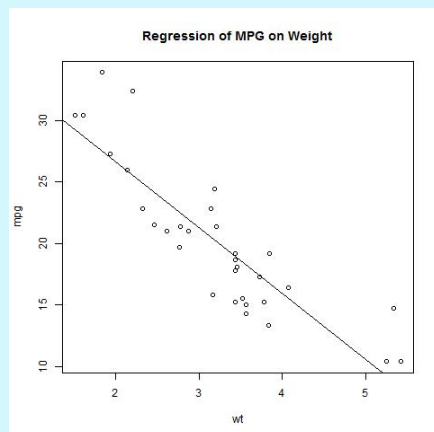


Basic code

```
pdf("mygraph.pdf")
attach(mtcars)
plot(wt, mpg)
abline(lm(mpg ~ wt))
title("Regression of MPG on
Weight")
detach(mtcars)
x <- c(1:10)
y <- x
lines(x, y, type="b", pch=22,
col="blue", lty=2)
dev.off()
```

Variations include `win.metafile()`, `png()`, `jpeg()`, `bmp()`, `tiff()`, `xfig()`, `postscript()`
Other functions include `dev.new()`, `dev.next()`, `dev.prev()`, `dev.set()`

Basic graph



Graphical parameters

```
par(optionname=value,
optionname=value, ...)
opar <- par(no.readonly=TRUE)
par(lty=2, pch=17)
plot(dose, drugA, type="b")
# type - "b": both points and
lines, "l": lines, "p": points,
"c": lines part of "b", "n": no
plotting, "o": both "overplotted"
par(opar)
```

Symbols and lines

pch Specifies the symbol to use when plotting points

cex Specifies the symbol size. A number indicating the amount plotting symbols are scaled relative to default. 1=default, 1.5 is 50% larger, 0.5 is 50% smaller, and so forth.

lty Specifies the line type

lwd Specifies the line width. Expressed relative to default (=1), e.g. `lwd=2` - a line twice as wide as the default

Plot symbols

plot symbols: pch=

□ 0 ◇ 5 ⊕ 10 ■ 15 ● 20 ▽ 25

○ 1 ▽ 6 ✕ 11 ● 16 ○ 21

△ 2 ✕ 7 ⊕ 12 ▲ 17 □ 22

+ 3 * 8 ✕ 13 ◆ 18 ◇ 23

× 4 ◇ 9 ✕ 14 ● 19 △ 24

Line types

line types: lty=

6 - - - - -

5 - - - - -

4

3

2 - - - - -

1 - - - - -

Color parameters

col Default plotting color. Some functions (e.g. lines and pie) accept a vector of values that are recycled.

col.axis Color for axis text

col.lab Color for axis labels

Color parameters (cont)

col.main Color for titles

col.sub Color for subtitles

fg The plot's foreground color

bg The plot's background color

rainbow (#) Produces # contiguous "rainbow" colors

gray(0:10/10) Specify gray levels as a vector of numbers between 0 and 1. This produces 10 gray levels.

Text parameters

cex Number indicating the amount by which plotted text should be scaled relative to the default

cex.axis Magnification of axis text relative to `cex`

cex.lab Magnification of axis labels relative to `cex`

cex.main Magnification of titles relative to `cex`

cex.sub Magnification of subtitles relative to `cex`

Font parameters

font Integer specifying font to use for plotted text. 1=plain, 2=bold, 3=italic, 4=bold italic, 5=symbol (Adobe symbol encoding)

font.ax Font for axis text

font.lab Font for axis labels

font.ma Font for titles

font.sub Font for subtitles

ps Font point size (roughly 1/72 inch). The text size = `ps*cex`.



Font parameters (cont)

family Font family for drawing text.
Standard values are `serif`, `sans`,
and `mono`.

Mapping for font family created via
`windowsFont()` function.

For Mac, use `quartzFonts()`.

```
windowsFonts (
A=windowsFont("Arial Black"),
B=windowsFont("Bookman Old Style"),
C=windowsFont("Comic Sans MS")
)
```

Graph and margin parameters

pin Plot dimensions (width, height) in inches

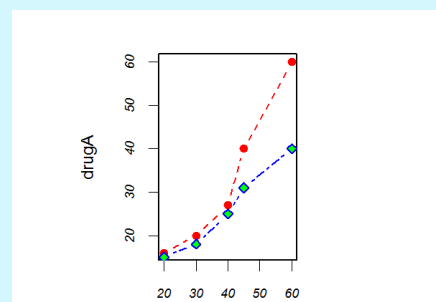
mai Numerical vector indicating margin size,
`c(bottom,left,top,right)`. Expressed in
inches

mar Numerical vector indicating margin size,
`c(bottom,left,top,right)`. Expressed in
lines. The default is `c(5,4,4,2) + 0.1`.

Example code

```
dose <- c(20,30,40,45,60)
drugA <- c(16,20,27,40,60)
drugB <- c(15,18,25,31,40)
opar <- par(no.readonly=TRUE)
par(pin=c(2,3))
par(lwd=2, cex=1.5)
par(cex.axis=.75, font.axis=3)
plot(dose, drugA, type="b", pch=19,
lty=2, col="red")
lines(dose, drugB, type="b",
pch=23, lty=6, col="blue",
bg="green")
par(opar)
```

Resulting graph



Graph text and customization

ann=FALSE Using in `plot()` statement or `par()`
SE statement remove default titles
and labels

title() `title(main="main title",`
`col.main="red",`
`sub="sub-title",`
`col.sub="blue", xlab="x-`
`axis label", ylab="y-axis`
`label", col.lab="green",`
`cex.lab=0.75)`

axis() Create custom axes. When
creating a custom axis, suppress
axis by using option
`axes=FALSE` (suppresses all
axes, including axis frame lines,
unless `frame.plot=TRUE`),
`xaxt="n" or yaxt="n"`. See
Axis options.

text() Add text within graph, typically
labeling points or text annotations,
e.g. `text(location,`
`"text", pos, ...)`

mtext() Add text to margin of plot

Graph text and customization (cont)

e.g. `mtext("text", side=4,`
`line=3, cex.lab=1,`
`las=2, col="blue")`

plotmath Add mathematical symbols and
h() formulas to graph

abline() Add reference lines to graph e.g.
`abline(h=yvalues,`
`v=xvalues)`

`abline(h=c(1,5,7))` adds
horizontal lines at `y=1, 5 & 7`

legend() location - (x,y) coordinate,
location locator(1), keywords, i.e.
`bottom, bottomleft,`
`n, left, topleft, top,`
`title, left, topleft, top,`
`legend, topright, right,`
`...` `bottomright, center.` Use
`inset=` to specify amount to
move legend into graph as
fraction of plot region.

title - character string for the
legend title (optional)

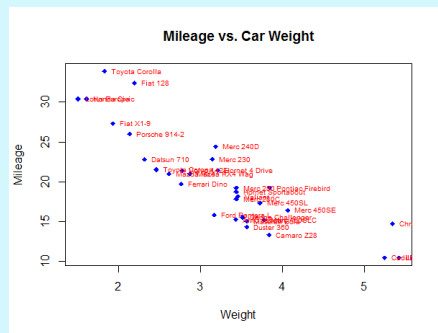
legend - character vector with the
labels

... - Other options: `col=, pch=,`
`lwd=, lty=, fill=, bty=, bg=,`
`cex=, text.col=,`
`horiz=TRUE`

Example code

```
attach(mtcars)
plot(wt, mpg, main="Mileage vs. Car Weight", xlab="Weight",
ylab="Mileage", pch=18,
col="blue")
text(wt, mpg, row.names(mtcars),
cex=0.6, pos=4, col="red")
detach(mtcars)
```

Resulting graph



Combining graphs

Used in `par()` or `layout()` function:

`mflow=c(nrows, ncols)` - fill by row

`mfc=c(nrows, ncols)` - fill by column

`layout(mat)`

where `mat` is matrix object specifying location of multiple plots to combine

e.g. `layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE))`

- one figure placed in row 1 and two figures placed in row 2

Optional parameters for `layout()`:

`widths()` - a vector of values for widths of columns

`heights()` - a vector of values for heights of rows

e.g. `layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE), widths=c(3, 1), heights=c(1, 2))`

Used in `par()` function:

`fig=c(x1, x2, y1, y2), new = TRUE`

Plot within limits of (x1,x2) and (y1,y2), `new = TRUE` option adds figure to existing graph

Axis options

side An integer indicating the side of the graph to draw the axis (1=bottom, 2=left, 3=top, 4=right)

at A numeric vector indicating where tick marks should be drawn

labels A character vector of labels to be placed at the tick marks (if `NULL`, the `at` values will be used)

pos The coordinate at which the axis line is to be drawn (i.e. the value on the other axis where it crosses)

lty Line type

col The line and tick mark color

las Labels are parallel (=0) or perpendicular (=2) to the axis

tck Length of tick mark as a fraction of the plotting region (a '-' number is outside the graph, a '+' number is inside, 0 suppresses ticks, 1 creates gridlines). Default is -0.01.

Adding minor tick marks require `Hmisc` package.

`library(Hmisc)`

`minor.tick(nx=n, ny=n,`

`tick.ratio=n)`

where `nx, ny` are no. of intervals to divide major tick marks on x- and y-axis, `tick.ratio` is size of minor relative to major tick mark

demo(plotmath)

Arithmetic Operators		Radicals	
$x + y$	<code>x+y</code>	$\sqrt{x}$	<code>sqrt(x)</code>
$x - y$	<code>x-y</code>	$\sqrt{x, y}$	<code>sqrt(x, y)</code>
Relations		Relations	
$x * y$	<code>xy</code>	$x == y$	<code>x == y</code>
x / y	<code>x/y</code>	$x \neq y$	<code>x != y</code>
$x \%+\% y$	<code>x % + y</code>	$x \leq y$	<code>x <= y</code>
$x \%/% y$	<code>x % / y</code>	$x < y$	<code>x < y</code>
$x \%*% y$	<code>x % * y</code>	$x \leq y$	<code>x <= y</code>
$x \%.% y$	<code>x % . y</code>	$x > y$	<code>x > y</code>
$-x$	<code>-x</code>	$x \geq y$	<code>x >= y</code>
$+x$	<code>+x</code>	$x \%--\% y$	<code>x @ y</code>
Sub/Superscripts		Sub/Superscripts	
$x[i]$	<code>x[i]</code>	$x \%==\% y$	<code>x == y</code>
x^2	<code>x^2</code>	$x \propto y$	<code>x %prop% y</code>
Juxtaposition		Typeface	
$x * y$	<code>xy</code>	plain(x)	<code>x</code>
$\text{paste}(x, y, z)$	<code>xyz</code>	italic(x)	<code>x</code>
Lists		bold(x)	<code>x</code>
$\text{list}(x, y, z)$	<code>x, y, z</code>	bolditalic(x)	<code>x</code>
		underline(x)	<code>x</code>

