

Common Graphing Functions

plot()

```
x=x, y=y, # data, if x is omitted then uses 1:length(y)
xlim=c(0,100), ylim=c(0,100), # limit to axis
main="Plot Title", # title of your plot
xlab="Label for X", ylab="Label for Y", # axes labels
col="steel blue", # color of the points, can be a vector
lty="b", # points="p", lines="l", both="b", or none="n"
pch=19, # type of points to plot
cex.lab=1.5, # aspect ratio for axes labels
cex=2, # aspect ratio for points
)
```

lines()

```
x=x, y=y, # draws a line by connecting points
lty="l", # type of lines, same as above
lwd=0.5, # aspect ratio for line thickness
)
```

text()

```
x=x, y=y, # draws a line by connecting points
labels=some.text, # vector of labels to plot on the graph
pos=3, # position: 1=below, 2=left, 3=above, 4=right
cex=2, # aspect ratio of text size
col="red" # color of text
)
```

points()

```
x=x, y=y, # plots points at the x,y positions
pch=19, # the type of point to plot
cex=2, # aspect ratio of point size
col="red", # color of points
bg="green" # fill color for open symbols
)
```



abline(

a=a, # intercept of the line
b=b # slope of the line
... # additional parameters similar to lines

)

abline(

h=seq(1,10,0.5), # locations of horizontal lines, can be a vector

)

abline(

v=seq(1,10,0.5), # locations of vertical lines, can be a vector

)

abline(

reg=lm.01, # bivariate regression model

)

segments(

x0=x0, y0=y0, # starting points of the segments (usually a vector)
x1=x1, y1=y1, # end points of the segments (usually a vector)
... # additional parameters similar to lines

)

arrows(

x0=x0, y0=y0, # starting points of the arrows (usually a vector)
x1=x1, y1=y1, # end points of the arrows (usually a vector)
code=1, # 1=head at end point, 2=head at start, 3=head at both ends
... # additional parameters similar to lines

)

See also:

identify(x,y)

locator()