LINE CHART:

What is a Line chart:

* A line chart is a type of chart used to show information that changes over time.
* Line charts are created by plotting a series of several points and connecting them with a straight line.
* Line charts are used to track changes over short and long periods.

When to use Line Chart:

* Data that is measured in a continuous progression works well in a line chart format.
* If your organization wants to track the behavior of data over a given period, line charts can be of great help.
* The User will see changes in the data, plotted out with a line connecting each data point as they changed over time.

Types of Line Chart:

* Simple Line Chart
* Multiple Line Chart
* Compound Line Chart

Implement all the types of Line Chart in R Code:

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

Syntax:

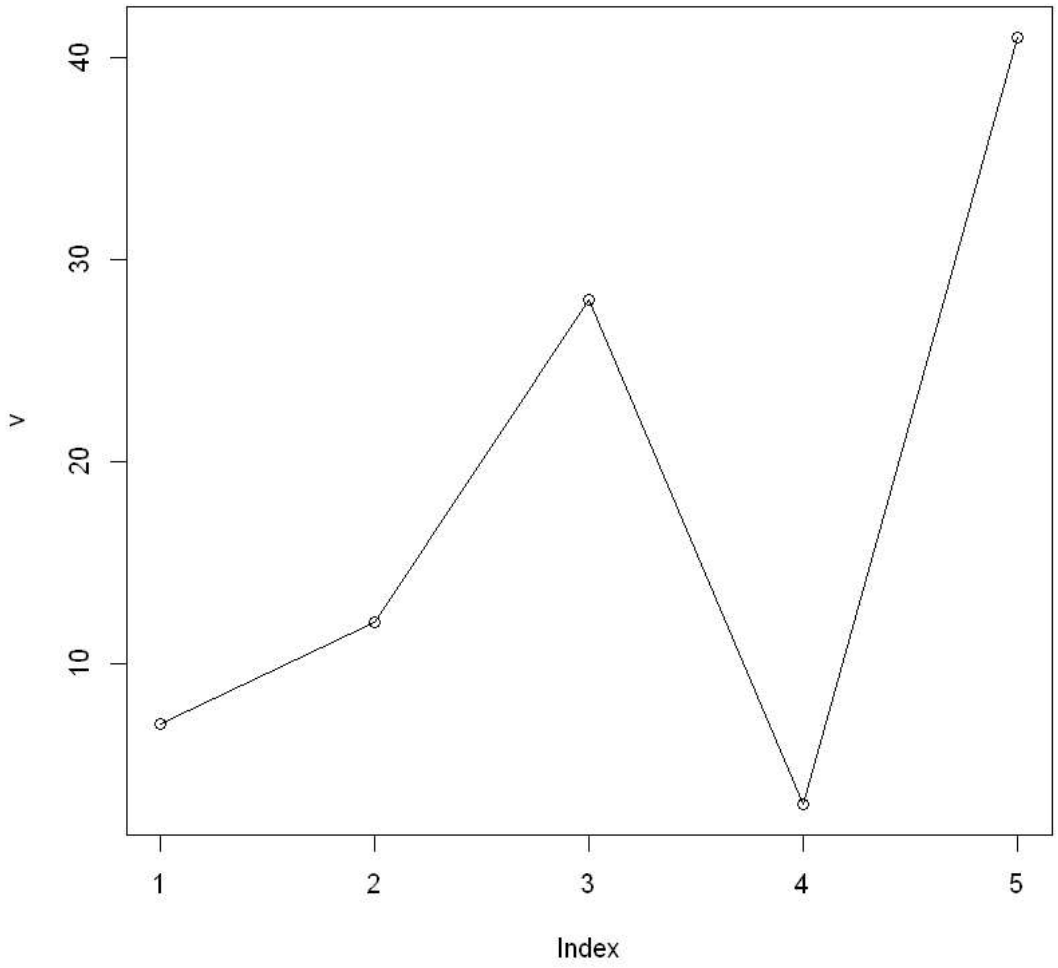
plot(v,type,col,xlab,ylab)

* "v" -- is a vector containing the numeric values.
* "type" -- takes the value
  - > "p" to draw only the points
  - > "l" to draw only the lines
  - > "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.

Simple Line Chart:

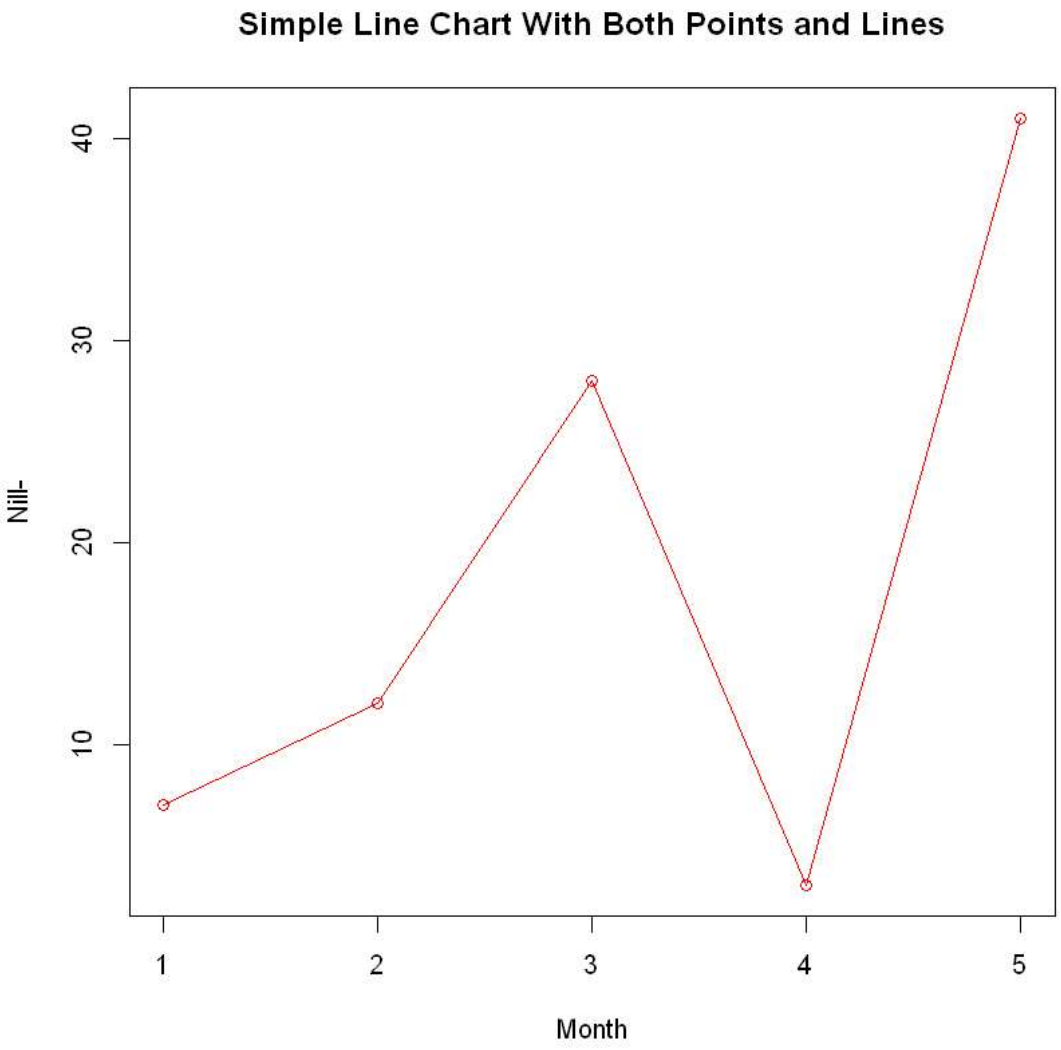
In [5]:

```
v <- c(7,12,28,3,41)
plot(v,type = "o")
```



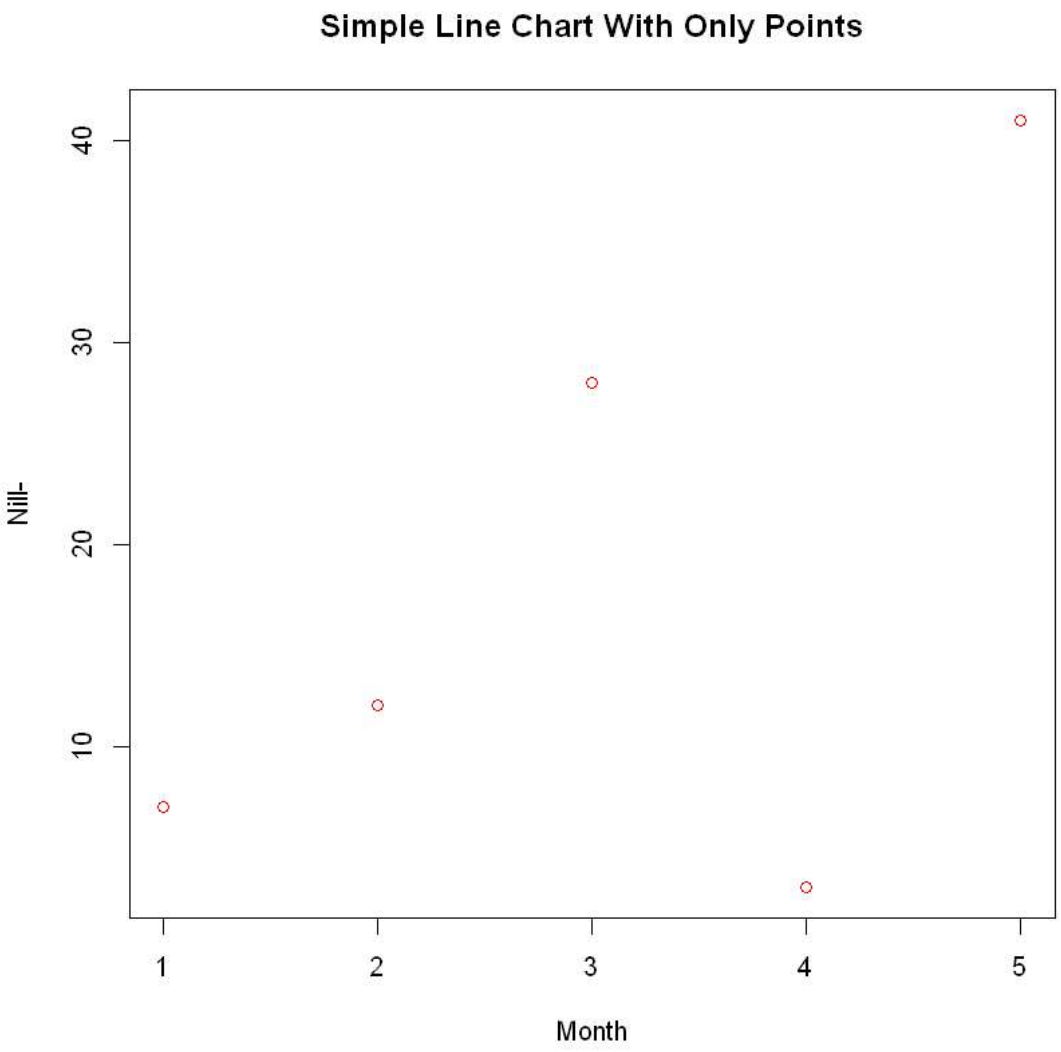
In [1]:

```
v <- c(7,12,28,3,41)
t <- c(12,7,6,11,3)
plot(v,type = "o",col = "red", xlab = "Month", ylab = "Nill-",
   main = "Simple Line Chart With Both Points and Lines")
```



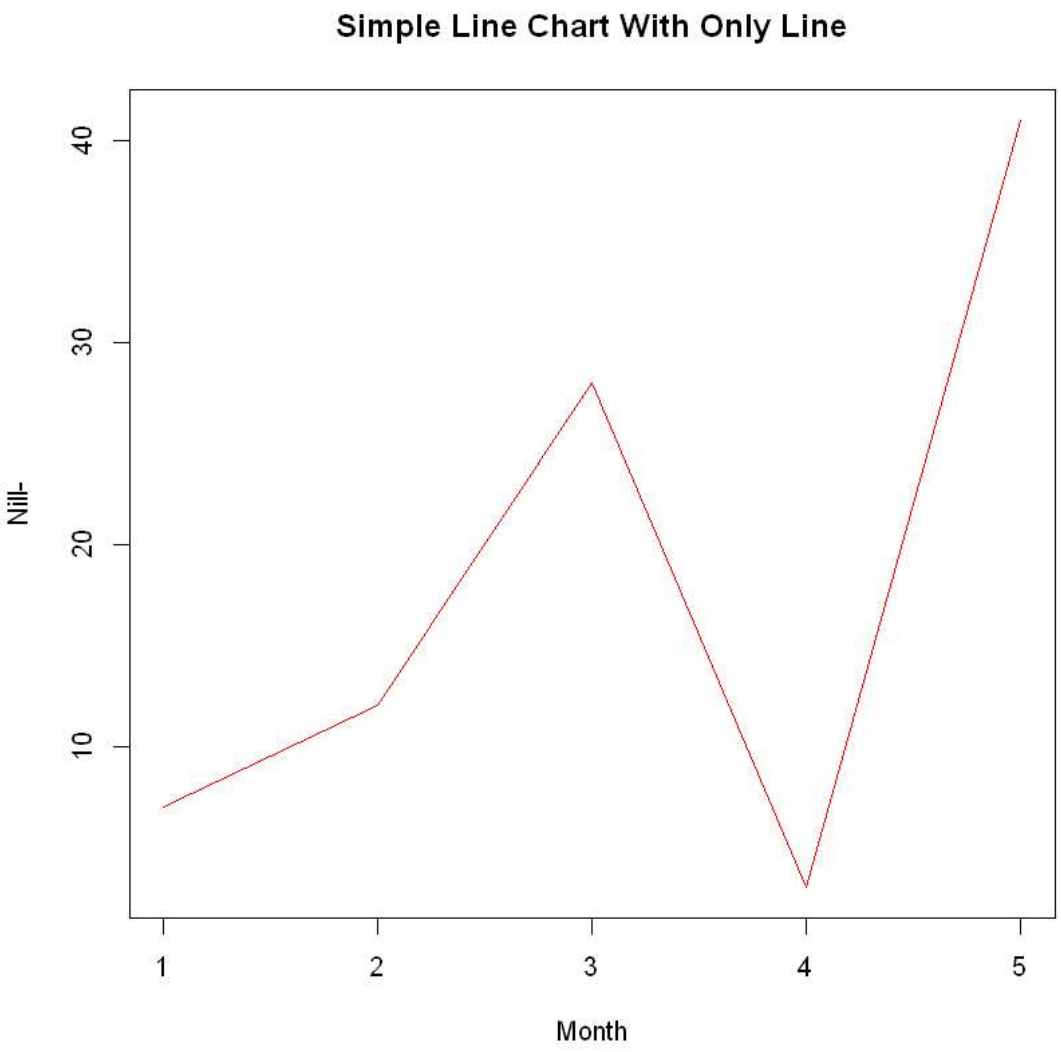
In [11]:

```
v <- c(7,12,28,3,41)
t <- c(12,7,6,11,3)
plot(v,type = "p",col = "red", xlab = "Month", ylab = "Nill-",
   main = "Simple Line Chart With Only Points")
```



In [12]:

```
v <- c(7,12,28,3,41)
t <- c(12,7,6,11,3)
plot(v,type = "l",col = "red", xlab = "Month", ylab = "Nill-",
   main = "Simple Line Chart With Only Line")
```

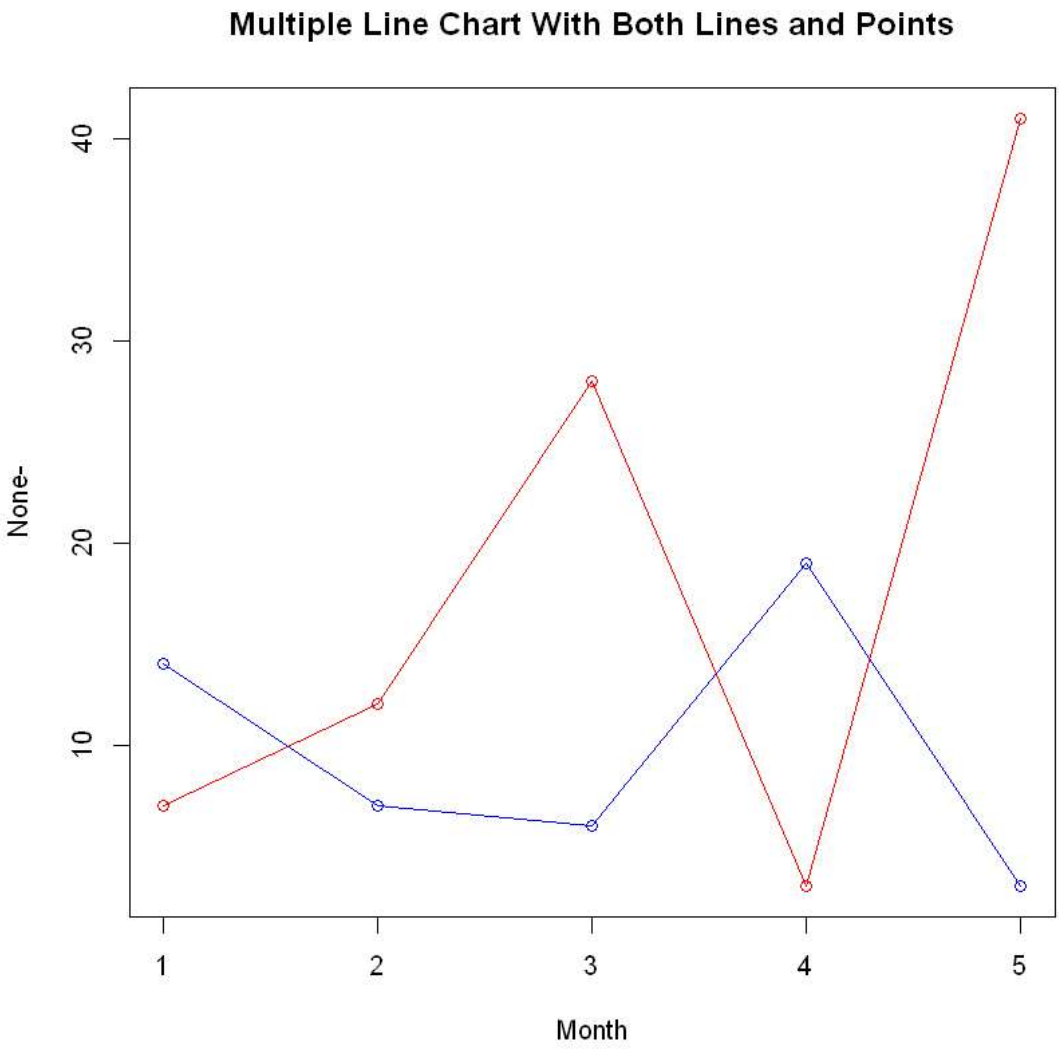


Multiple Line Chart:

* More than one line can be drawn on the same chart by using the lines()function.
* After the first line is plotted, the lines() function can use an additional vector as input to draw the second line in the chart

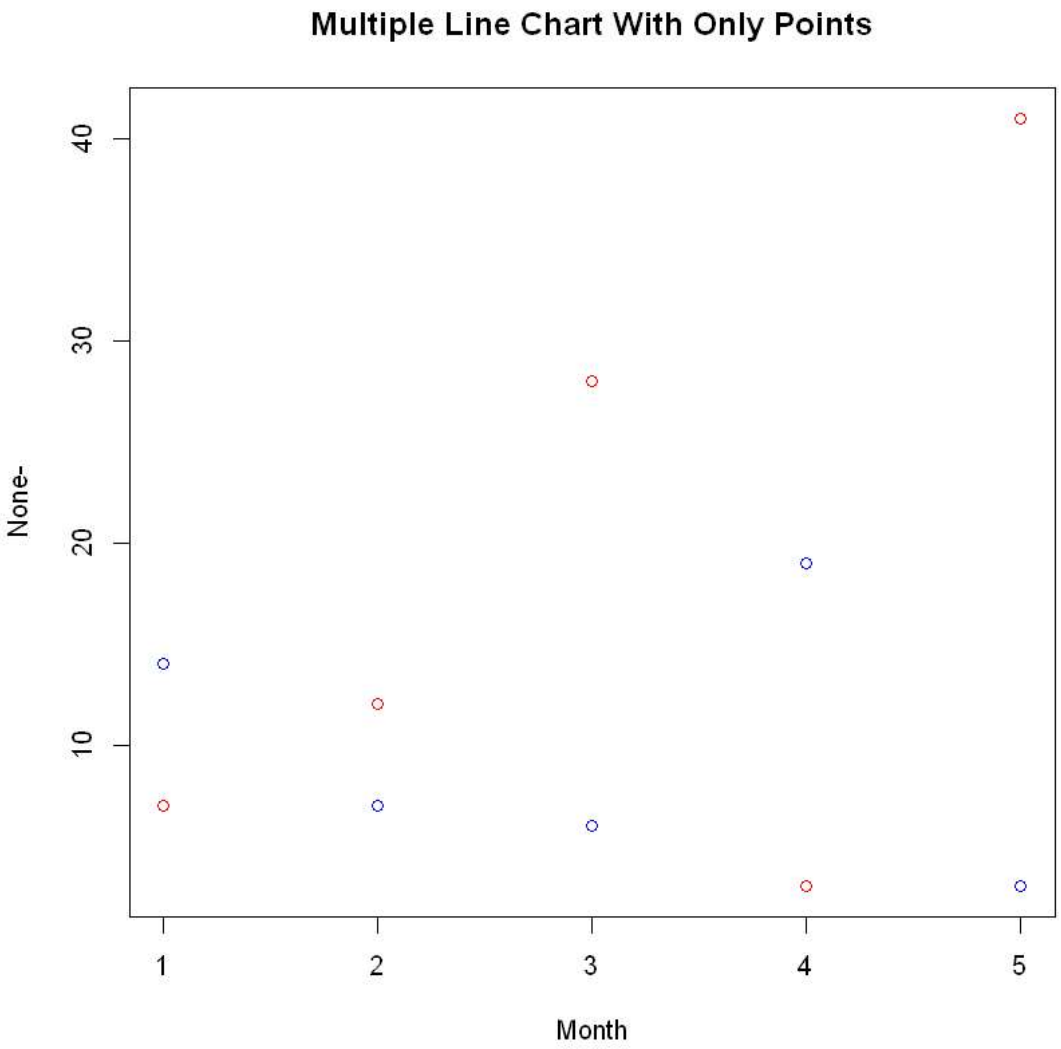
In [16]:

```
v <- c(7,12,28,3,41)
t <- c(14,7,6,19,3)

plot(v,type = "o",col = "red", xlab = "Month", ylab = "None-",
   main = "Multiple Line Chart With Both Lines and Points")

lines(t, type = "o", col = "blue")
```



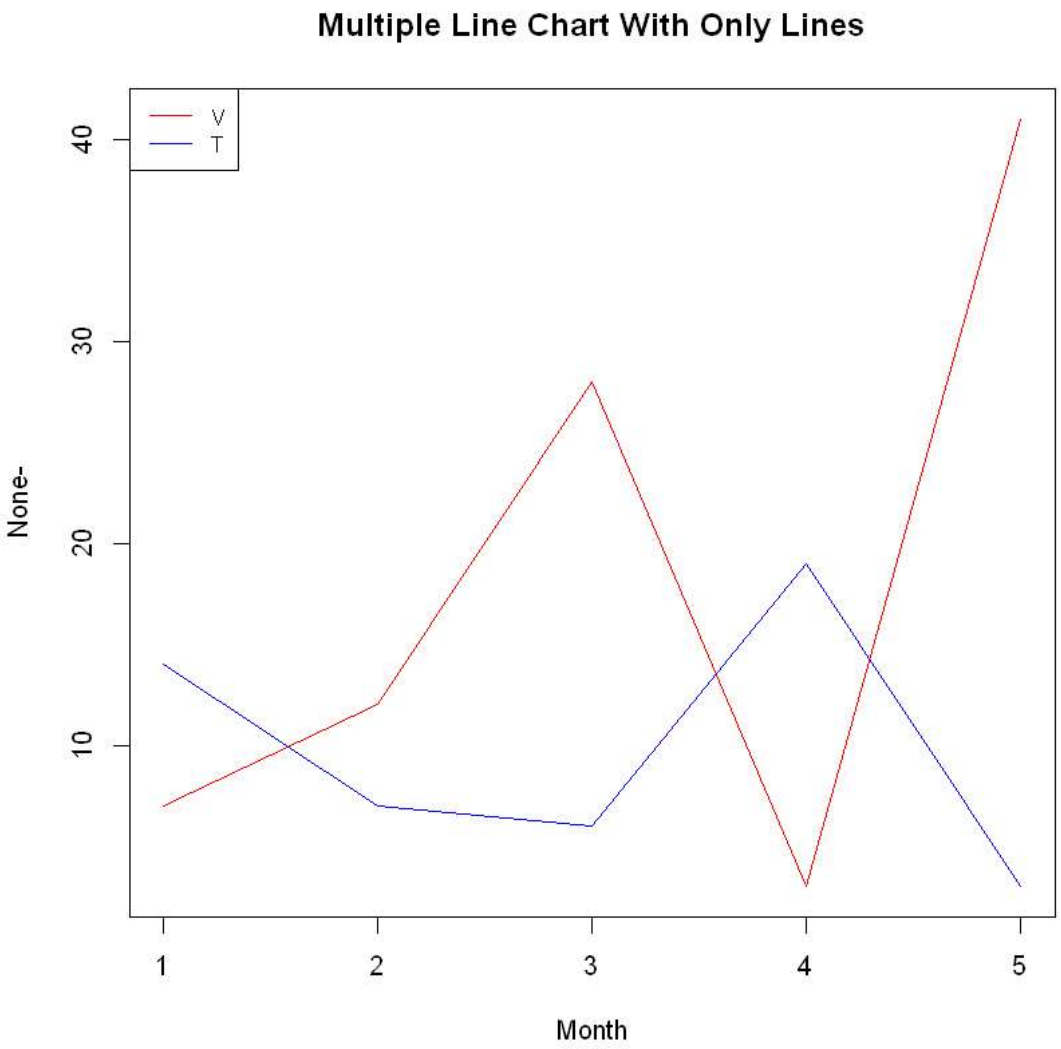
In [18]:

```
v <- c(7,12,28,3,41)
t <- c(14,7,6,19,3)

plot(v,type = "p",col = "red", xlab = "Month", ylab = "None-",
   main = "Multiple Line Chart With Only Points")

lines(t, type = "p", col = "blue")
```



In [3]:

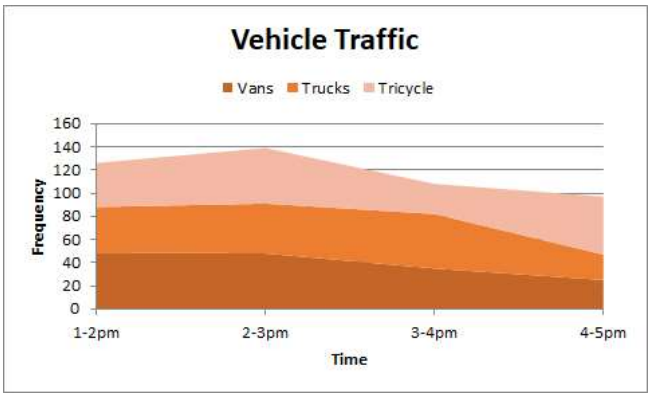
```
v <- c(7,12,28,3,41)
t <- c(14,7,6,19,3)

plot(v,type = "l",col = "red", xlab = "Month", ylab = "None-",
   main = "Multiple Line Chart With Only Lines")

lines(t, type = "l", col = "blue")
legend("topleft", legend=c("V", "T"),
       col=c("red", "blue"), lty=1:1, cex=0.8)
```



Compound Line Chart:

When constructing a compound line chart, you need to first construct multiple line graphs, then shade each part to indicate the component of each data from the total. Each of the bottom lines indicates a part of the total, while the top line is the total.

On a compound line graph, the distance between every 2 consecutive lines shows the size of each part, with the b ottom line being bounded by the origin.


Vehicle Traffic
Vans
Trucks
Tricycle
160
140
120
100
80
60
40
20
0
Frequency
1-2pm
2-3pm
3-4pm
4-5pm
Time