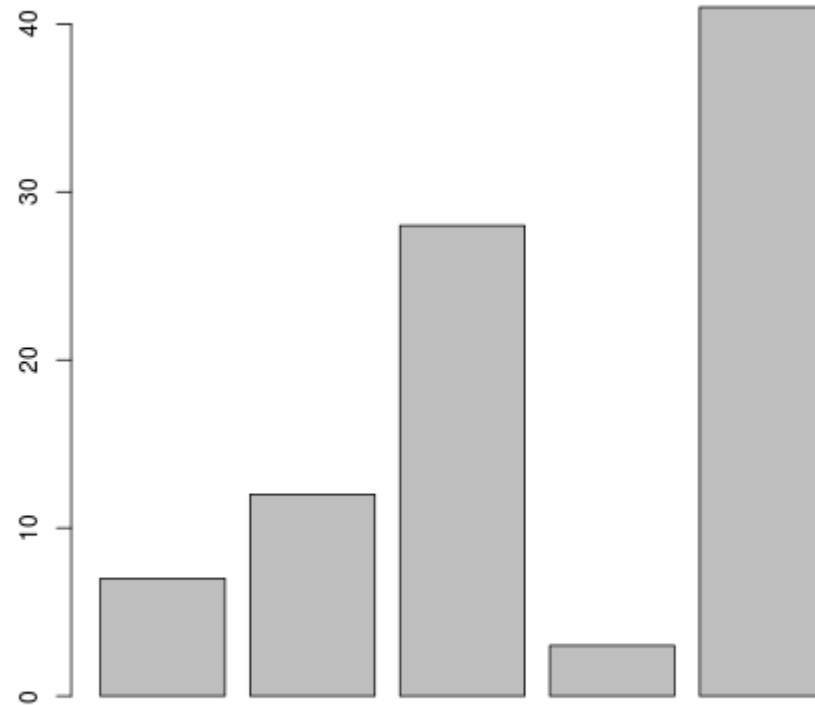
```
# Plot the bar chart.
```

```
barplot(H)
```

```
# Save the file.
```

```
dev.off()
```

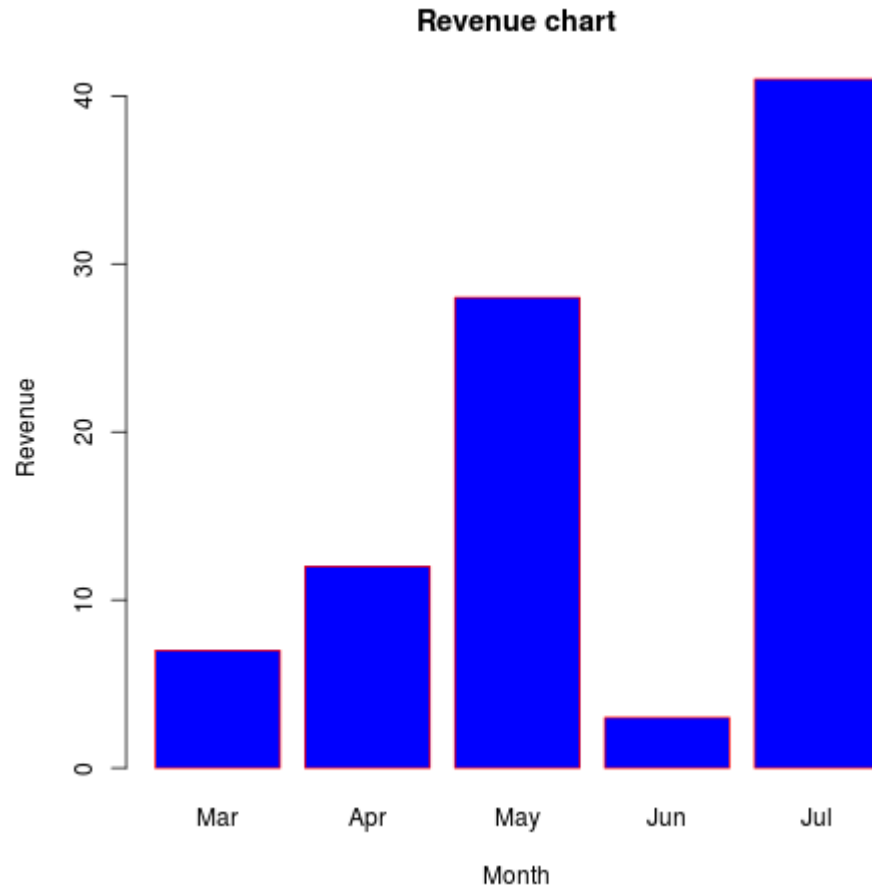
Bar charts – Example



Bar chart with attributes

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

Bar chart with attributes



Bar chart – Stacked

```
colors <- c("green", "orange", "brown")
months <- c("Mar", "Apr", "May", "Jun", "Jul")
regions <- c("East", "West", "North")

Values <- matrix(c(2, 9, 3, 11, 9, 4, 8, 7, 3, 12, 5, 2, 8, 10, 11), nrow =
3, ncol = 5, byrow = TRUE)

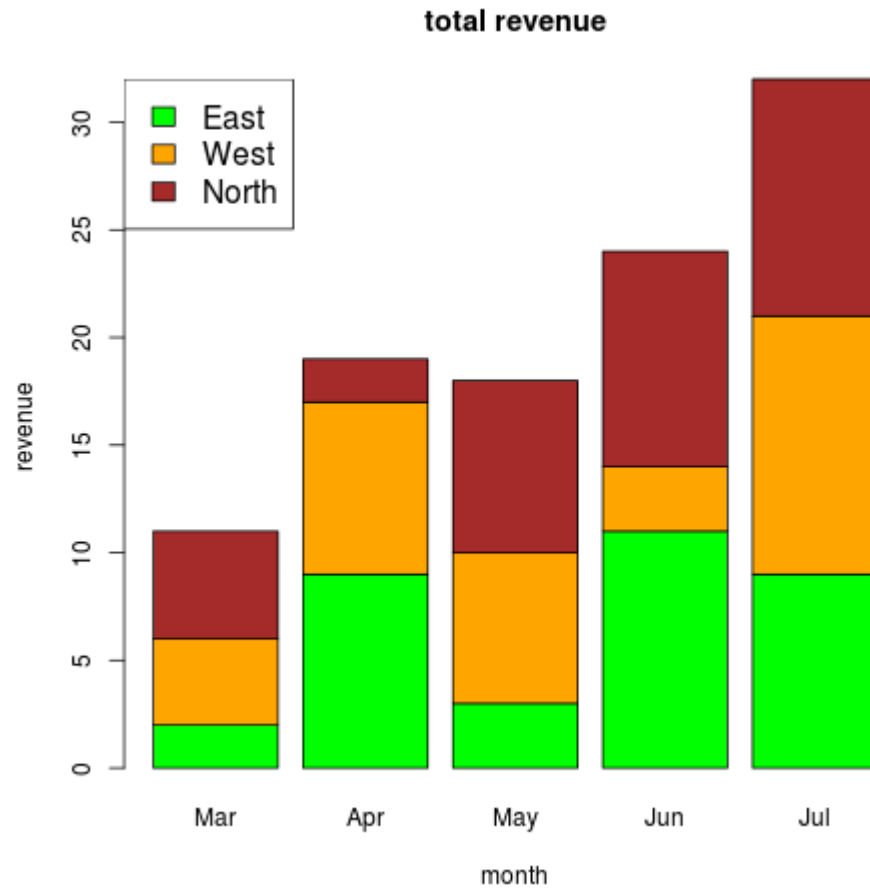
png(file = "barchart_stacked.png")

barplot(Values, main = "total revenue", names.arg = months, xlab =
"month", ylab = "revenue", col = colors)

legend("topleft", regions, cex = 1.3, fill = colors)

dev.off()
```

Bar chart – Stacked



Boxplot

- 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.
- Boxplots are created in R by using the `boxplot()` function.

Boxplot – Syntax

- The basic syntax to create a boxplot in R is –
`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.

Boxplot – Example

- We use the data set "mtcars" available in the R environment to create a basic boxplot. Let's look at the columns "mpg" and "cyl" in mtcars.

```
input <- mtcars[,c('mpg', 'cyl')]
print(head(input))
```

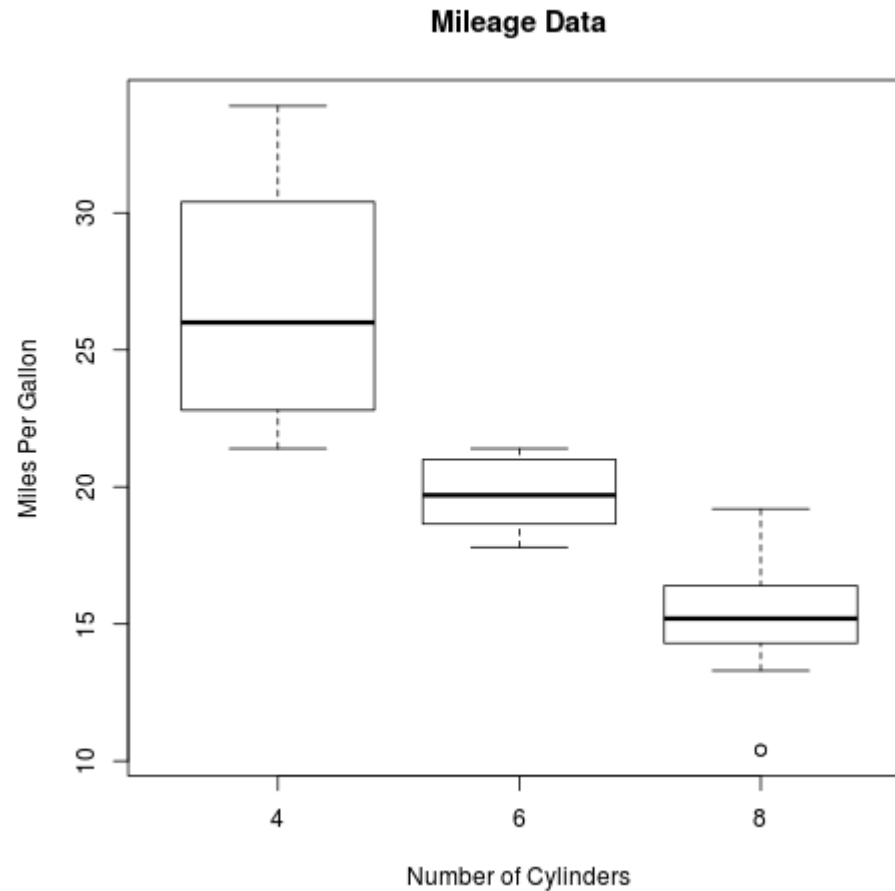
```
> input <- mtcars[,c('mpg','cyl')]
> print(head(input))
```

	mpg	cyl
Mazda RX4	21.0	6
Mazda RX4 Wag	21.0	6
Datsun 710	22.8	4
Hornet 4 Drive	21.4	6
Hornet Sportabout	18.7	8
Valiant	18.1	6

Boxplot – Example

```
# Give the chart file a name.  
png(file = "boxplot.png")  
  
# Plot the chart.  
boxplot(mpg ~ cyl, data = mtcars, xlab =  
"Number of Cylinders", ylab = "Miles  
Per Gallon", main = "Mileage Data")  
  
# Save the file.  
dev.off()
```


Boxplot – Example

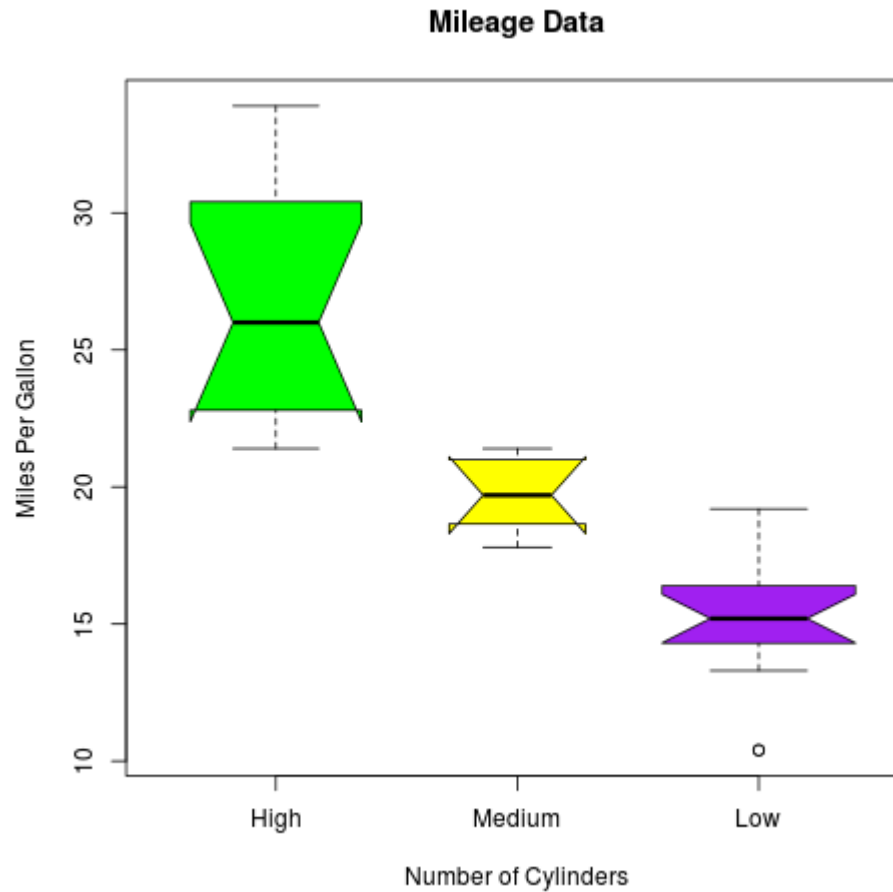


Boxplot with notch

```
png(file = "boxplot_with_notch.png")

# 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")
)
dev.off()
```

Boxplot with notch



Histogram

- A histogram represents the frequencies of values of a variable bucketed into ranges.
- Histogram is similar to bar chart 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.
- R creates histogram using `hist()` function. This function takes a vector as an input and uses some more parameters to plot histograms.

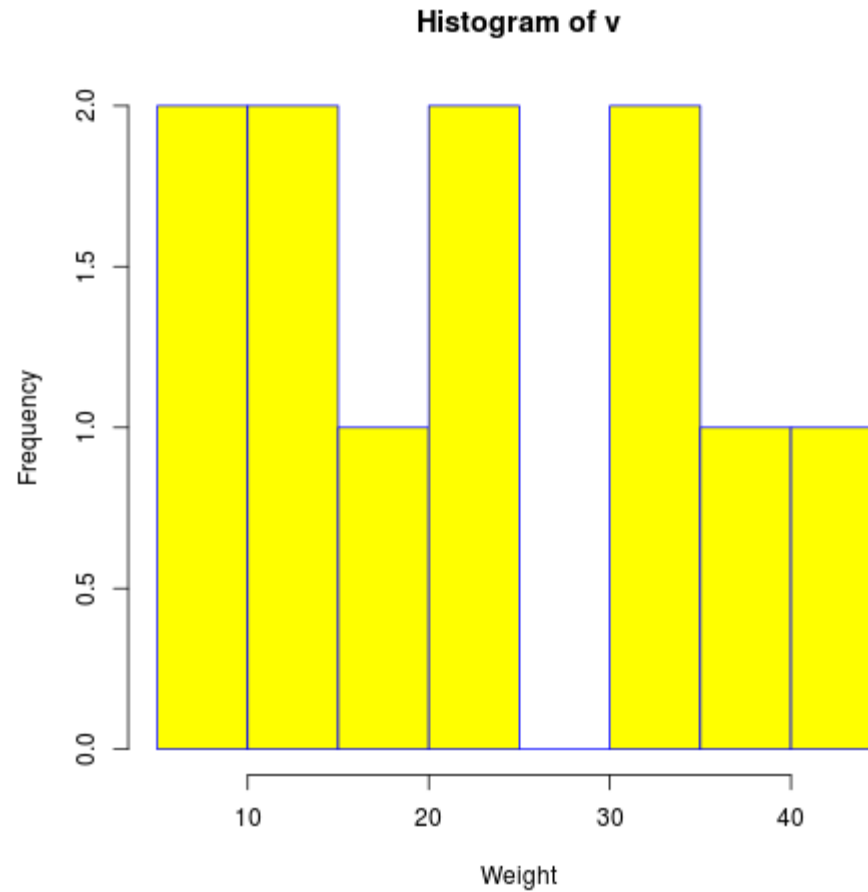
Histogram – Syntax

- The basic syntax for creating a histogram using R is –
`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.

Histogram – Example

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

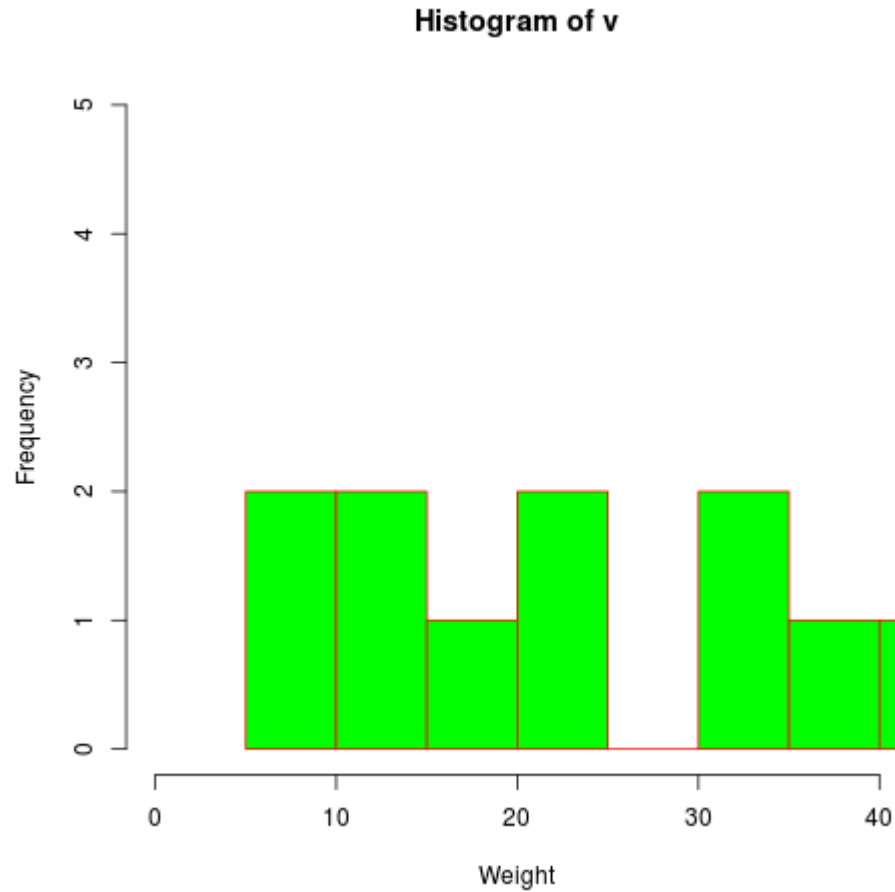
Histogram – Example



Histogram – Example

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


Histogram – Example



Line graph

- 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.
- The `plot()` function in R is used to create the line graph.

Line graph – Syntax

- The basic syntax to create a line chart in R is –
`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.

Line graph – Example

```
# Create the data for the chart.
```

```
v <- c(7,12,28,3,41)
```

```
# Give the chart file a name.
```

```
png(file = "line_chart.png")
```

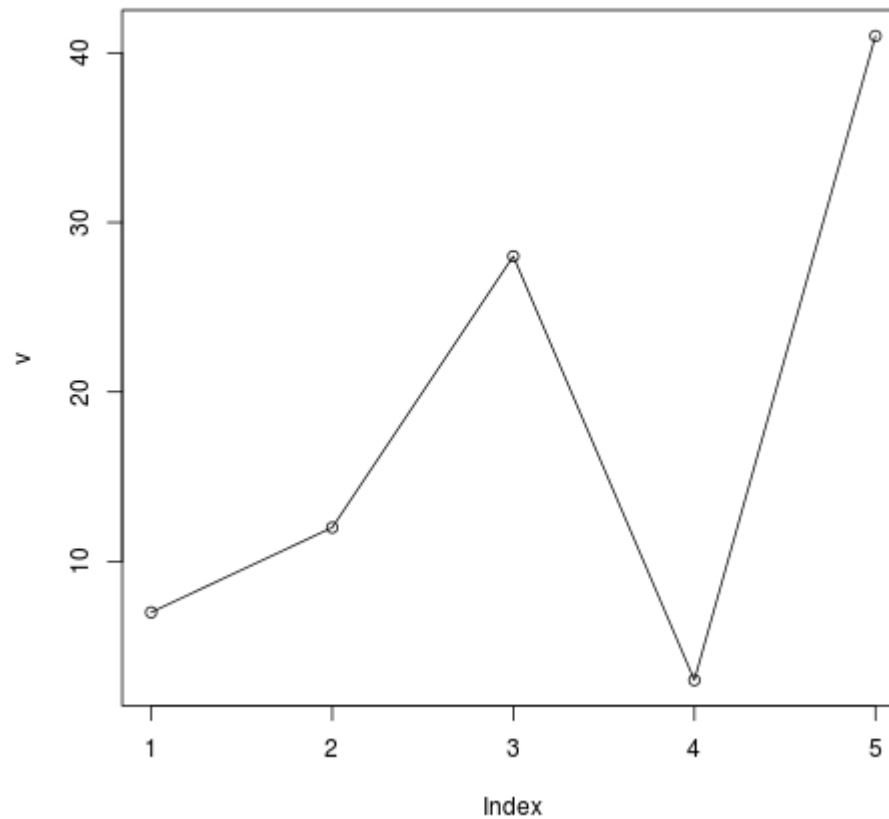
```
# Plot the line graph.
```

```
plot(v,type = "o")
```

```
# Save the file.
```

```
dev.off()
```

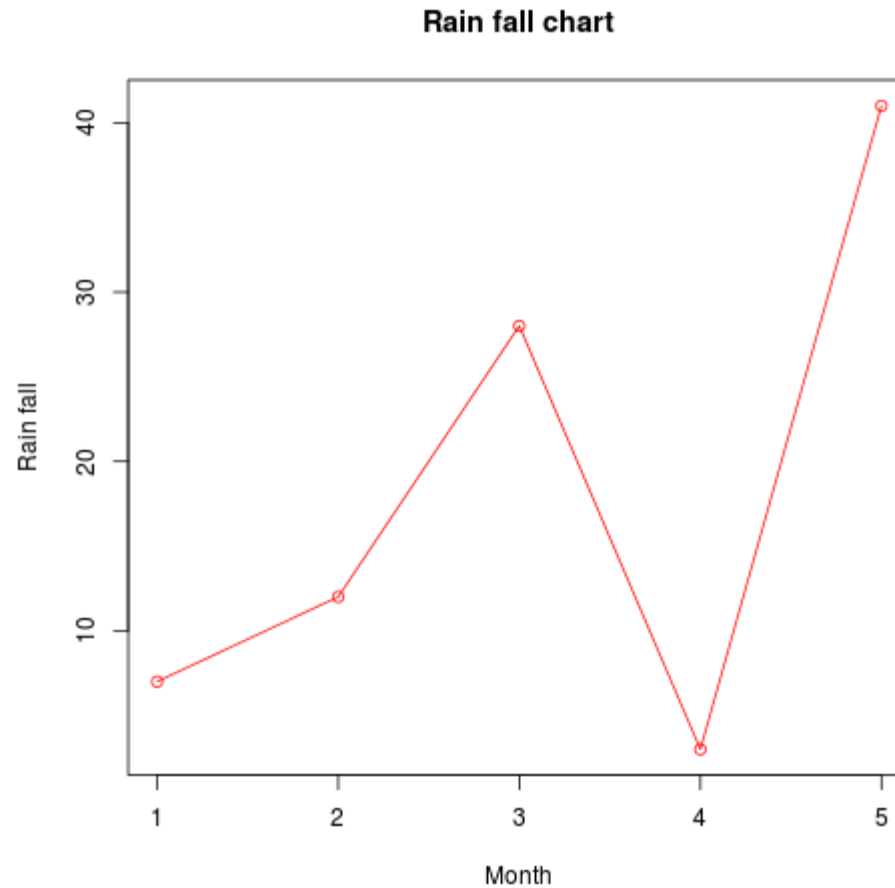
Line graph – Example



Line graph – example.

```
# Create the data for the chart.  
v <- c(7,12,28,3,41)  
  
# Give the chart file a name.  
png(file = "line_chart_label_colored.png")  
  
# Plot the bar chart.  
plot(v,type = "o", col = "red", xlab = "Month",  
ylab = "Rain fall", main = "Rain fall chart")  
  
# Save the file.  
dev.off()
```

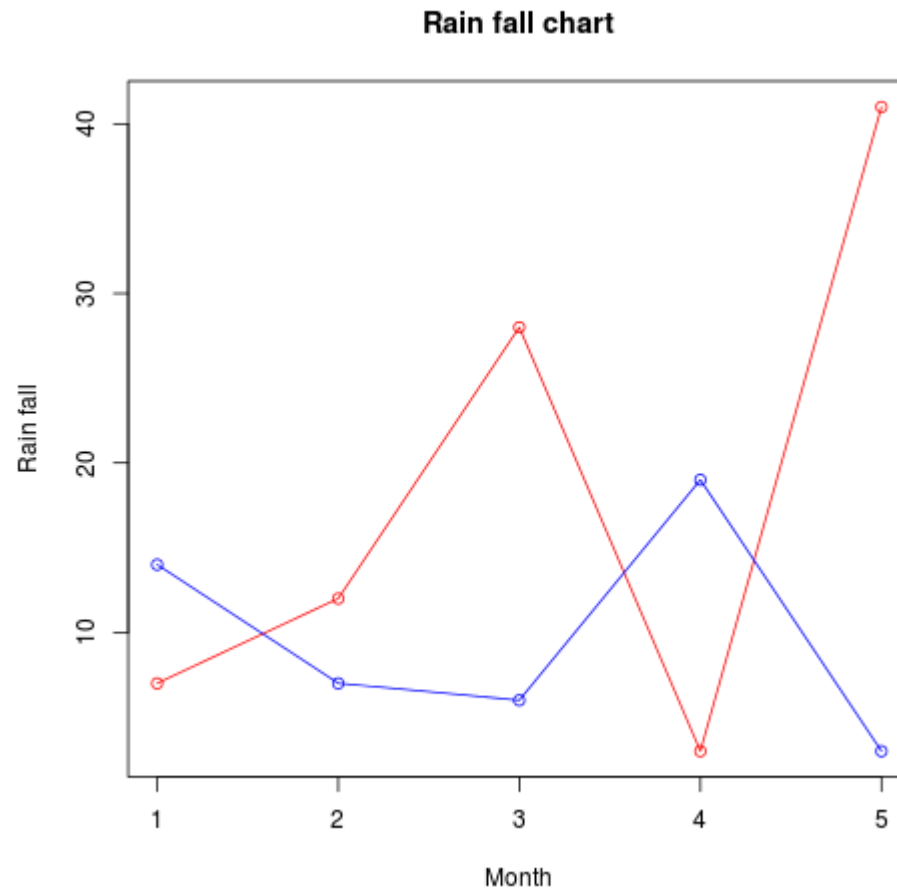
Line graph – example.



Multiple lines in chart

```
# Create the data for the chart.  
v <- c(7,12,28,3,41)  
t <- c(14,7,6,19,3)  
  
# Give the chart file a name.  
png(file = "line_chart_2_lines.png")  
  
# 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")  
  
dev.off()
```


Multiple lines in chart



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.
- The simple scatterplot is created using the `plot()` function.

ScatterPlot – Example

- We use the data set "mtcars" available in the R environment to create a basic scatterplot.
- Let's use the columns "wt" and "mpg" in mtcars.

```
input <- mtcars[,c('wt', 'mpg')]
print(head(input))
```

```
Console ~/ ↵
> input <- mtcars[,c('wt', 'mpg')]
> print(head(input))
      wt  mpg
Mazda RX4      2.620 21.0
Mazda RX4 Wag  2.875 21.0
Datsun 710     2.320 22.8
Hornet 4 Drive 3.215 21.4
Hornet Sportabout 3.440 18.7
Valiant       3.460 18.1
```

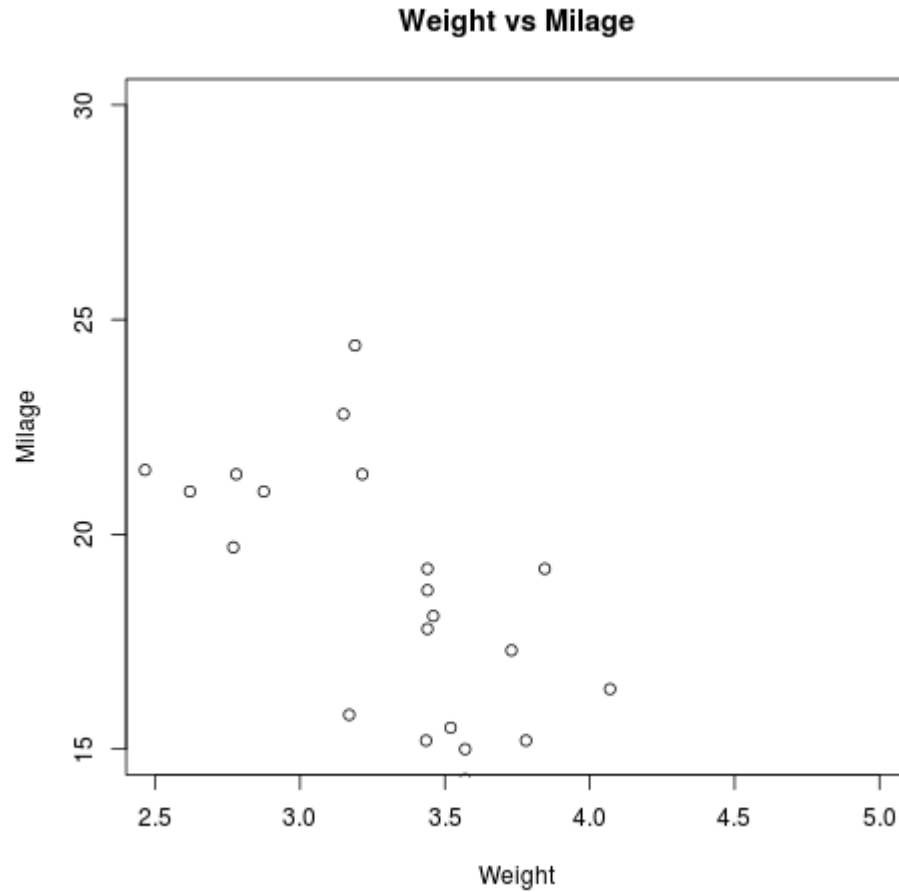
ScatterPlot – Example

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

png(file = "scatterplot.png")

# 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"
)
dev.off()
```

ScatterPlot – Example



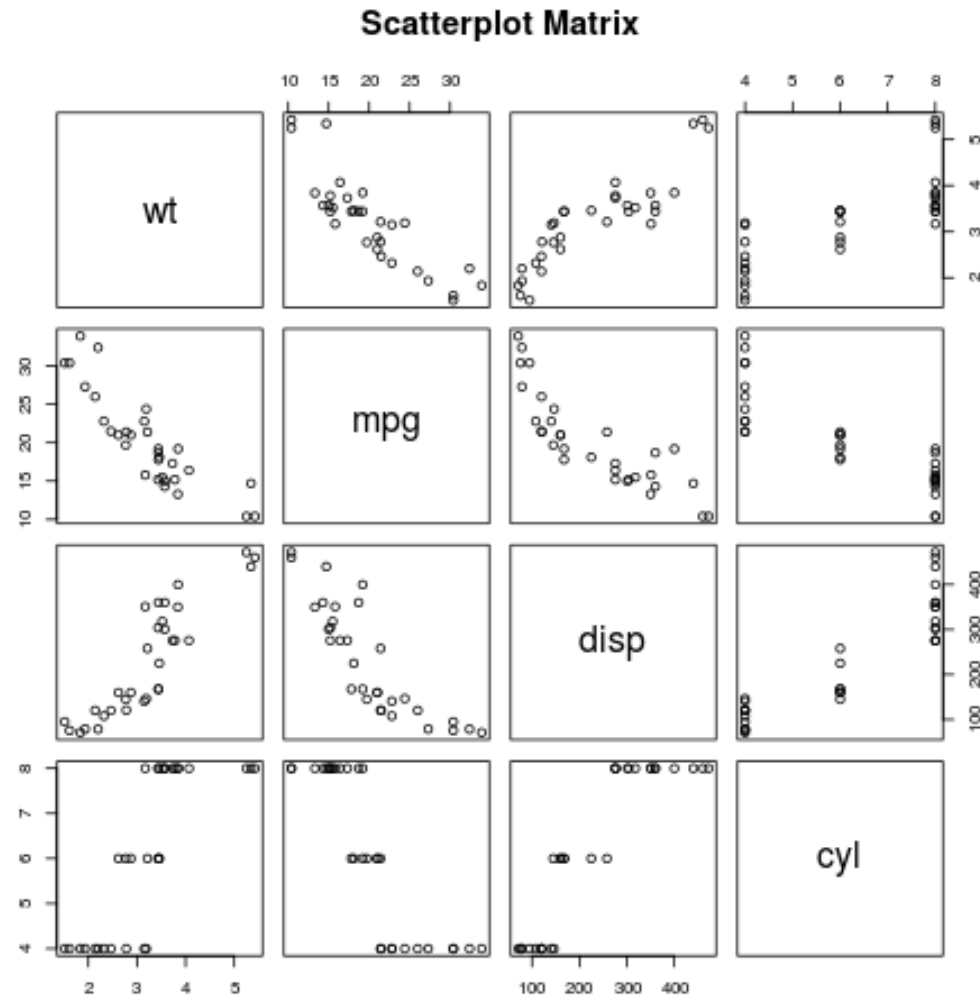
Scatter Plot matrices

- When we have more than two variables and we want to find the correlation between one variable versus the remaining ones we use scatterplot matrix.
- We use `pairs()` function to create matrices of scatterplots.
 - Syntax:
`pairs(formula, data)`
 - Following is the description of the parameters used –
formula represents the series of variables used in pairs.
data represents the data set from which the variables will be taken.

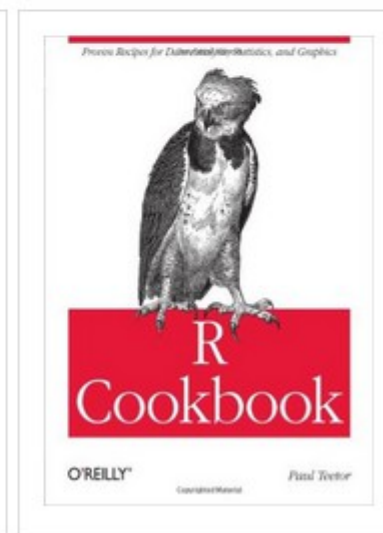
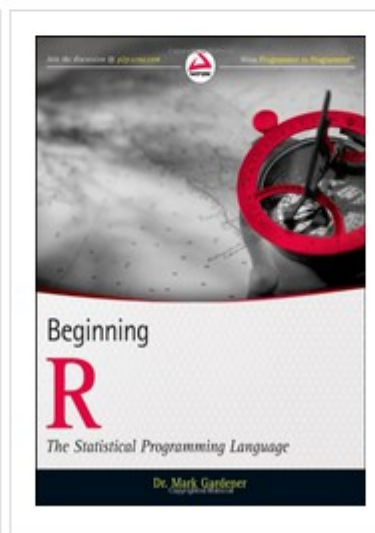
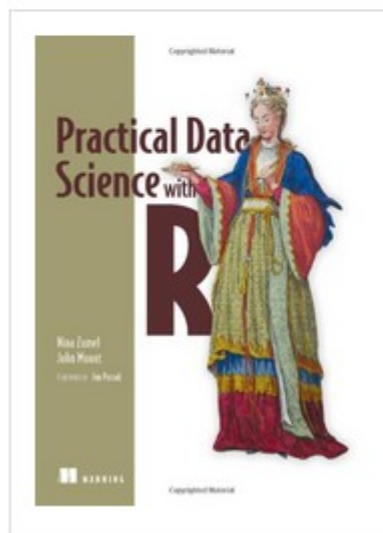
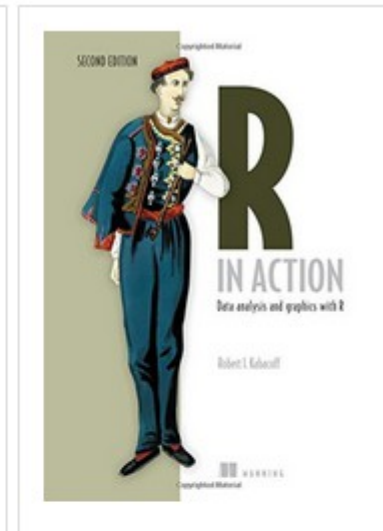
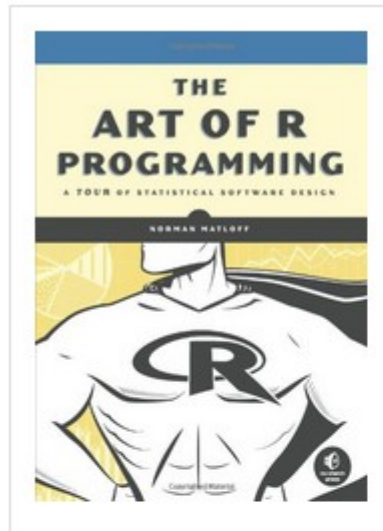
Scatter Plot matrices – Example

```
# Give the chart file a name.  
png(file = "scatterplot_matrices.png")  
  
# 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")  
  
dev.off()
```

Scatter Plot matrices – Example



Useful resources



Thank you

This presentation is created using LibreOffice Impress 4.2.8.2, can be used freely as per GNU General Public License

Web Resources

<http://mitu.co.in>
<http://tusharkute.com>

Blogs

<http://digitallocha.blogspot.in>
<http://kyamputar.blogspot.in>

tushar@tusharkute.com