

## Line Charts

### Source

Reprinted with permission from Quick-R (<http://www.statmethods.net>).  
© Copyright 2008 Robert I. Kabacoff, Ph.D. All rights reserved.

### Background

Line charts are created with the function `lines(x, y, type=)` where `x` and `y` are numeric vectors of `(x,y)` points to connect. `type=` can take the following values:

type	description
p	points
l	lines
o	overplotted points and lines
b, c	points (empty if "c") joined by lines
s, S	stair steps
h	histogram-like vertical lines
n	does not produce any points or lines

The `lines()` function adds information to a graph. It can not produce a graph on its own. Usually it follows a `plot(x, y)` command that produces a graph.

By default, `plot()` plots the `(x,y)` points. Use the `type="n"` option in the `plot()` command, to create the graph with axes, titles, etc., but without plotting the points.

### Apply Each type =options to Same Data Set

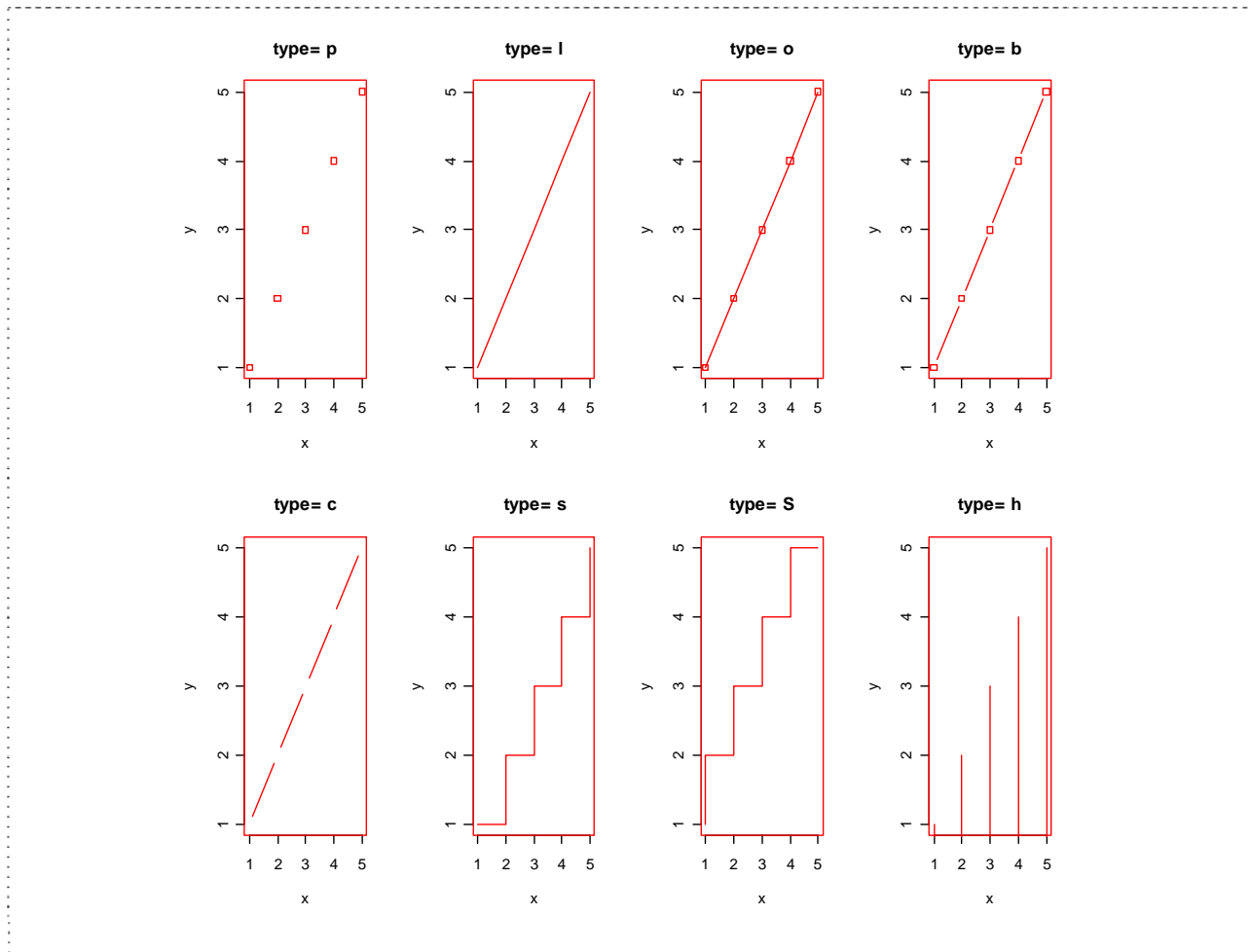
In the following code each of the `type=` options is applied to the same dataset. The `plot()` command sets up the graph, but does not plot the points.

```
# create some data
x <- c(1:5); y <- x

# plotting symbol and color
par(pch=22, col="red")

# all plots on one page
par(mfrow=c(2,4))
opts = c("p","l","o","b","c","s","S","h")

for(i in 1:length(opts))
{
  heading = paste("type=",opts[i])
  plot(x, y, type="n", main=heading)
  lines(x, y, type=opts[i])
}
```



### Demonstrate Each of the type = options

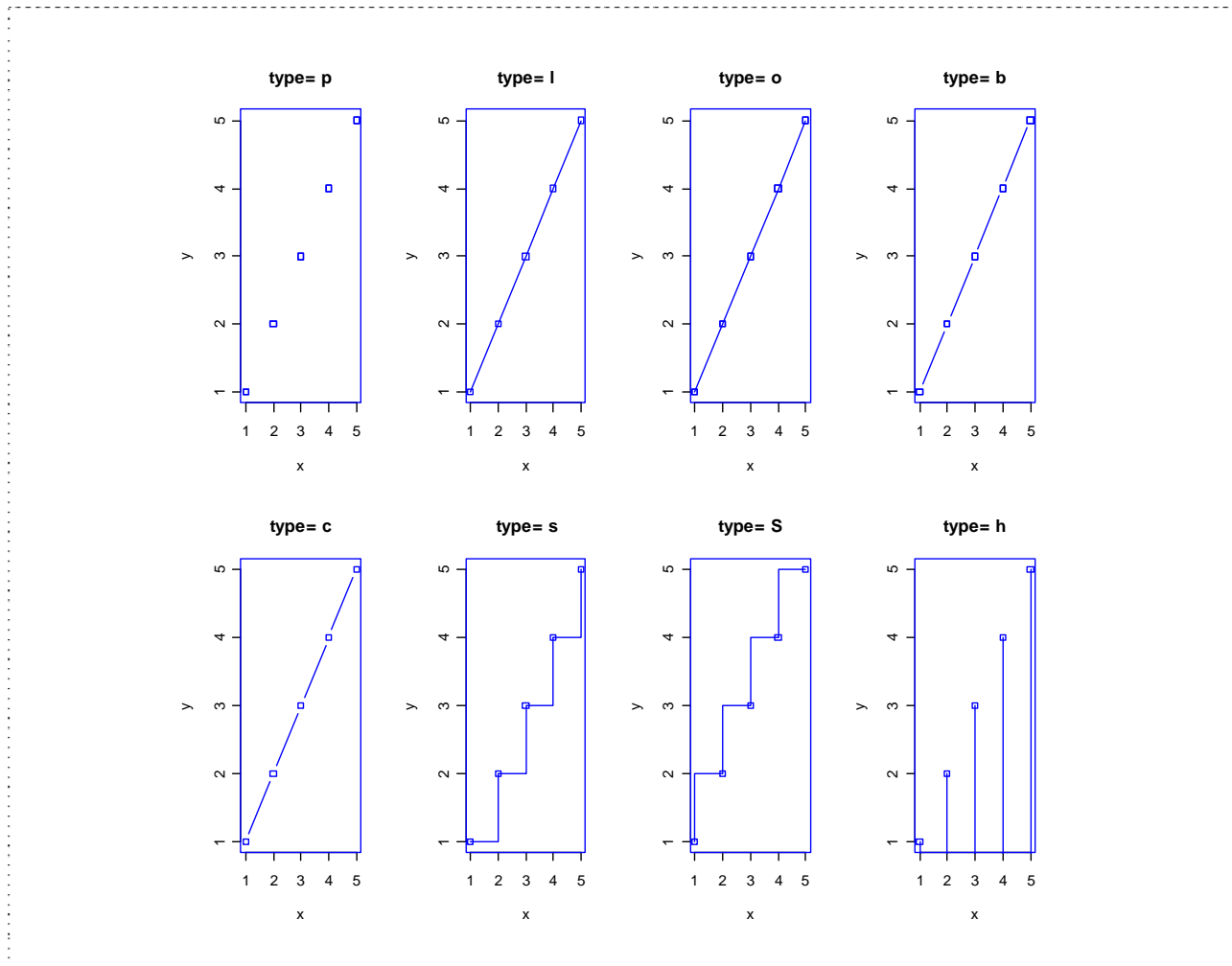
Next, we demonstrate each of the type= options when plot( ) sets up the graph and does plot the points.

```
# create some data
x <- c(1:5); y <- x

# plotting symbol and color
par(pch=22, col="blue")

# all plots on one page
par(mfrow=c(2,4))
opts = c("p","l","o","b","c","s","S","h")

for(i in 1:length(opts))
{
  heading = paste("type=",opts[i])
  plot(x, y, main=heading)
  lines(x, y, type=opts[i])
}
```



As you can see, the type="c" option only looks different from the type="b" option if the plotting of points is suppressed in the plot() command.

### Create Complex Line Chart

To demonstrate the creation of a more complex line chart, let's plot the growth of 5 orange trees over time. Each tree will have its own distinctive line. The data come from the dataset Orange.

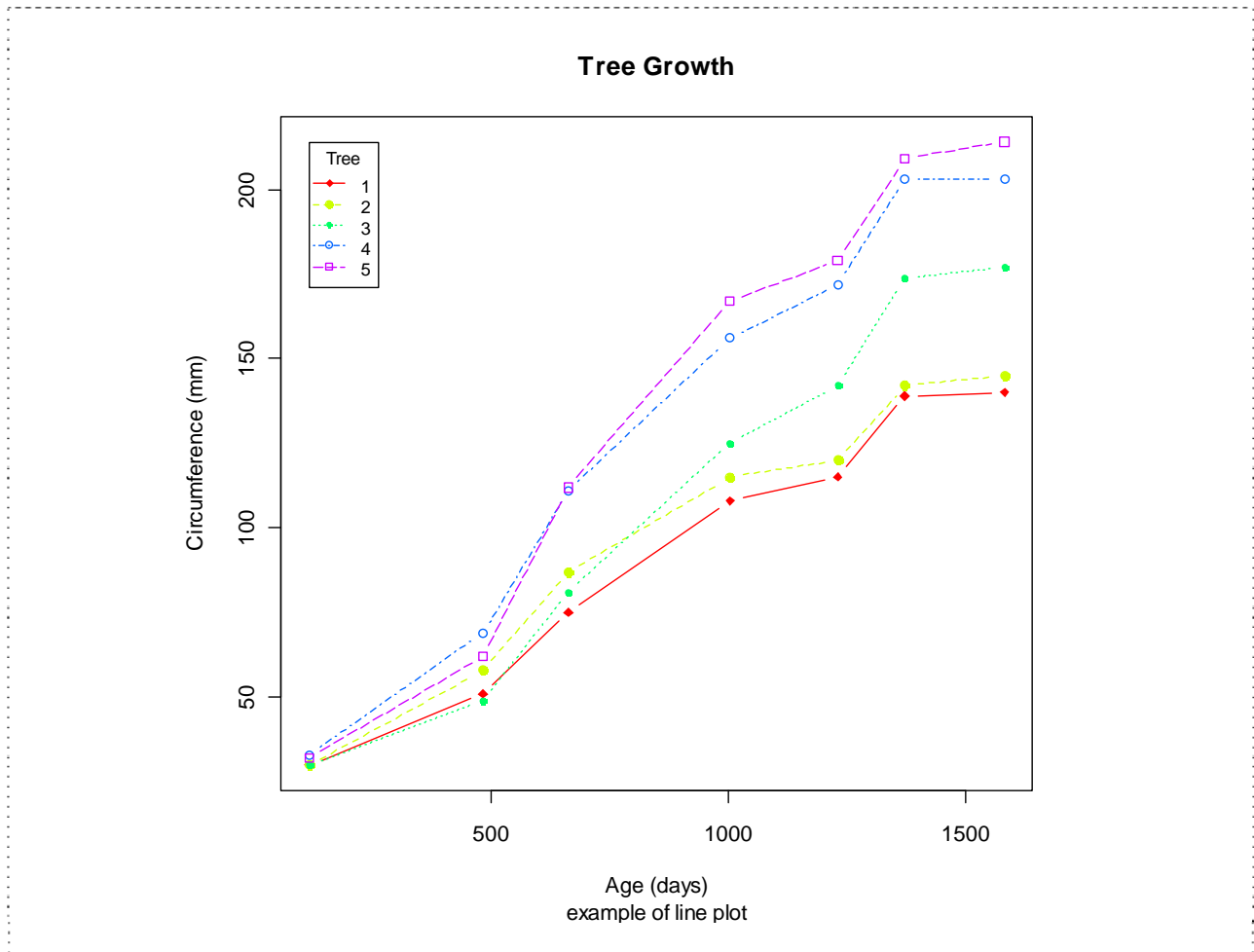
```
# convert factor to numeric for convenience
Orange$Tree <- as.numeric(Orange$Tree)
ntrees <- max(Orange$Tree)

# get the range for the x and y axis
xrange <- range(Orange$age)
yrange <- range(Orange$circumference)

# set up the plot
show <- function()
{
```

```
plot(
  xrange,
  yrange,
  type="n",
  xlab="Age (days)",
  ylab="Circumference (mm)"
)
colors <- rainbow(ntrees)
linetype <- c(1:ntrees)
plotchar <- seq(18,18+ntrees,1)
# add lines
for (i in 1:ntrees)
{
  tree <- subset(Orange, Tree==i)
  lines(
    tree$age,
    tree$circumference,
    type="b",
    lwd=1.5,
    lty=linetype[i],
    col=colors[i],
    pch=plotchar[i]
  )
}
# add a title and subtitle
title("Tree Growth", "example of line plot")
# add a legend
legend(
  xrange[1],
  yrange[2],
  1:ntrees,
  cex=0.8,
  col=colors,
  pch=plotchar,
  lty=linetype,
  title="Tree"
)
}

show()
```



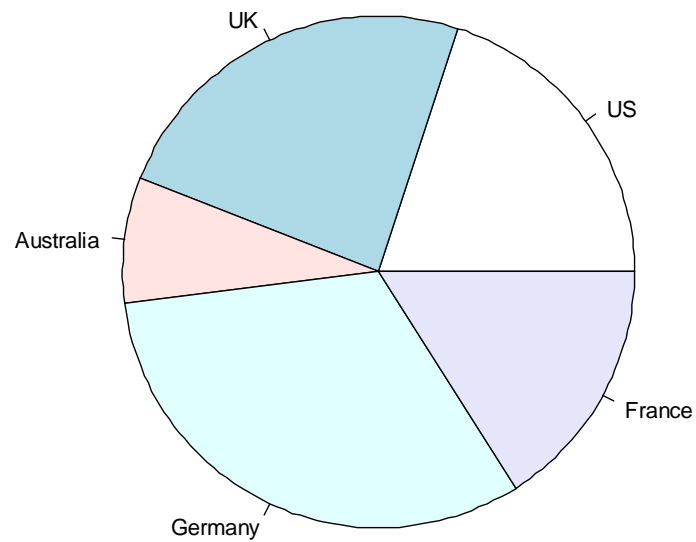
## Pie Charts

Pie charts are not recommended in the R documentation, and their features are somewhat limited. The authors recommend bar or dot plots over pie charts because people are able to judge length more accurately than volume. Pie charts are created with the function `pie(x, labels=)` where `x` is a non-negative numeric vector indicating the area of each slice and `labels=` notes a character vector of names for the slices.

### Simple Pie Chart

```
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pie(
  slices,
  labels = lbls,
  main="Pie Chart of Countries"
)
```

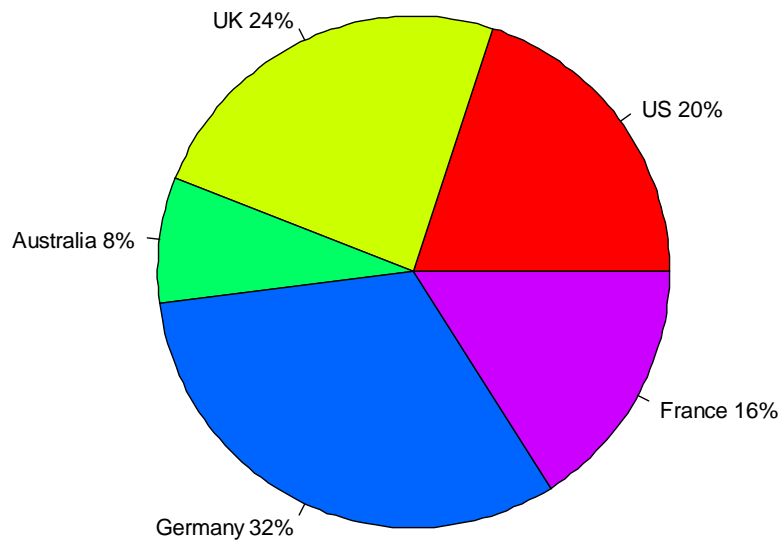
**Pie Chart of Countries**



### Pie Chart with Annotated Percentages

```
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(
  slices,
  labels = lbls,
  col=rainbow(length(lbls)),
  main="Pie Chart of Countries"
)
```

**Pie Chart of Countries**



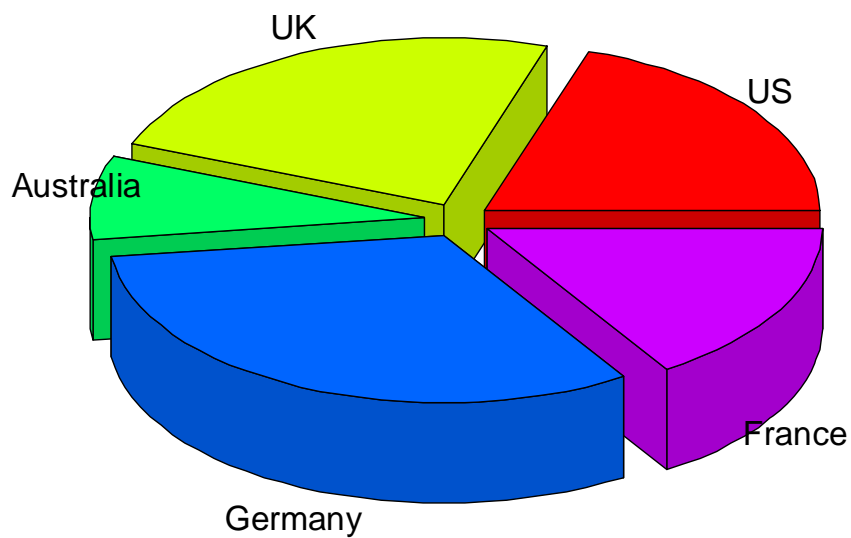
### 3D Exploded Pie Chart

---

The `pie3D()` function in the `plotrix` package provides 3D exploded pie charts.

```
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
pie3D(
  slices,
  labels=lbls,
  explode=0.1,
  main="Pie Chart of Countries "
)
```

Pie Chart of Countries



### Creating Annotated Pies from a Dataframe with Appended Sample Sizes

```
mytable <- table(iris$Species)
lbls <- paste(names(mytable), "\n", mytable, sep="")
pie(
  mytable,
  labels = lbls,
  main="Pie Chart of Species\n (with sample sizes)"
)
```

**Pie Chart of Species  
(with sample sizes)**

