

Plotting Concepts in S-PLUS

There are three basic plotting functions in S-PLUS: high-level plots, low-level plots, and the layout command `par`. Basically, a high-level plot function creates a complete plot and a low-level plot function adds to an existing plot, that is, one created by a high-level plot command.

High-Level Plot Functions

Some of the basic plot functions include:

Function	Description
<code>plot</code>	scatter/line plot
<code>hist</code>	histogram
<code>barplot</code>	barplot
<code>boxplot</code>	boxplot
<code>qqnorm</code>	normal-quantile
<code>qqplot</code>	Q-Q plot
<code>pairs</code>	scatterplot matrix between 2 or more variables

High-level functions may also take optional arguments that enhance the plot.

```
> x <- seq(-3,3,length=25)
> y <- x^2
> plot(x,y)
> plot(x,y,type="l")
> plot(x,y,type="b")
```

Option	Plot type
<code>type="p"</code>	points (default)
<code>type="l"</code>	line
<code>type="b"</code>	both line/pts
<code>type="n"</code>	no points plotted

Other options to `plot` will change the color, type of line, the range of the x- and y-axis, labels for the x- and y-axis, etc.

```
> plot(x,y,type="b",pch="$")
> plot(x,y,type="b",pch=5,col=6)
> plot(x,y,ylim=c(-1,3),ylab="x-squared")
```

Notice the error messages because your y range is now too small for your data.

Option	Description
<code>pch</code>	plot character
<code>lty</code>	line type (solid, dashed,...)
<code>col</code>	color (col=1,2,3...)
<code>xlim</code>	x-axis limits: <code>xlim=c(min,max)</code>
<code>ylim</code>	y-axis limits
<code>xlab</code>	x-axis label
<code>ylab</code>	y-axis label
<code>main</code>	main title
<code>sub</code>	sub title

```
> plot(x,dnorm(x),type="l",
+ main="normal, mu=0,sigma=1")
```

The `barchart`, `hist`, `boxplot` and `qqnorm` will take most of the above options also (the ones that make sense for the type of plot).

Low-level Plot Functions

Low-level plot functions can be executed only after a high-level plot has been created. For example,

```
> plot(x,x^3,type="l")
> lines(x, x^2,lty=2)
> points(c(-1.5, 1.5, 2.5), c(-10, 5,15))
> text(c(-2,0,2),c(5,-5,5),
+ c("fish","frog","fly"))
```

Another example:

```
> hist(jan.temp$Florida)
> abline(v=mean(jan.temp$Florida))
> text(mean(jan.temp$Florida)+5,
+ 100, "mean temp")
```

Function	Description
lines	add a line plot
points	add points
text	add text
mtxt	margin text
abline	add a straight line
qqline	add line to qqnorm
title	add a title

The `abline` command can create vertical or horizontal lines,

```
> plot(x,x^2,type="l")
> abline(h=seq(0,8,by=2),lty=2)
```

or a line of the form $y = 2 + (1/2)x$,

```
> abline(2,1/2, lty=3)
```

For this case, you provide `abline` with the y-intercept and slope.

The par Command

The `par` command controls the layout of the graphsheet. The option you will use most often will probably be `mfrow` (multi-figure, by row), or `mfcol`. For example, to have a 3x2 layout where the plots are added by row, set

```
> par(mfrow=c(3,2))
```

This setting will exist throughout the life of the graphsheet unless you change it back to the default `mfrow=c(1,1)`.

You can also change the default color, plot character, etc. for the graphs created on the graphsheet.

```
> par(col=4,lwd=2,pch=6)
> plot(x,x^2,type="b")
> plot(x,sin(x),type="b")
```

But specifying certain options in the plot function will override these settings.

```
> plot(x,sin(x),type="b",col=3)
```

Incidentally, if you create a new graphsheet,

```
> graphsheet()
```

this new sheet will have the default layout settings.

Misc.

You can save your graphs in either an S-PLUS transport format (Windows extension `.sgr`) which you can then re-open in another S-PLUS session, or you can export to some common file formats (`jpg`, `bmp`, `ps`).

If you want to reopen the graph in a later S-PLUS session, choose from the menu **File > Save**. To re-open, **File > Open**.

To export to another format, choose **File > Export Graph Sheet**. From the command line, you can also type, for example, if the graphsheet name is `GSD4`,

```
> export.graph("A:\\myGraph.sgr",
+ Name="GSD4",exportType="jpeg")
```

The basic syntax is `export.graph(destination, Name="name of graphsheet", exportType="format")`

See the **Help** file for `export.graph` to see the available formats.