

# Exploratory Data Analysis

## Chapter 4

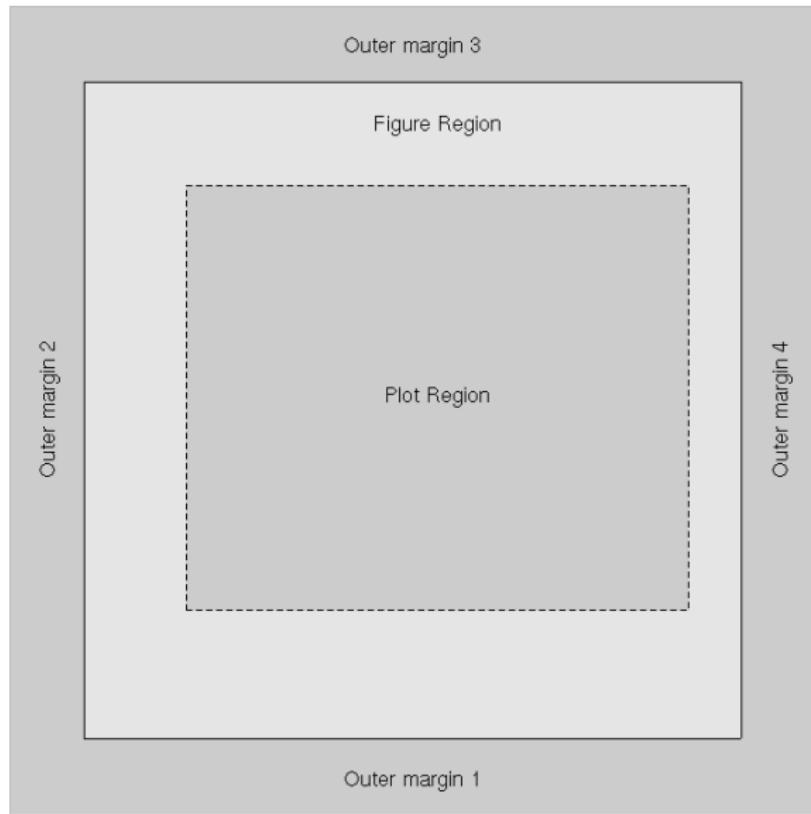
Instructor: Seokho Lee

Hankuk University of Foreign Studies

## 4. R Graphics

### 4.1. Graphic device

- Graphic devices provided by R:
  - windows: console, printer, window metafile for window os
  - postscript: PostScript file
  - pdf: PDF file
  - pictex: LaTeX/PicTeX file
  - png: PNG file
  - jpeg: JPEG file
  - bmp: BMP file
  - xfig: XFIG file



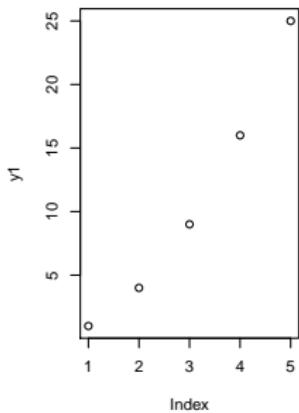
## Example (graphic devices)

```
> dev.new()      # first graphic window. In the window OS, win.graph() can be used
> dev.new()      # second graphic window
> dev.new()      # third graphic window
> dev.list()     # list all graphic devices
quartz quartz quartz
    2      3      4
> dev.cur()      # current graphic device
quartz
    4
> dev.set(which=2)      # change the current graphic device
quartz
    2
> dev.cur()      # current graphic device is changed
quartz
    2
> dev.off()      # close the current graphic device
quartz
    3
> dev.cur()      # current graphic device is changed into 3
quartz
    3
> dev.list()
quartz quartz
    3      4
> graphics.off()      # close all graphic devices
> dev.list()
NULL
```

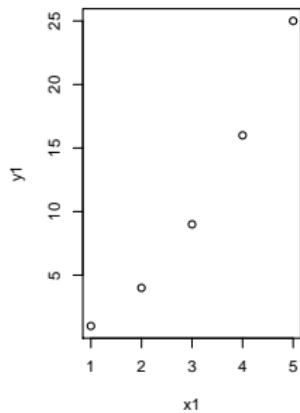
## Example (basic graphic function: plot())

```
> dev.new()      # open graphic window
> x1<-1:5
> y1<-x1^2
> z1<-5:1
> (mat1<-cbind(x1,y1,z1))
      x1  y1  z1
[1,]  1  1  5
[2,]  2  4  4
[3,]  3  9  3
[4,]  4 16  2
[5,]  5 25  1
> par(mfrow=c(2,3))      # divide the window: 2 rows and 3 columns
> plot(y1,main='using index')
> plot(x=x1,y=y1,main='x^2')
> plot(mat1,main='using matrix')
> plot(x1,y1,type='l',main='line')
> plot(x1,y1,type='h',main='high density')
> plot(x1,y1,type='n',main='no plotting')
```

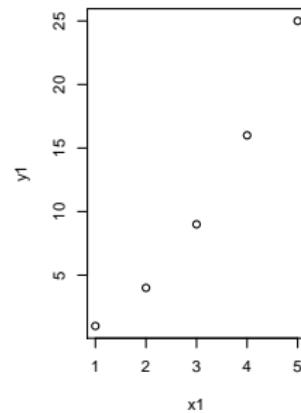
**using index**



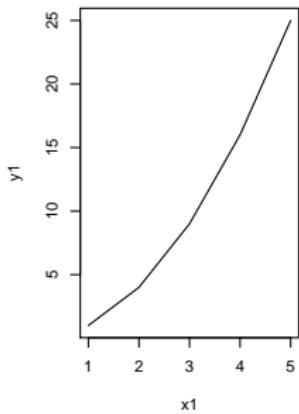
**x^2**



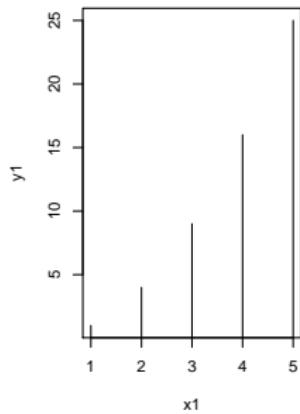
**using matrix**



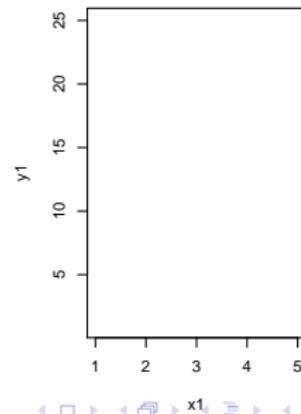
**line**



**high density**



**no plotting**



type argument for `plot()` function:

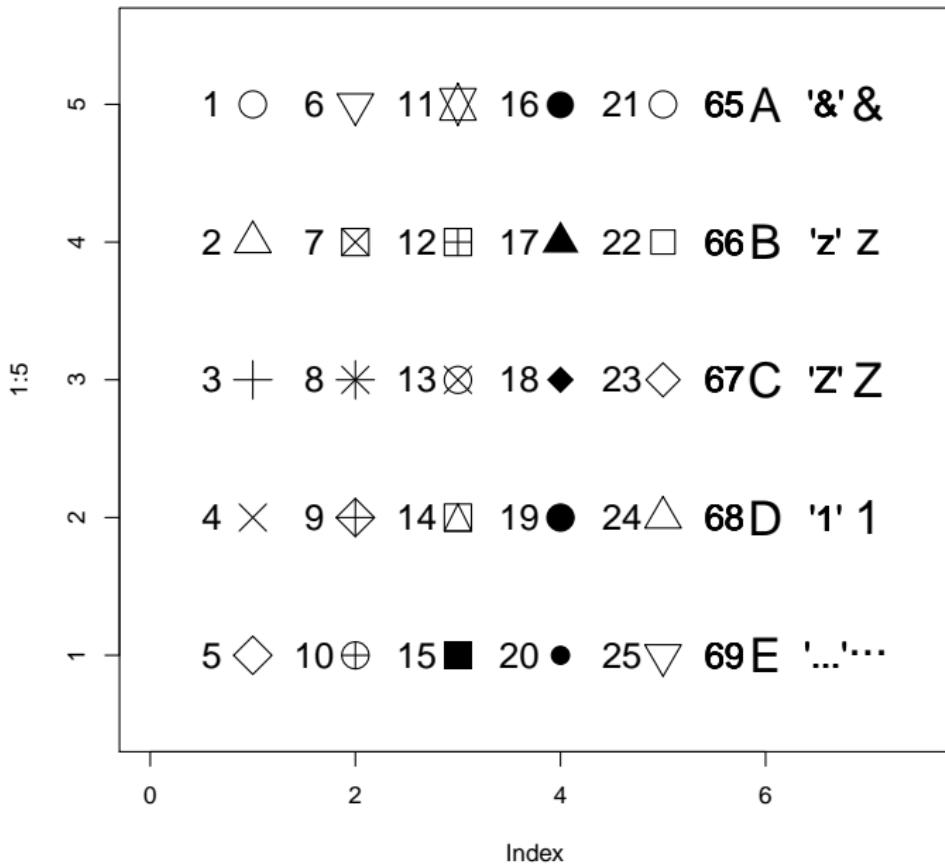
- “p”: for points (default)
- “l”: for lines
- “b”: for both points and lines
- “c”: for the lines part alone of “b”
- “o”: for both ‘overplotted’
- “h”: for ‘histogram’ like (or ‘high-density’) vertical lines
- “s” for stair steps
- “S” for other steps
- “n” for no plotting

Other options for `plot()` will be discussed during the lecture. For a while, we will discuss some graphic functions which will be used to customize.

## Example (points)

```
> x<-rep(1:5,rep(5,5))
> x
[1] 1 1 1 1 1 2 2 2 2 3 3 3 3 3 4 4 4 4 4 4 5 5 5 5 5
> y<-rep(5:1,5)
> y
[1] 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1 5 4 3 2 1
> pchs<-c("&","z","Z","1","†")
> plot(1:5,type='n',xlim=c(0,7.5),ylim=c(0.5,5.5),main="points by 'pch'")
> points(x,y,pch=1:25,cex=2.5)
> text(x-0.4,y,labels=as.character(1:25),cex=1.5)
> points(rep(6,5),5:1,pch=65:69,cex=2)
> text(rep(6,5)-0.4,y,labels=as.character(65:69),cex=1.5)
> points(rep(7,5),5:1,pch=pchs,cex=2)
> text(rep(7,5)-0.4,y,labels=paste("",pchs,"",sep=""),cex=1.5)
```

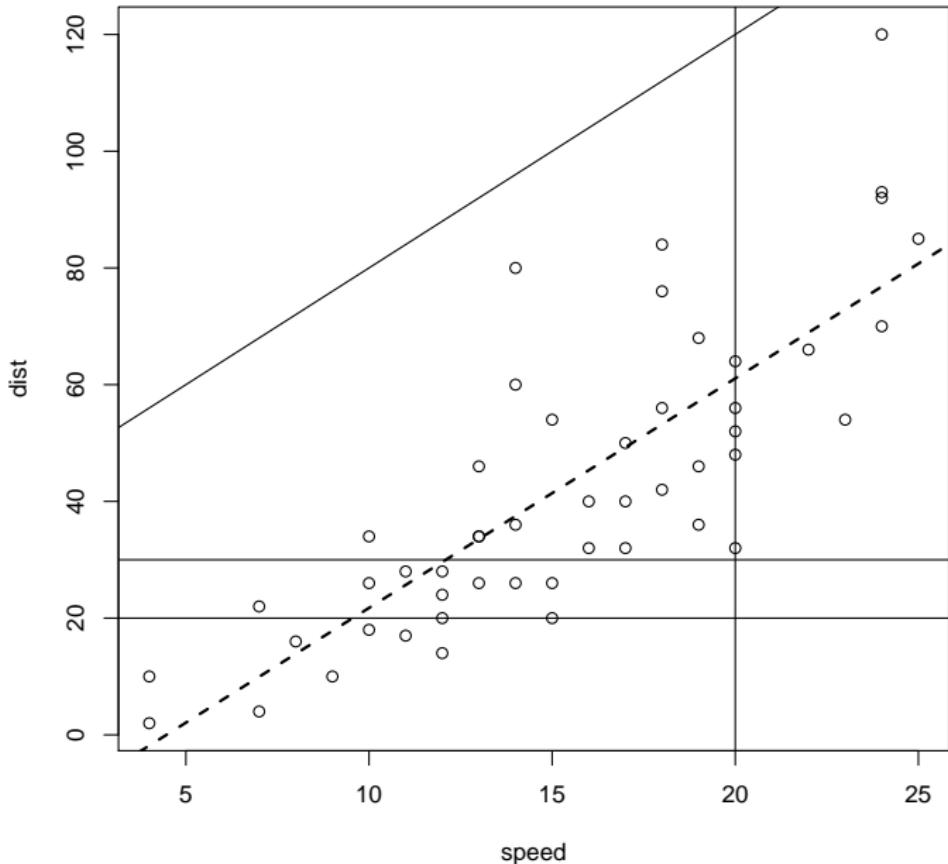
points by 'pch'



## Example (abline)

```
> head(cars)
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
> z<-lm(dist~speed,data=cars)
> z$coef
(Intercept)      speed
-17.579095     3.932409
> plot(cars,main='abline')
> abline(h=20)
> abline(h=30)
> abline(v=20)
> abline(a=40,b=4)
> abline(z,lty=2,lwd=2)
```

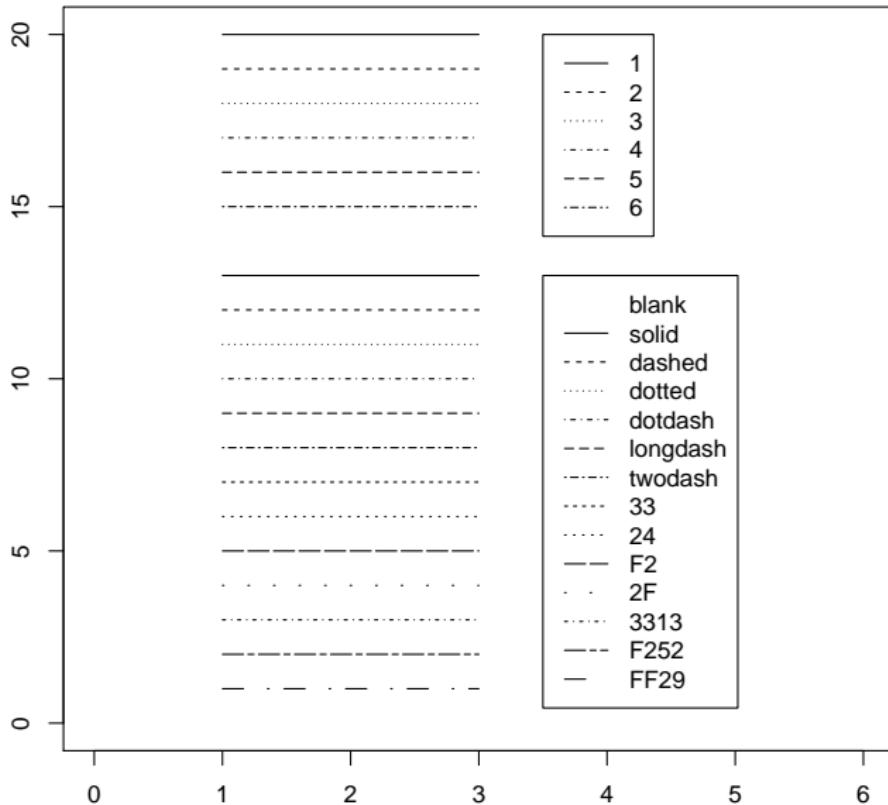
### abline



## Example (lines)

```
> lty1<-c('blank','solid','dashed','dotted','dotdash','longdash','twodash')
> lty2<-c('33','24','F2','2F','3313','F252','FF29')
> plot(0:6,0:6,type='n',ylim=c(0,20),xlab='',ylab='',main='lines')
> lines(c(1,3),c(20,20),lty=1)
> lines(c(1,3),c(19,19),lty=2)
> lines(c(1,3),c(18,18),lty=3)
> lines(c(1,3),c(17,17),lty=4)
> lines(c(1,3),c(16,16),lty=5)
> lines(c(1,3),c(15,15),lty=6)
> lines(c(1,3),c(14,14),lty=lty1[1])
> lines(c(1,3),c(13,13),lty=lty1[2])
> lines(c(1,3),c(12,12),lty=lty1[3])
> lines(c(1,3),c(11,11),lty=lty1[4])
> lines(c(1,3),c(10,10),lty=lty1[5])
> lines(c(1,3),c(9,9),lty=lty1[6])
> lines(c(1,3),c(8,8),lty=lty1[7])
> lines(c(1,3),c(7,7),lty=lty2[1])
> lines(c(1,3),c(6,6),lty=lty2[2])
> lines(c(1,3),c(5,5),lty=lty2[3])
> lines(c(1,3),c(4,4),lty=lty2[4])
> lines(c(1,3),c(3,3),lty=lty2[5])
> lines(c(1,3),c(2,2),lty=lty2[6])
> lines(c(1,3),c(1,1),lty=lty2[7])
> legend(3.5,20,legend=1:6,lty=1:6)
> legend(3.5,13,legend=c(lty1,lty2),lty=c(lty1,lty2))
```

# lines



## Example (arrows)

```
> plot(1:9,type='n',axes=F,xlab='',ylab='',main='arrows')
> arrows(1,9,4,9,angle=30,length=0.25,code=2)
> arrows(1,8,4,8,length=0.5)
> arrows(1,7,4,7,length=0.1)
> arrows(1,6,4,6,angle=60)
> arrows(1,5,4,5,angle=90)
> arrows(1,4,4,4,angle=120)
> arrows(1,3,4,3,code=0)
> arrows(1,2,4,2,code=1)
> arrows(1,1,4,1,code=3)
> text(4.5,9,adj=0,'angle=30, length=0.25,code=2 (default)')
> text(4.5,8,adj=0,'length=0.5')
> text(4.5,7,adj=0,'length=0.1')
> text(4.5,6,adj=0,'angle=60')
> text(4.5,5,adj=0,'angle=90')
> text(4.5,4,adj=0,'angle=120')
> text(4.5,3,adj=0,'code=0')
> text(4.5,2,adj=0,'code=1')
> text(4.5,1,adj=0,'code=3')
```

## arrows



angle=30, length=0.25,code=2 (default)



length=0.5



length=0.1



angle=60



angle=90



angle=120



code=0



code=1

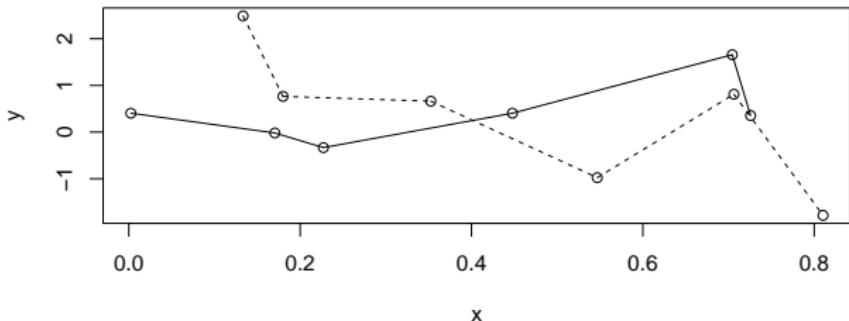


code=3

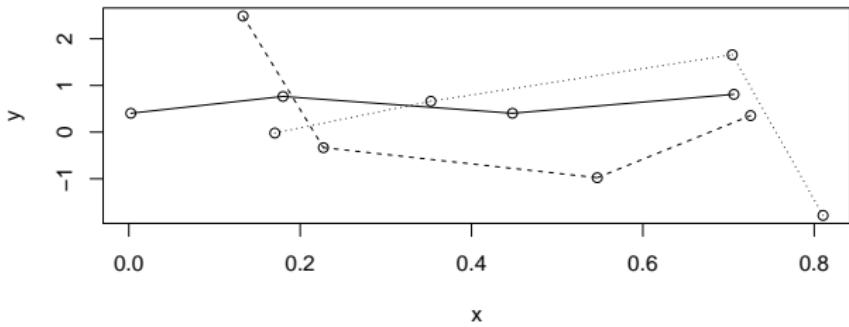
## Example (`segments`)

```
> x<-runif(12); y<-rnorm(12)
> i<-order(x); x<-x[i]; y<-y[i]
> par(mfrow=c(2,1))
> plot(x,y,main="2 segments by segments function")
> s<-seq(length(x)-1)
> segments(x[s],y[s],x[s+2],y[s+2],lty=1:2)
> plot(x,y,main="3 segments by segments function")
> s<-seq(length(x)-2)
> segments(x[s],y[s],x[s+3],y[s+3],lty=1:3)
> box(which='outer')
```

### 2 segments by segments function



### 3 segments by segments function

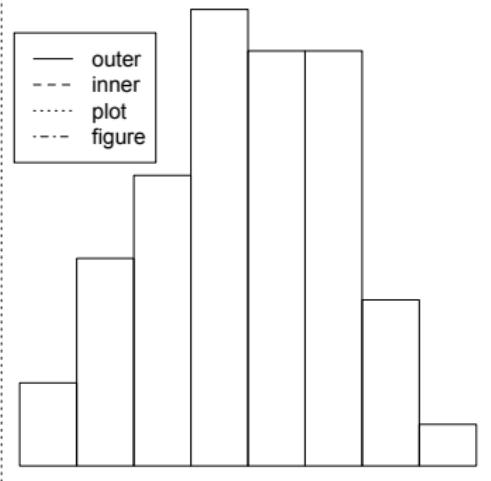


## Example (box)

```
par(mar=c(4,4,4,4))
par(oma=c(4,4,4,4))
hist(rnorm(50),axes=F,xlab="",ylab="",main="box")
whichs<-c("outer","inner","plot","figure")
box(which=whichs[1],lty=1)
box(which=whichs[2],lty=2)
box(which=whichs[3],lty=3)
box(which=whichs[4],lty=4)
legend(locator(1),legend=whichs,lty=1:4)
```

# box

- outer
- - inner
- ... plot
- - - figure



## Example (rect)

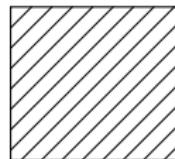
```
> plot(1:10,type="n",main="rect",xlab="",ylab="",axes=F)
> rect(xleft=1,ybottom=7,xright=3,ytop=9)
> rect(1,4,3,6,col="gold")
> rect(1,1,3,3,border="gold")
> rect(4,7,6,9,density=10)
> rect(4,4,6,6,density=10,angle=315)
> rect(4,1,6,3,density=25)
> rect(7,7,9,9,lwd=2)
> rect(7,4,9,6,lty=2)
> rect(7,1,9,3,lty=2,density=10)
> text(2,9.5,adj=0.5,"default")
> text(2,6.5,adj=0.5,"col=\"gold\"")
> text(2,3.5,adj=0.5,"border=\"gold\"")
> text(5,9.5,adj=0.5,"density=10")
> text(5,6.5,adj=0.5,"density=10, angle=315")
> text(5,3.5,adj=0.5,"density=25")
> text(8,9.5,adj=0.5,"lwd=2")
> text(8,6.5,adj=0.5,"lty=2")
> text(8,3.5,adj=0.5,"lty=2, density=10")
```

## rect

default



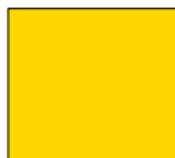
density=10



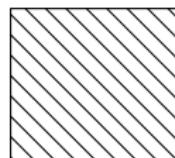
lwd=2



col="gold"



density=10, angle=315



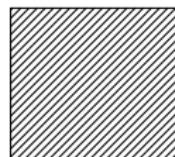
lty=2



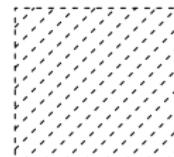
border="gold"



density=25



lty=2, density=10

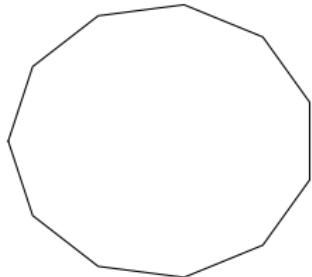


## Example (polygon)

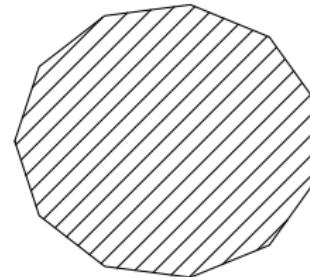
```
> theta<-seq(-pi,pi,length=12)
> x<-cos(theta)
> y<-sin(theta)
> plot(1:6,type="n",main="polygon",xlab="",ylab="",axes=F)
> x1<-x+2; y1<-y+4.5
> polygon(x1,y1)
> x2<-x+2; y2<-y+2
> polygon(x2,y2,col="gold")
> x3<-x+5; y3<-y+4.5
> polygon(x3,y3,density=10)
> x4<-x+5; y4<-y+2
> polygon(x4,y4,lty=2,lwd=2)
> text(2,5.7,adj=0.5,"default")
> text(2,3.2,adj=0.5,"col=\"gold\"")
> text(5,5.7,adj=0.5,"density=10")
> text(5,3.2,adj=0.5,"lty=2, lwd=2")
```

## **polygon**

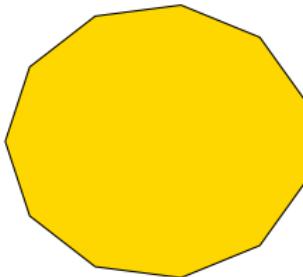
default



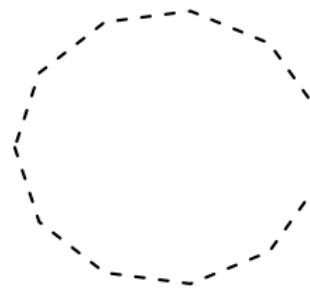
density=10



col="gold"



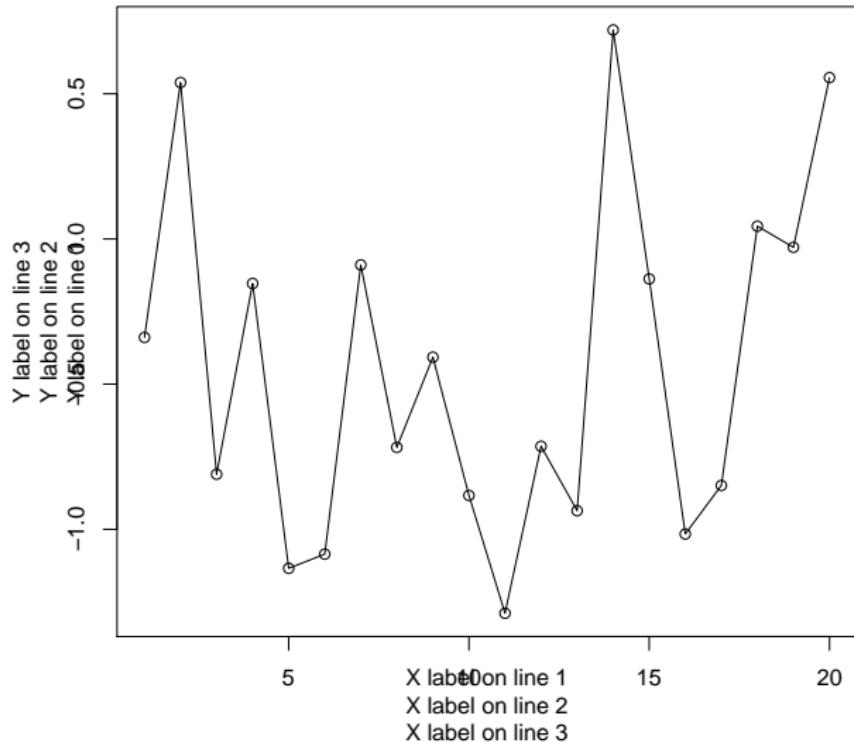
lty=2, lwd=2



## Example (title)

```
> par(mar=c(4,4,4,4),oma=c(4,0,0,0))
> plot(rnorm(20),type="o",xlab="",ylab="")
> title(main="Main title on line 1", line=1)
> title(main="Main title on line 2", line=2)
> title(main="Main title on line 3", line=3)
> title(sub="subtitle on line 1",line=1,outer=T)
> title(sub="subtitle on line 2",line=2,outer=T)
> title(sub="subtitle on line 3",line=3,outer=T)
> title(xlab="X label on line 1", line=1)
> title(xlab="X label on line 2", line=2)
> title(xlab="X label on line 3", line=3)
> title(ylab="Y label on line 1", line=1)
> title(ylab="Y label on line 2", line=2)
> title(ylab="Y label on line 3", line=3)
```

Main title on line 3  
Main title on line 2  
Main title on line 1



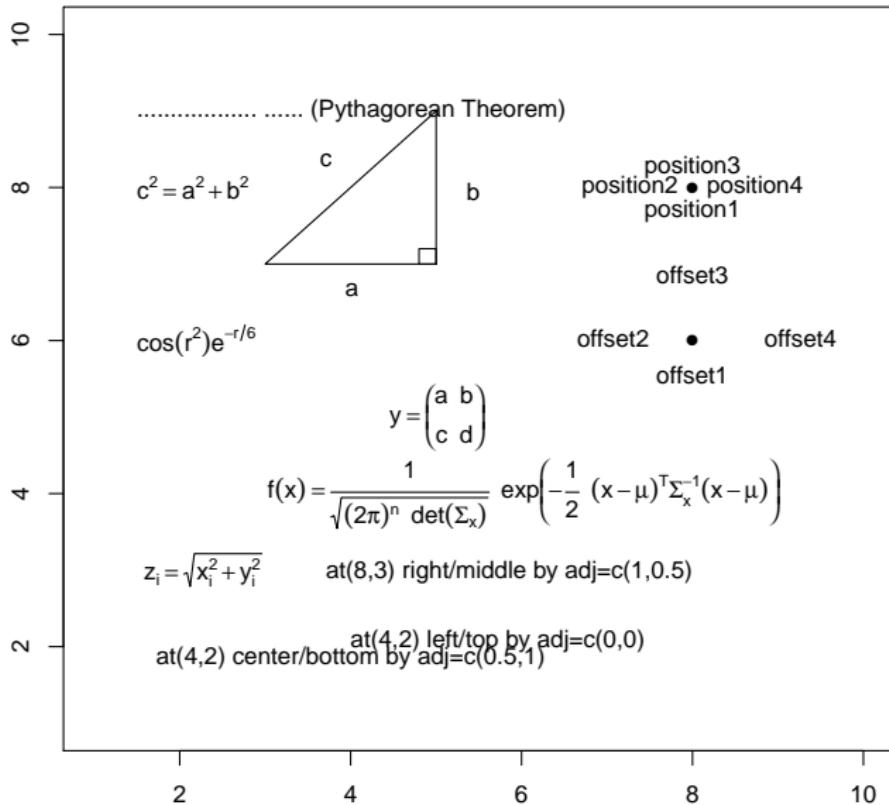
subtitle on line 1  
subtitle on line 2  
subtitle on line 3

## Example (text)

```
> plot(1:10,1:10,type="n",xlab="",ylab="",main="text")
> text(1.5,9,adj=0,labels="勾股定理 (Pythagorean Theorem)")
> polygon(c(5,3,5),c(9,7,7)); polygon(c(5,5,4.8,4.8),c(7,7.2,7.2,7))
> text(3.64,8.36,adj=0,labels="c")
> text(3.94,6.67,adj=0,labels="a")
> text(5.36,7.95,adj=0,labels="b")
> # Example expression labels
> text(1.5,8,adj=0,labels=expression(c^2==a^2+b^2))
> text(1.5,6,adj=0,labels=expression(cos(r^2)*e^{(-r/6)}))
> text(2,3,adj=0.3,labels=expression(z[i]==sqrt(x[i]^2+y[i]^2)))
> text(9,4,adj=1,labels=expression(
+   f(x)==frac(1,sqrt((2*pi)^n ~~ det(Sigma[x])))) ~~ exp * bgroup("(",
+   -frac(1,2) ~~(x-mu)^T*Sigma[x]^-1 *(x-mu), ")"))
> text(5,5,adj=0.5,
+   labels=expression(y==bgroup("(,atop(a~~b,c~~d),")")))
> # Example position by pos
> points(8,8,pch=16)
> text(8,8,"position1", pos=1)
> text(8,6,"offset3",pos=3,offset=2)
> text(8,6,"offset4",pos=4,offset=2.5)
> # Example adj by adj(x,y)
> text(4,2,"at(4,2) left/top by adj=c(0,0)", adj=c(0,0))
> text(4,2,"at(4,2) center/bottom by adj=c(0.5,1)", adj=c(0.5,1))
> text(8,3,"at(8,3) right/middle by adj=c(1,0.5)", adj=c(1,0.5))
> text(8,8,"position2", pos=2)
> text(8,8,"position3", pos=3)
> text(8,8,"position4", pos=4)
> # Example offset
> points(8,6,pch=16)
> text(8,6,"offset1",pos=1,offset=1)
> text(8,6,"offset2",pos=2,offset=1.5)
```

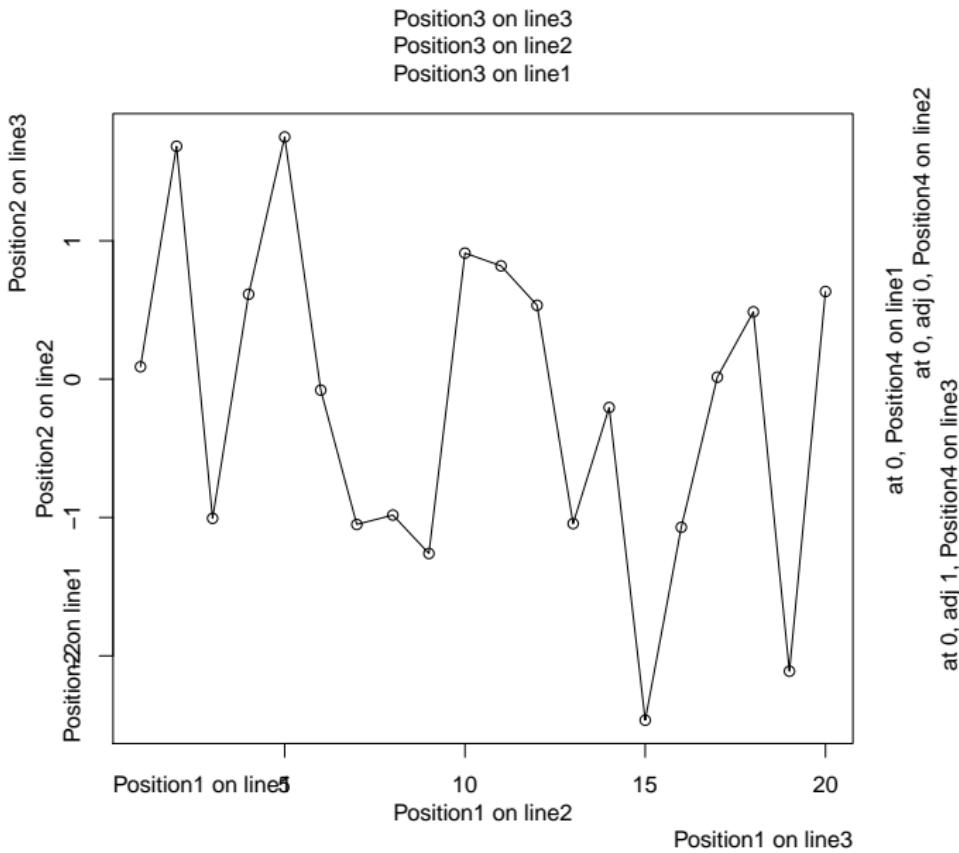


## text



## Example (mtext)

```
> par(mar=c(4,4,4,4),oma=c(4,0,0,0))
> plot(rnorm(20),type="o",xlab="",ylab="")
> mtext("Position3 on line1", line=1)
> mtext("Position3 on line2", side=3, line=2)
> mtext("Position3 on line3", side=3, line=3)
> mtext("Outer position1 on line1", side=1, line=1, outer=T)
> mtext("Outer position1 on line2", side=1, line=2, outer=T)
> mtext("Outer position1 on line3", side=1, line=3, outer=T)
> mtext("Position1 on line1", side=1, line=1, adj=0)
> mtext("Position1 on line2", side=1, line=2, adj=0.5)
> mtext("Position1 on line3", side=1, line=3, adj=1)
> mtext("Position2 on line1", side=2, line=1, adj=0)
> mtext("Position2 on line2", side=2, line=2, adj=0.5)
> mtext("Position2 on line3", side=2, line=3, adj=1)
> mtext("at 0, Position4 on line1", side=4, line=1, at=0)
> mtext("at 0, adj 0, Position4 on line2", side=4, line=2, at=0, adj=0)
> mtext("at 0, adj 1, Position4 on line3", side=4, line=3, at=0, adj=1)
```

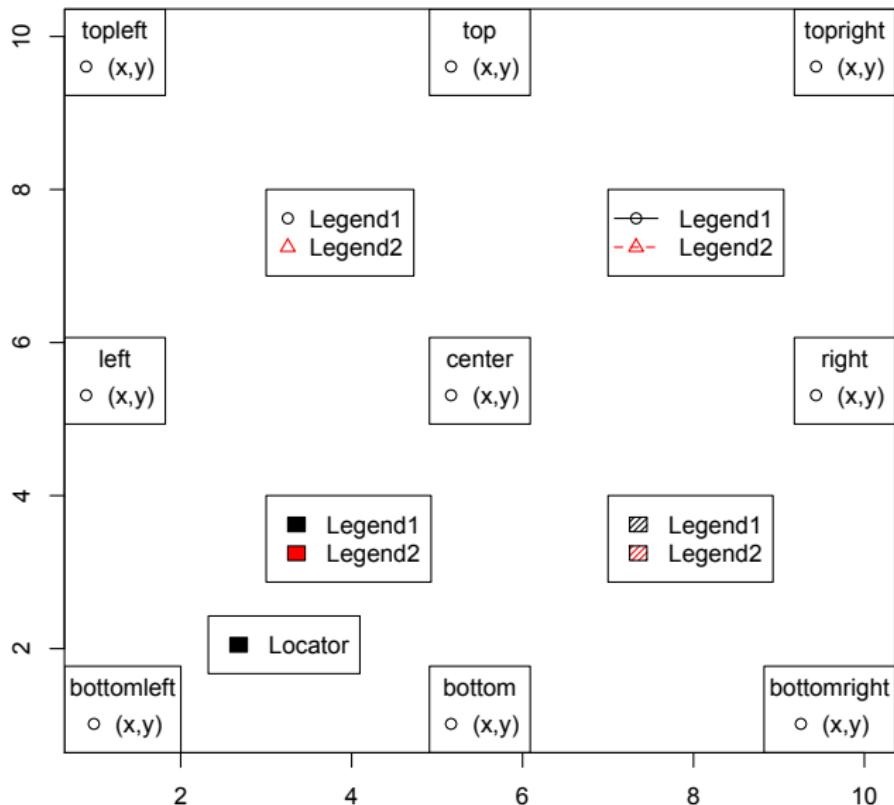


Outer position1 on line1  
Outer position1 on line2  
Outer position1 on line3

## Example (legend)

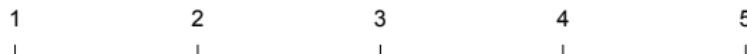
```
> plot(1:10,type="n",xlab="",ylab="",main="legend")
> legend("bottomright","(x,y)", pch=1,title="bottomright")
> legend("bottom","(x,y)",pch=1,title="bottom")
> legend("bottomleft","(x,y)",pch=1,title="bottomleft")
> legend("left","(x,y)",pch=1,title="left")
> legend("topleft","(x,y)",pch=1,title="topleft")
> legend("top","(x,y)",pch=1,title="top")
> legend("topright","(x,y)",pch=1,title="topright")
> legend("right","(x,y)",pch=1,title="right")
> legend("center","(x,y)",pch=1,title="center")
> legends<-c("Legend1","Legend2")
> legend(3,8,legend=legends,pch=1:2,col=1:2)
> legend(7,8,legend=legends,pch=1:2,col=1:2,lty=1:2)
> legend(3,4,legend=legends,fill=1:2)
> legend(7,4,legend=legends,fill=1:2,density=30)
> legend(locator(1),legend="Locator",fill=1)
```

## legend

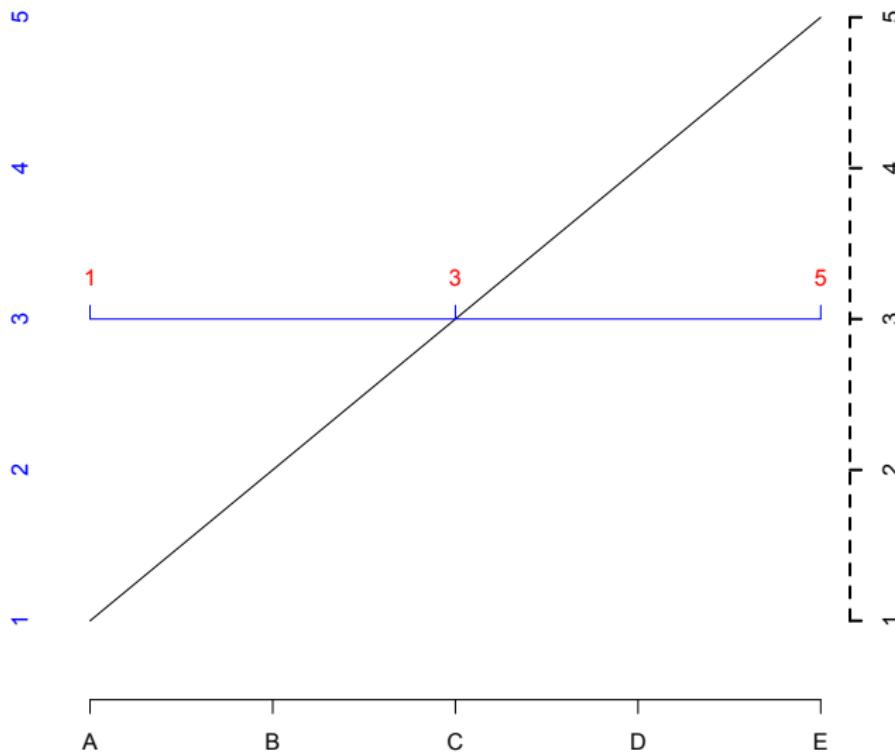


## Example (axis)

```
> par(oma=c(0,0,2,0))
> plot(1:5,type="l",main="axis",axes=FALSE,xlab="",ylab="")
> axis(side=1,at=1:5,labels=LETTERS[1:5],line=2)
> axis(side=2,tick=F, col.axis="blue")
> axis(side=3,outer=T)
> axis(side=3,at=c(1,3,5),pos=3,col="blue",col.axis="red")
> axis(side=4,lty=2,lwd=2)
```



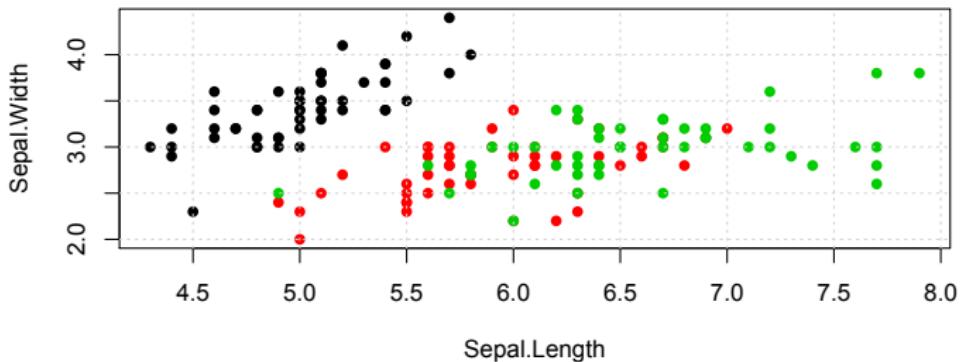
axis



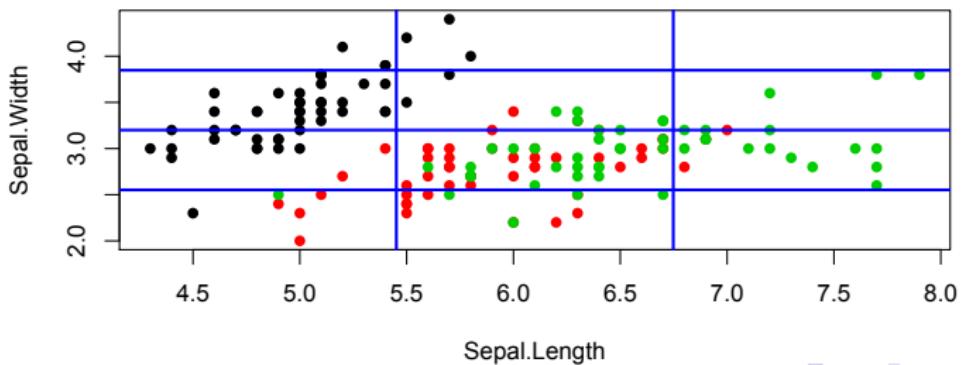
## Example (grid)

```
> attach(iris)
> str(iris)
'data.frame': 150 obs. of 5 variables:
 $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
> head(iris)
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1        3.5         1.4        0.2  setosa
2          4.9        3.0         1.4        0.2  setosa
3          4.7        3.2         1.3        0.2  setosa
4          4.6        3.1         1.5        0.2  setosa
5          5.0        3.6         1.4        0.2  setosa
6          5.4        3.9         1.7        0.4  setosa
> par(mfrow=c(2,1))
> plot(Sepal.Length, Sepal.Width, pch=16, col=as.integer(Species))
> grid()
> title("grid()")
> plot(Sepal.Length, Sepal.Width, pch=16, col=as.integer(Species))
> grid(3,4,lty=1,lwd=2,col="blue")
> title("grid(3,4,lty=1,lwd=2,col=\"blue\")")
> detach(iris)
```

**grid()**



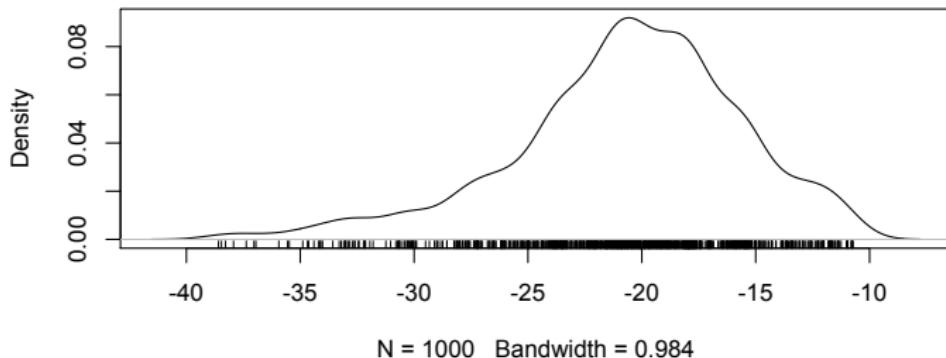
**grid(3,4,lty=1,lwd=2,col="blue")**



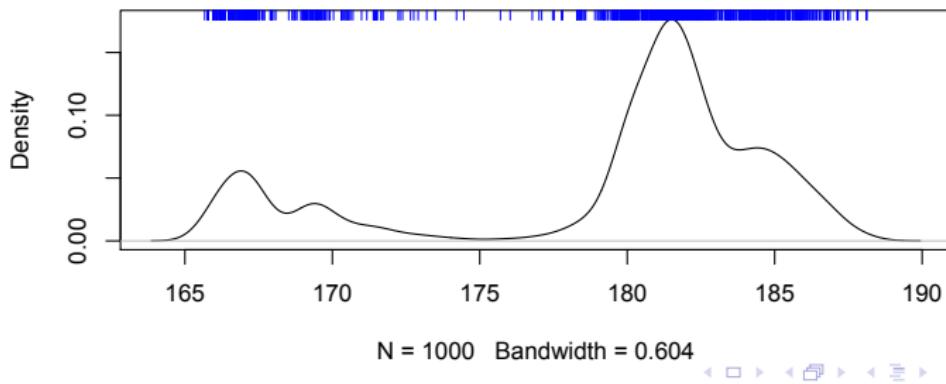
## Example (rug)

```
> attach(quakes)
> str(quakes)
'data.frame': 1000 obs. of 5 variables:
$ lat      : num -20.4 -20.6 -26 -18 -20.4 ...
$ long     : num 182 181 184 182 182 ...
$ depth    : int 562 650 42 626 649 195 82 194 211 622 ...
$ mag      : num 4.8 4.2 5.4 4.1 4 4 4.8 4.4 4.7 4.3 ...
$ stations: int 41 15 43 19 11 12 43 15 35 19 ...
> head(quakes)
   lat   long depth mag stations
1 -20.42 181.62 562 4.8      41
2 -20.62 181.03 650 4.2      15
3 -26.00 184.10 42 5.4      43
4 -17.97 181.66 626 4.1      19
5 -20.42 181.96 649 4.0      11
6 -19.68 184.31 195 4.0      12
> par(mfrow=c(2,1))
> plot(density(lat),main="rug(lat)")
> rug(lat)
> plot(density(long),main="side=3, col=\"blue\",lwd=0.8,ticksize=0.04")
> rug(long,side=3,col="blue",lwd=0.8,ticksize=0.04)
> detach(quakes)
```

**rug(lat)**



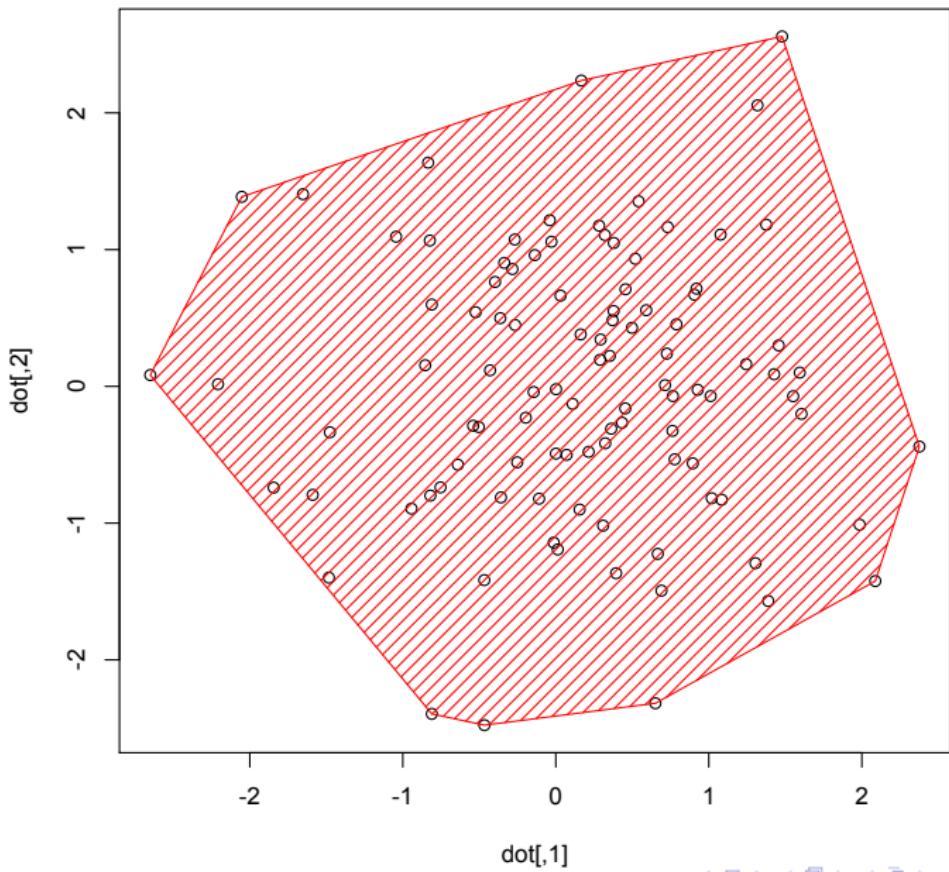
**side=3, col="blue", lwd=0.8,ticksize=0.04**



## Example (chull)

```
> dot<-matrix(rnorm(200),ncol=2)
> plot(dot)
> chull.data<-chull(dot)
> polygon(dot[chull.data,],angle=45,density=15,col="red")
> title(main="Polygon by chull")
```

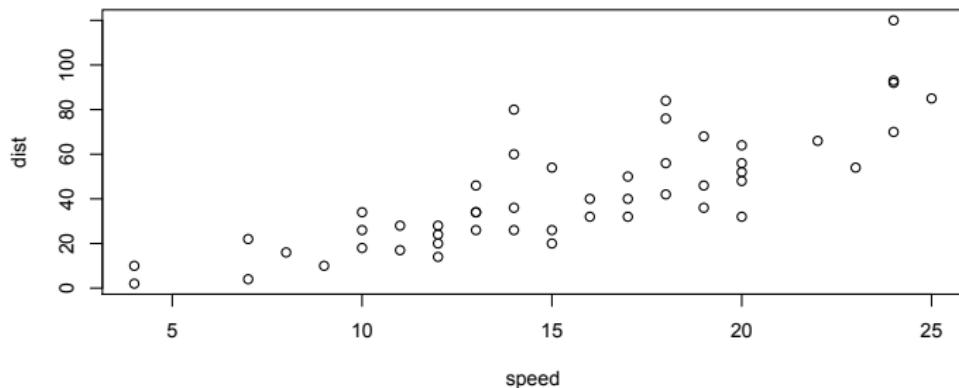
## Polygon by chull



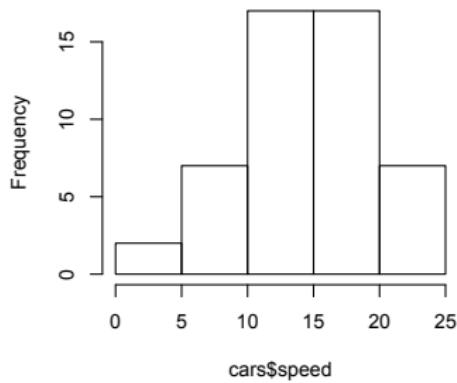
## Example (layout)

```
> m<-matrix(c(1,1,2,3),ncol=2,byrow=T)
> m
     [,1] [,2]
[1,]    1    1
[2,]    2    3
> layout(mat=m)
> plot(cars,main="scatter plot of cars data")
> hist(cars$speed)
> hist(cars$dist)
```

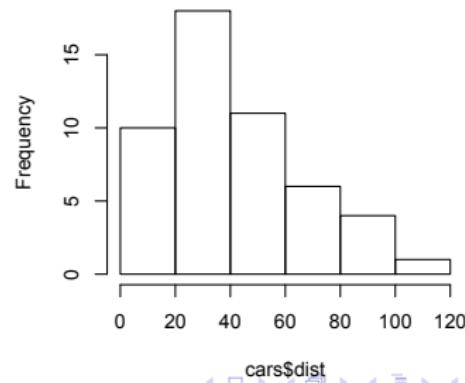
scatter plot of cars data



Histogram of cars\$speed



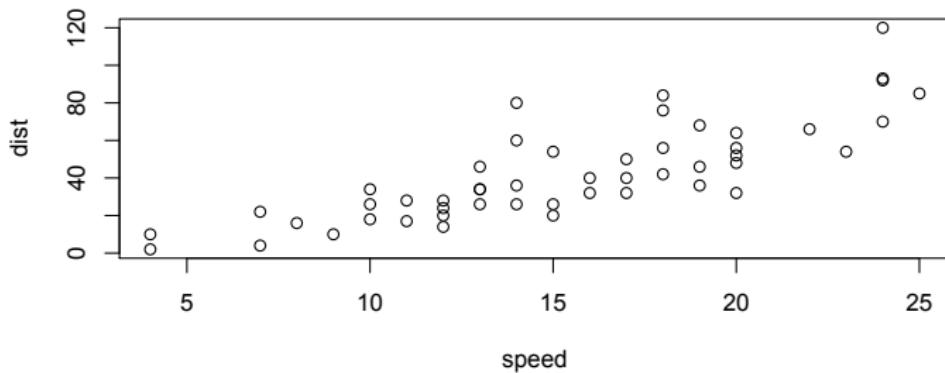
Histogram of cars\$dist



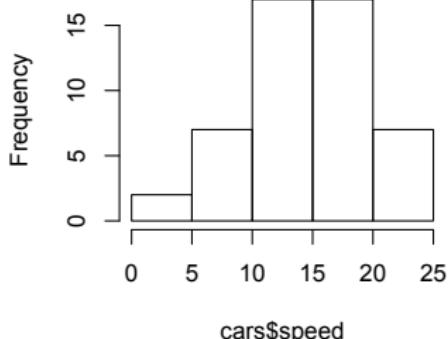
## Example (split.screen)

```
> par(bg="white")      # background color is set by white. Default is transparent
> split.screen(fig=c(2,1))    # split screen into top(1) and bottom(2)
[1] 1 2
> split.screen(c(1,2),screen=2)    # split bottom screen(2) into left(3) and right(4)
[1] 3 4
> screen(n=3)      # set screen 3
> hist(cars$speed)
> screen(1)      # set screen 1
> plot(cars,main="scatter plot of cars data by split.screen")
> screen(4)      # set screen 4
> hist(cars$dist)
> erase.screen(n=1)    # erase plot in screen 1 by pouring bg color
> screen(1)      # set screen 1
> plot(cars,main="scatter plot of cars data by split.screen")
> close.screen(all=TRUE)      # finish screen split definition
```

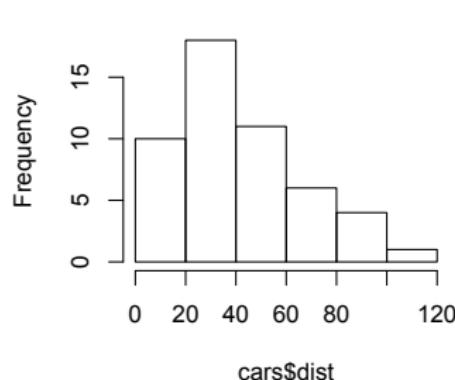
### scatter plot of cars data by split.screen



Histogram of cars\$speed



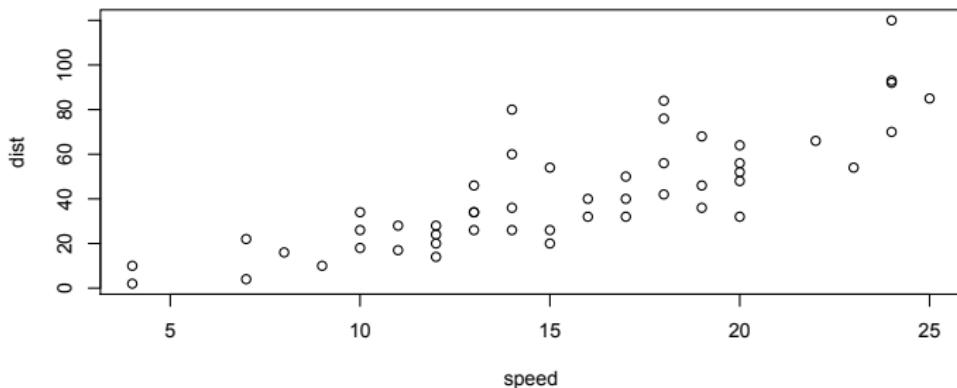
Histogram of cars\$dist



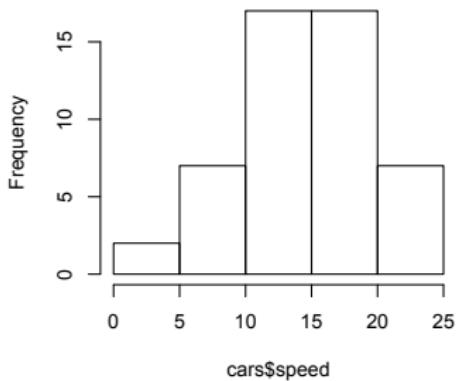
## Example (par(fig))

```
> par(fig=c(0,1,0.5,1))
> plot(cars,main="scatter plot of cars data by fig")
> par(fig=c(0,0.5,0,0.5),new=T)
> hist(cars$speed,main="Histogram of cars$speed by fig")
> par(fig=c(0.5,1,0,0.5),new=T)
> hist(cars$dist,main="Histogram of cars$dist by fig")
```

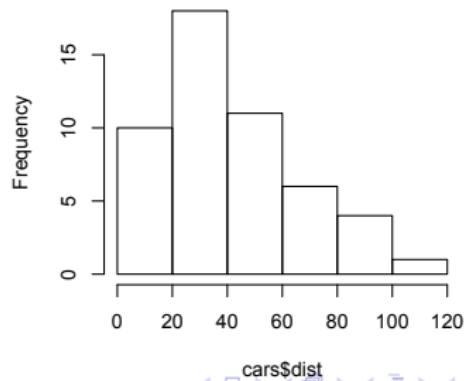
scatter plot of cars data by fig



Histogram of cars\$speed by fig



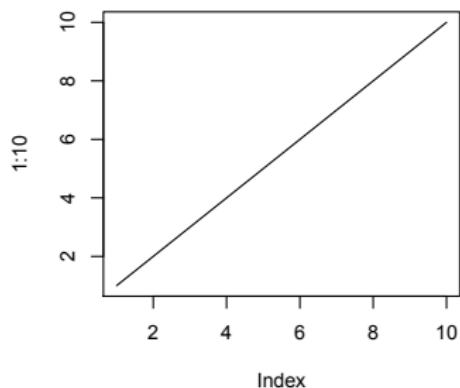
Histogram of cars\$dist by fig



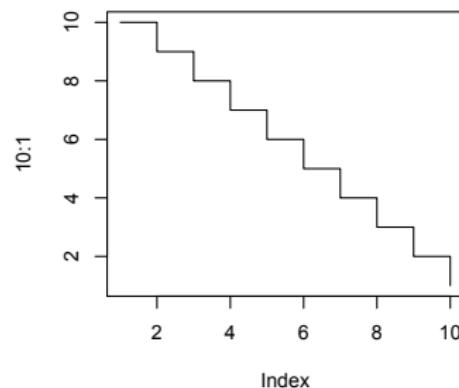
## Example (par(new))

```
# new:  
# logical, defaulting to FALSE. If set to TRUE, the next high-level plotting  
# command (actually plot.new) should not clean the frame before drawing as if it  
# were on a new device. It is an error (ignored with a warning) to try to use  
# new = TRUE on a device that does not currently contain a high-level plot.  
#  
> par(mfrow=c(2,2))  
> plot(1:10,type="l",main="plot")  
> par(new=F)  
> plot(10:1,type="s",main="plot by new=F")  
> plot(1:10,type="l")  
> par(new=T)  
> plot(10:1,type="s",main="plot by new=T")  
> x<-rnorm(10)  
> plot(x)  
> par(new=T)  
> hist(x)
```

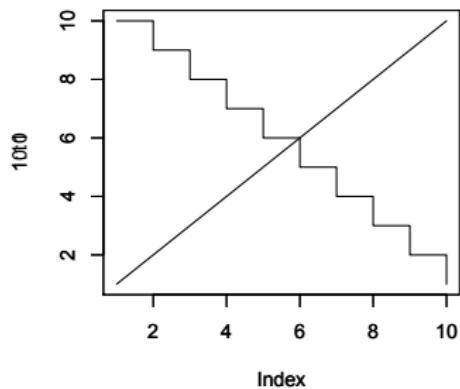
**plot**



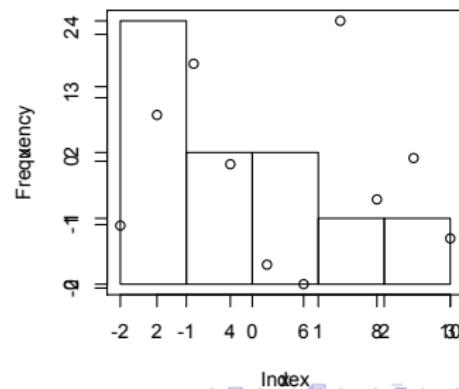
**plot by new=F**



**plot by new=T**



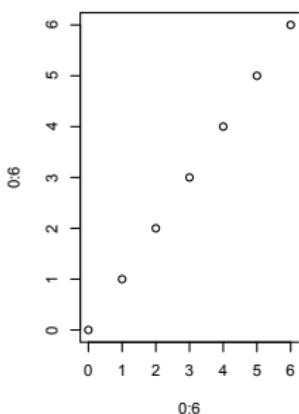
**Histogram of x**



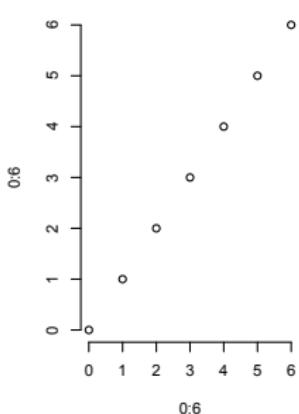
## Example (par(bty))

```
# bty: box type
# A character string which determined the type of box which is drawn about plots.
# If bty is one of "o" (the default), "l", "7", "c", "u", or "]" the resulting box
# resembles the corresponding upper case letter. A value of "n" suppresses the box.
#
> par(mfrow=c(2,3),bty="l")      # default is "0". "l" is set for all subfigures
> plot(0:6,0:6,bty="c",main="bty=\"c\"")      # C-shape (1, 2, 3 areas)
> plot(0:6,0:6,bty="n",main="bty=\"n\"")      # no
> plot(0:6,0:6,bty="o",main="bty=\"o\"")      # O-shape (1, 2, 3, 4)
> plot(0:6,0:6,bty="7",main="bty=\"7\"")      # 7-shape (3, 4)
> plot(0:6,0:6,bty="u",main="bty=\"u\"")      # U-shape (1, 2, 4)
> plot(0:6,0:6,main="bty=\"1\"")      # L-shape (1, 2)
```

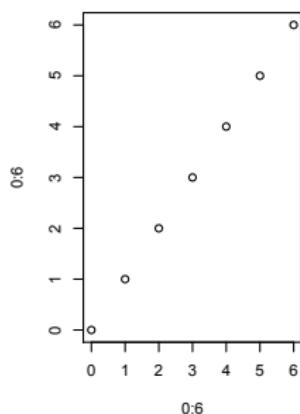
**bty="c"**



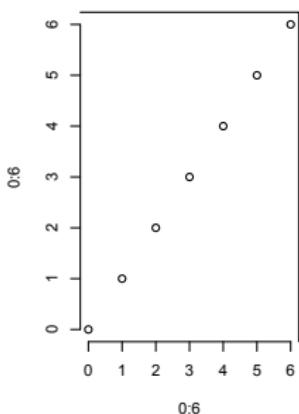
**bty="n"**



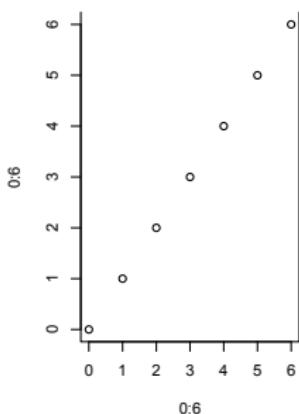
**bty="o"**



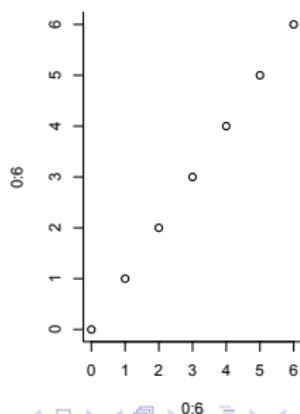
**bty="7"**



**bty="u"**



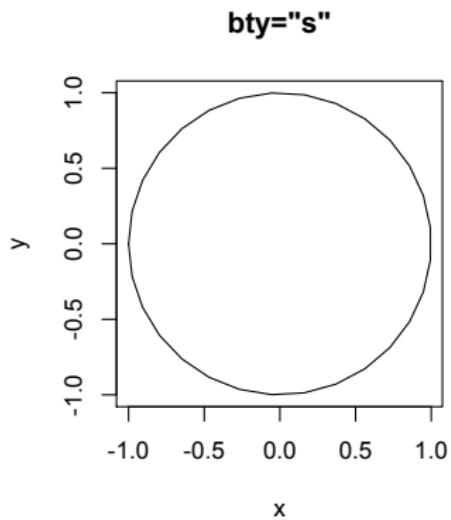
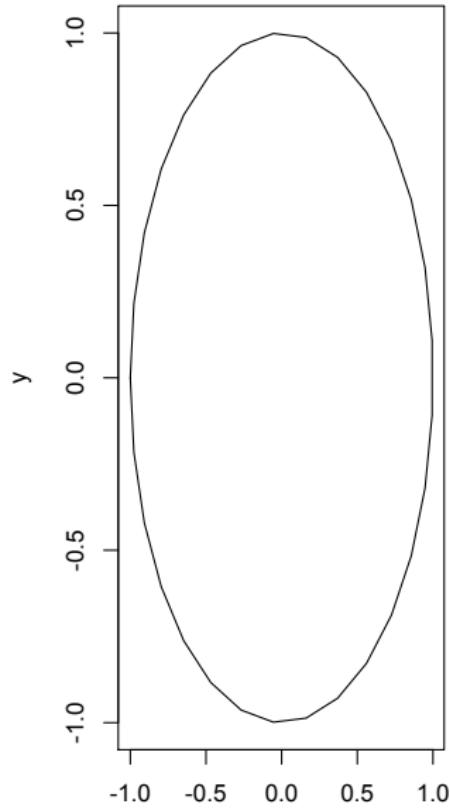
**bty="l"**



## Example (par(pty))

```
# pty: plot type
# A character specifying the type of plot region to be used; "s" generates
# a square plotting region and "m" generates the maximal plotting region.
#
> theta<-seq(-pi,pi,length=30)
> x<-cos(theta)
> y<-sin(theta)
> par(mfrow=c(1,2),pty="s",bty="o")
> plot(x,y,type="l",main="bty=\"s\"")
> par(pty="m")
> plot(x,y,type="l",main="bty=\"m\"")
```

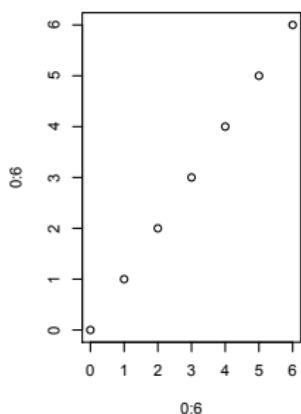
**bty="m"**



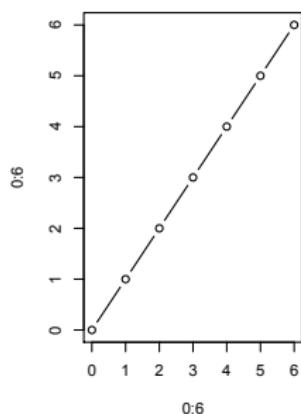
## Example (par(type))

```
# type:  
# what type of plot should be drawn. Possible types are  
# "p" for points,  
# "l" for lines,  
# "b" for both,  
# "c" for the lines part alone of "b",  
# "o" for both 'overplotted',  
# "h" for 'histogram' like (or 'high-density') vertical lines,  
# "s" for stair steps,  
# "S" for other steps, see 'Details' below,  
# "n" for no plotting.  
#  
> par(mfrow=c(2,3))  
> plot(0:6,0:6,main="default")  
> plot(0:6,0:6,type="b",main="type=\"b\"")  
> plot(0:6,0:6,type="c",main="type=\"c\"")  
> plot(0:6,0:6,type="o",main="type=\"o\"")  
> plot(0:6,0:6,type="s",main="type=\"s\"")  
> plot(0:6,0:6,type="S",main="type=\"S\"")
```

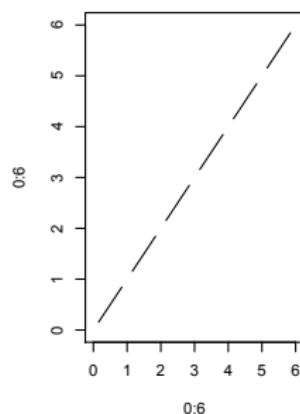
**default**



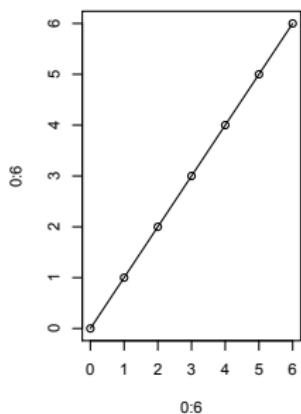
**type="b"**



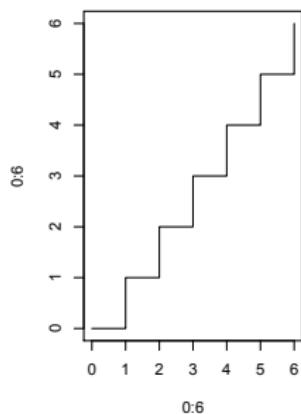
**type="c"**



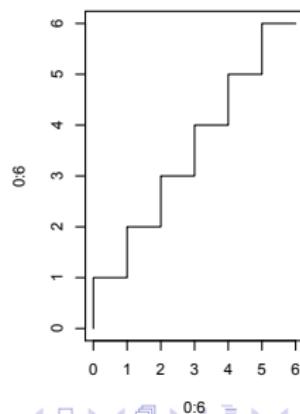
**type="o"**



**type="s"**



**type="S"**

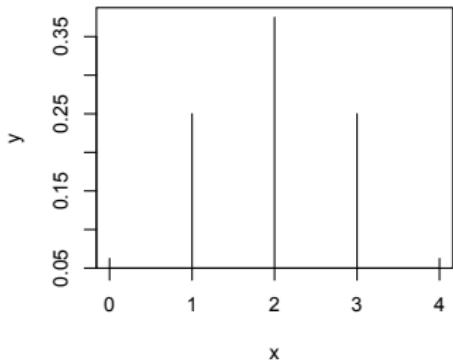


## Example (par(xlim, ylim))

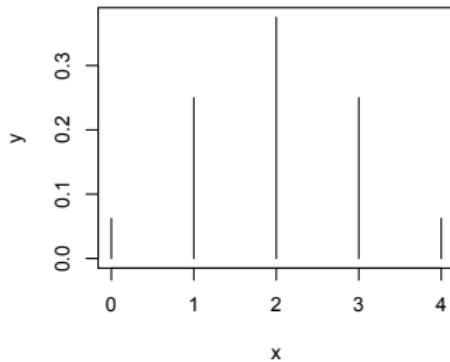
```
# xlim, ylim:  
#  
> x<-0:4  
> (y<-dbinom(x,size=4,prob=0.5))  
[1] 0.0625 0.2500 0.3750 0.2500 0.0625  
> par(oma=c(0,0,2,0),mfrow=c(2,2))  
> plot(x,y,type="h",main="default")  
> plot(x,y,type="h",ylim=c(0,max(y)),main="ylim=c(0,max(y))")  
> plot(x,y,type="h",ylim=c(0.1,0.3),main="ylim=c(0.1,0.3)")  
> plot(x,y,type="h",xlim=c(1,3),main="xlim=c(1,3)")  
> title(main="binomial density",line=0,outer=T)
```

### binomial density

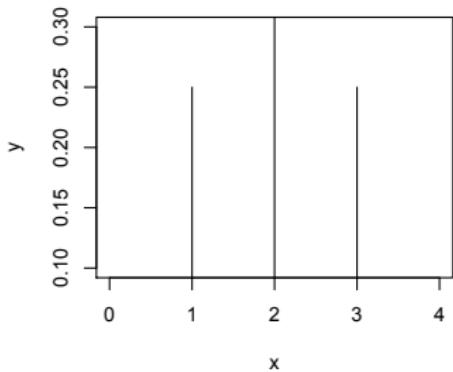
default



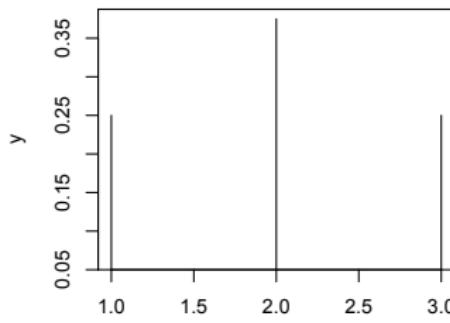
ylim=c(0,max(y))



ylim=c(0.1,0.3)



xlim=c(1,3)

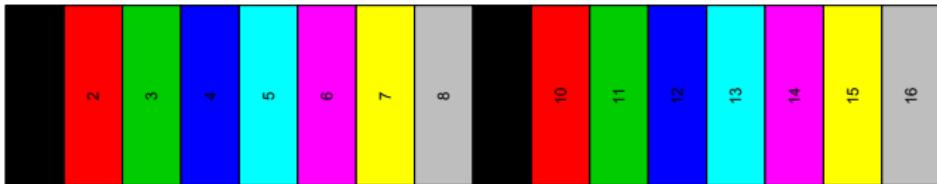


## Example (par(col))

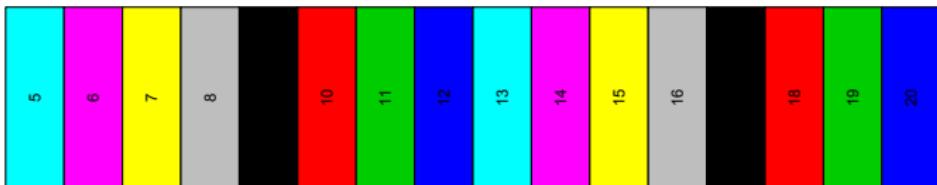
```
# col: color
# A specification for the default plotting color. See section 'Color Specification'.
# Some functions such as lines and text accept a vector of values which are
# recycled and may be interpreted slightly differently.
#
> help(plot)
> help(par)
> help(plot)
> col.table<-function(cols){
+   n=length(cols)
+   plot(seq(n),rep(1,n),xlim=c(0,n),ylim=c(0,1),type="n",xlab="",ylab="",axes=F)
+   title(paste("Color Table by",deparse(substitute(cols))))
+   for(i in 1:n){
+     polygon(c(i-1,i-1,i,i),c(0.05,1,1,0.05),col=cols[i])
+     text(mean(c(i-1,i)),0.52,labels=cols[i],srt=90,adj=0.5)
+   }
+ }
> par(mfrow=c(4,1),mar=c(0,0,2,0))
> col.table(1:16)
> col.table(5:20)
> col.table(rainbow(20))
> col.table(heat.colors(20))
>
> dev.new()
> par(mfrow=c(4,1),mar=c(0,0,2,0))
> col.table(terrain.colors(20))
> col.table(topo.colors(20))
> col.table(cm.colors(20))
> col.table(colors()[1:20])
```



Color Table by 1:16



Color Table by 5:20



Color Table by rainbow(20)



Color Table by heat.colors(20)



Color Table by terrain.colors(20)

white	#00FFFFFF
aliceblue	#F0F8FF
antiquewhite	#FAEBD7
antiquewhite1	#FFFACD
antiquewhite2	#F0E68C
antiquewhite3	#FADBD8
antiquewhite4	#F0E68C
aquamarine	#70D9CF
aquamarine1	#A9F5E0
aquamarine2	#B0EBF2
aquamarine3	#C0F9E6
aquamarine4	#D0E0E3
azure	#F0F8FF
azure1	#F0F8FF
azure2	#F0F8FF
azure3	#E6EAF2
azure4	#D9D9D9
beige	#FFF2E0
bisque	#FFF0E6
bisque1	#FFF0E6

Color Table by topo.colors(20)

white	#00FFFFFF
blue	#0000CD
darkblue	#00008B
darkred	#8B0000
darkgreen	#008B00
darkcyan	#00FFFF
darkmagenta	#8B008B
darkorange	#FF8C00
darkslategray	#2F4F4F
darkslateblue	#4682B4
darkkhaki	#BDB76B
darkolivegreen	#6B8E23
darktan	#A9A9A9
darkseagreen	#808080
darkgray	#A9A9A9
lightgray	#D3D3D3
lightbrown	#D9C3B8
lightolivegreen	#B0C4DE
lighttan	#D9C3B8
lightseagreen	#228B22
lightgray2	#D3D3D3
lightgray3	#C9C9C9
lightgray4	#B3B3B3
lightgray5	#A9A9A9
lightgray6	#939393
lightgray7	#808080
lightgray8	#707070
lightgray9	#606060
lightgray10	#505050
lightgray11	#404040
lightgray12	#303030
lightgray13	#202020
lightgray14	#101010
lightgray15	#000000

Color Table by cm.colors(20)

white	#00FFFFFF
aliceblue	#F0F8FF
antiquewhite	#FAEBD7
antiquewhite1	#FFFACD
antiquewhite2	#F0E68C
antiquewhite3	#FADBD8
antiquewhite4	#F0E68C
aquamarine	#70D9CF
aquamarine1	#A9F5E0
aquamarine2	#B0EBF2
aquamarine3	#C0F9E6
aquamarine4	#D0E0E3
azure	#F0F8FF
azure1	#F0F8FF
azure2	#F0F8FF
azure3	#E6EAF2
azure4	#D9D9D9
beige	#FFF2E0
bisque	#FFF0E6
bisque1	#FFF0E6

Color Table by colors()(1:20]

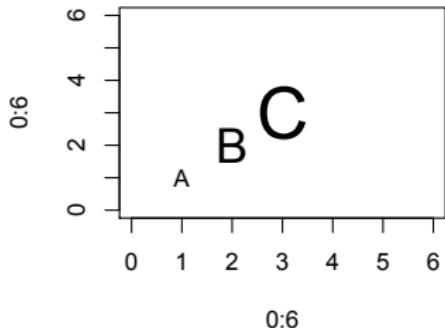
white	#00FFFFFF
aliceblue	#F0F8FF
antiquewhite	#FAEBD7
antiquewhite1	#FFFACD
antiquewhite2	#F0E68C
antiquewhite3	#FADBD8
antiquewhite4	#F0E68C
aquamarine	#70D9CF
aquamarine1	#A9F5E0
aquamarine2	#B0EBF2
aquamarine3	#C0F9E6
aquamarine4	#D0E0E3
azure	#F0F8FF
azure1	#F0F8FF
azure2	#F0F8FF
azure3	#E6EAF2
azure4	#D9D9D9
beige	#FFF2E0
bisque	#FFF0E6
bisque1	#FFF0E6

## Example (par(cex))

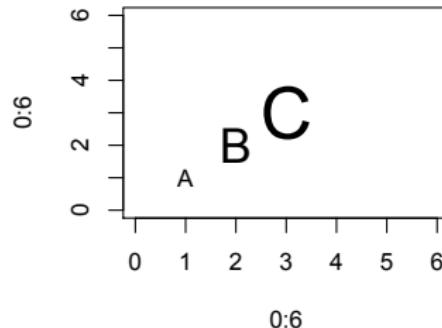
```
# cex: character expansion
# A numerical value giving the amount by which plotting text and symbols should be
# magnified relative to the default. This starts as 1 when a device is opened, and
# is reset when the layout is changed, e.g. by setting mfrow.
#
> par(mfrow=c(2,2),oma=c(0,0,2,0),cex=1)
> plot(0:6,0:6,type="n",main="cex in text")
> text(1:3,1:3,labels=LETTERS[1:3],cex=1:3)
> plot(0:6,0:6,type="n",cex=2,main="cex in plot")
> text(1:3,1:3,labels=LETTERS[1:3],cex=1:3)
> par(cex=1.2)
> plot(0:6,0:6,type="n",main="cex in par")
> text(1:3,1:3,labels=LETTERS[1:3],cex=1:3)
> plot(0:6,0:6,type="n",main="cex in par")
> text(1:3,1:3,labels=c("⊤","⊥","⊤"),cex=1:3)
> points(3:5,1:3,pch=1:3,cex=1:3)
> title(main="cex",line=0,outer=T)
```

**cex**

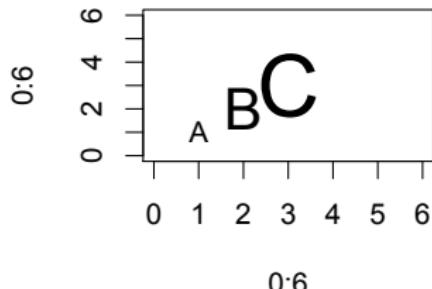
**cex in text**



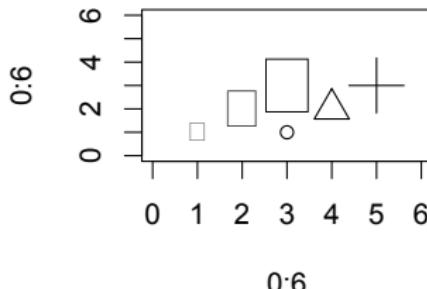
**cex in plot**



**cex in par**

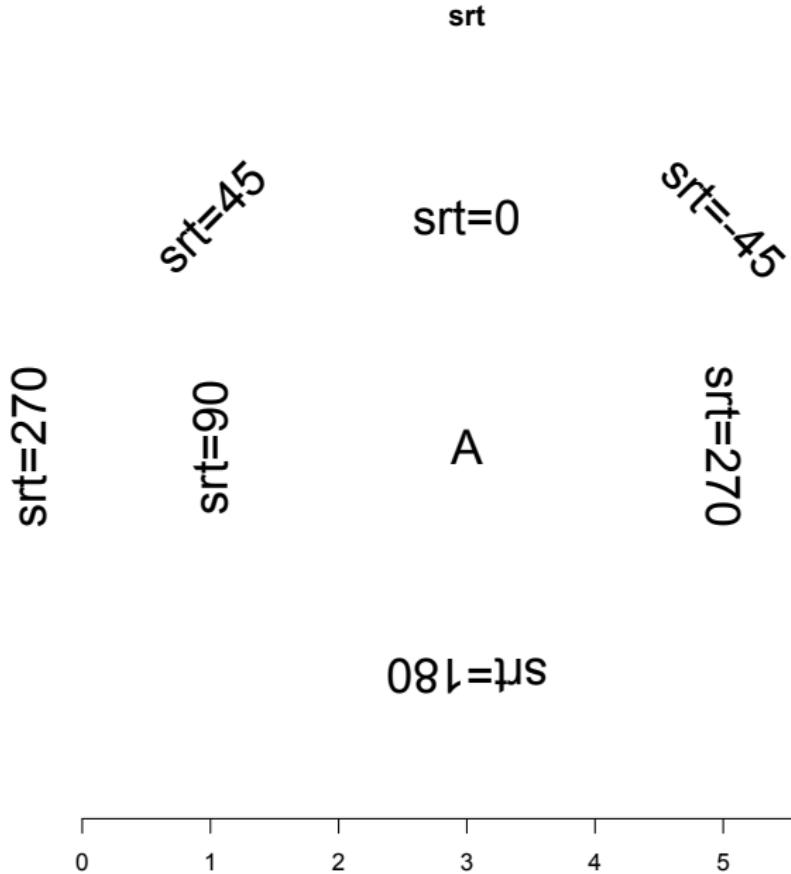


**cex in par**



## Example (par(srt))

```
# srt: string rotation
# The string rotation in degrees. See the comment about crt. Only supported by text.
#
> par("srt")
[1] 0
> plot(0:6,0:6,type="n",axes=F,xlab="",ylab="")
> text(3,5,"srt=0",srt=0,cex=2)
> text(1,3,"srt=90",srt=90,cex=2)
> text(3,1,"srt=180",srt=180,cex=2)
> text(5,3,"srt=270",srt=270,cex=2)
> text(5,5,"srt=-45",srt=-45,cex=2)
> text(1,5,"srt=45",srt=45,cex=2)
> points(3,3,pch="A",srt=45,cex=2)
> title("srt",srt=45)
> mtext(side=2,"srt=270",srt=270,cex=2)
> axis(side=1,srt=45)
```

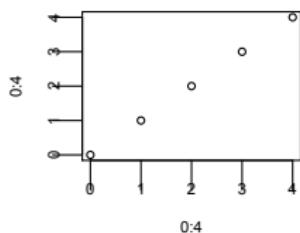


## Example (par(tck))

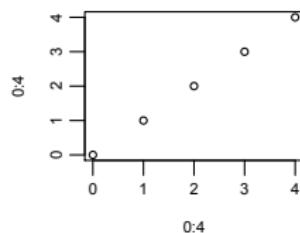
```
# tck: tick marks
# The length of tick marks as a fraction of the width or height of
# the plotting region. If tck >= 0.5 it is interpreted as a fraction of
# the relevant side, so if tck = 1 grid lines are drawn. The default
# setting (tck = NA) is to use tcl = -0.5.
#
> par(mfrow=c(3,3),oma=c(0,0,2,0))
> plot(0:4,0:4,tck=-0.2,main="tck=-0.2")
> plot(0:4,0:4,tck=-0.1,main="tck=-0.1")
> plot(0:4,0:4,tck=0,main="tck=0")
> plot(0:4,0:4,tck=0.3,main="tck=0.3")
> plot(0:4,0:4,tck=0.5,main="tck=0.5")
> plot(0:4,0:4,tck=0.7,main="tck=0.7")
> plot(0:4,0:4,tck=1,main="tck=1")
> par(tck=0.2)
> plot(0:4,0:4,main="tck defined in par")
> plot(0:4,0:4,tck=-0.1,main="tck defined in both")
> title(main="tck",line=0,outer=T)
```

tck

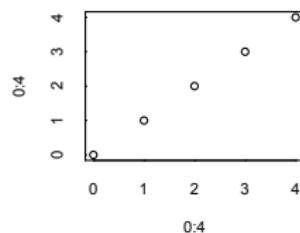
tck=-0.2



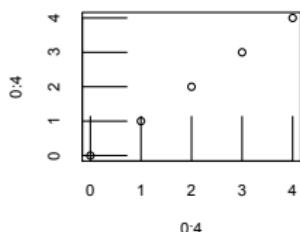
tck=-0.1



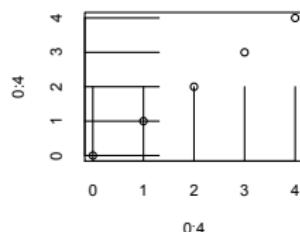
tck=0



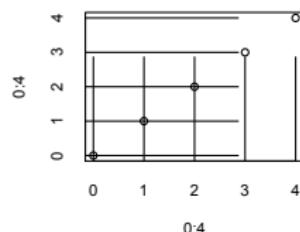
tck=0.3



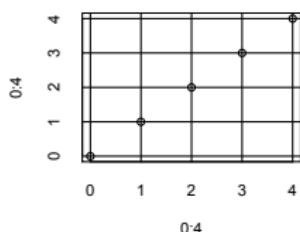
tck=0.5



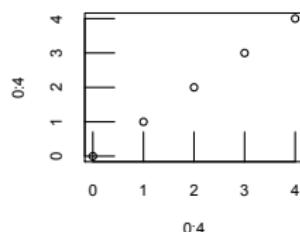
tck=0.7



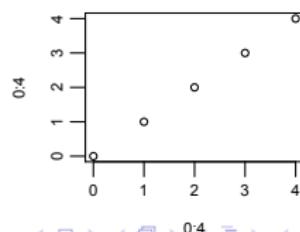
tck=1



tck defined in par

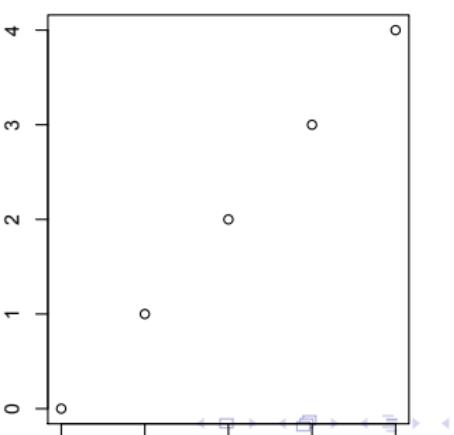
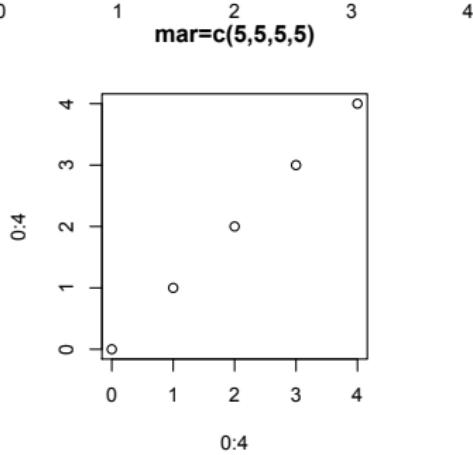
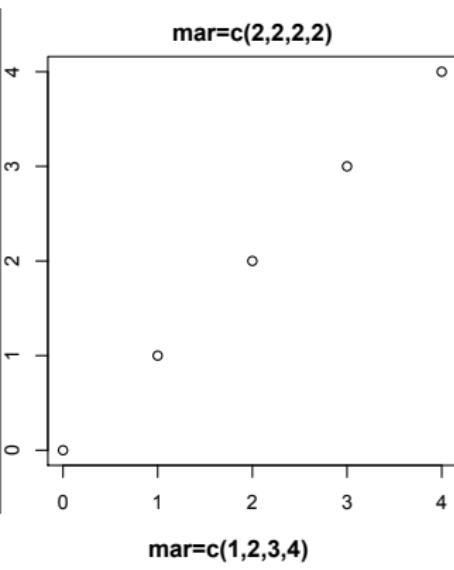
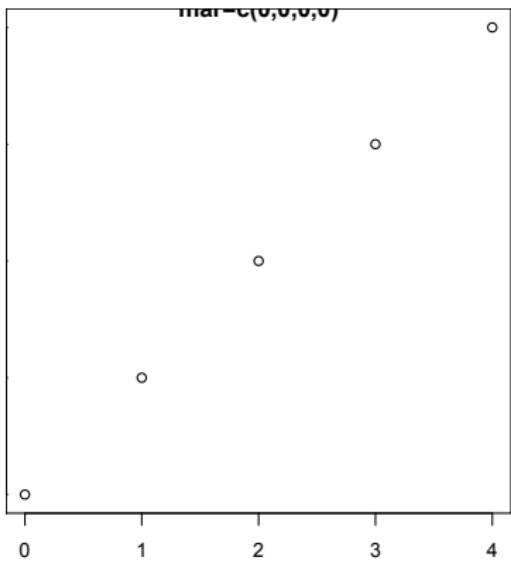


tck defined in both



## Example (par(mar))

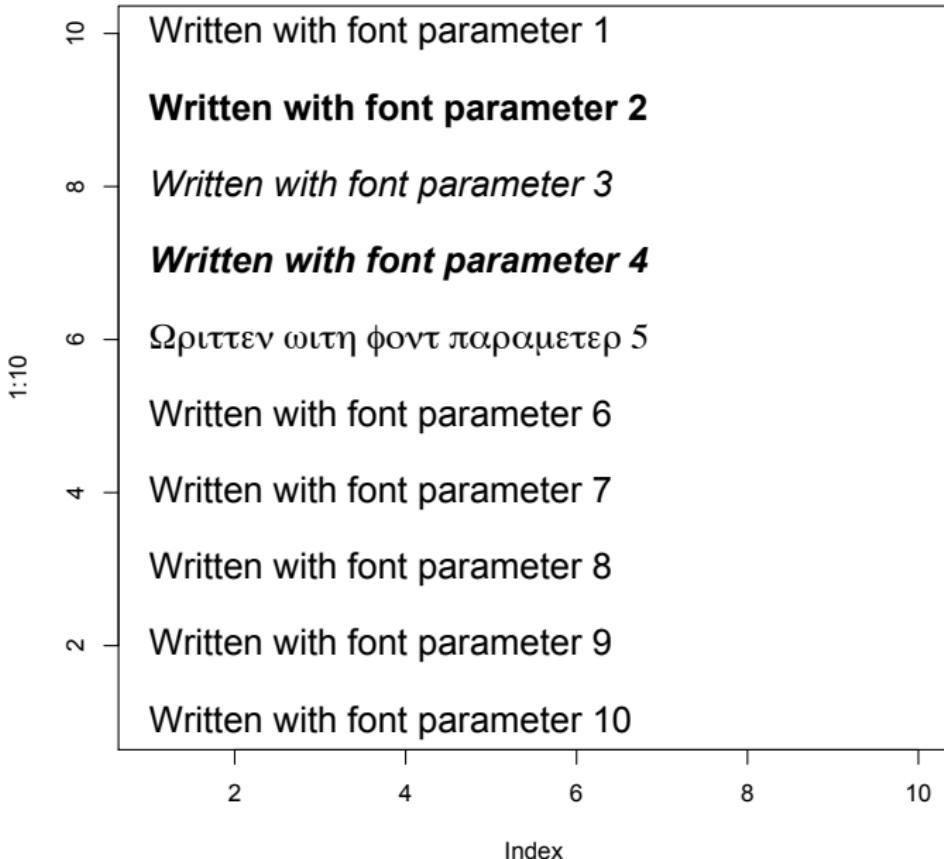
```
# mar: margin
# A numerical vector of the form c(bottom, left, top, right) which gives
# the number of lines of margin to be specified on the four sides of the plot.
# The default is c(5, 4, 4, 2) + 0.1.
#
> par(mfrow=c(2,2))
> par("mar")
[1] 5.1 4.1 4.1 2.1
> # Figure 1
> par(mar=c(0,0,0,0))
> plot(0:4,0:4,main="mar=c(0,0,0,0)")
> # Figure 2
> par(mar=c(2,2,2,2))
> plot(0:4,0:4,main="mar=c(2,2,2,2)")
> # Figure 3
> par(mar=c(5,5,5,5))
> plot(0:4,0:4,main="mar=c(5,5,5,5)")
> # Figure 4
> par(mar=c(1,2,3,4))
> plot(0:4,0:4,main="mar=c(1,2,3,4)")
```



## Example (par(font))

```
# font: font
# An integer which specifies which font to use for text. If possible, device
# drivers arrange so that 1 corresponds to plain text (the default), 2 to bold face,
# 3 to italic and 4 to bold italic. Also, font 5 is expected to be the symbol font,
# in Adobe symbol encoding. On some devices font families can be selected by
# family to choose different sets of 5 fonts.
#
> plot(1:10, type="n", main="par(font)")
> lab<- "Written with font parameter"
> for(i in 1:10){
+   par(font=i)
+   text(1,11-i,labels=paste(lab,i),adj=0,cex=1.5)
+ }
```

**par(font)**



## Example (par(fg, bg))

```
# fg: foreground color
# The color to be used for the foreground of plots. This is the default color used
# for things like axes and boxes around plots. When called from par() this also
# sets parameter col to the same value. See section ‘Color Specification’. A few
# devices have an argument to set the initial value, which is otherwise "black".
#
# bg: background color
# The color to be used for the background of the device region. When called from
# par() it also sets new=FALSE. See section ‘Color Specification’ for suitable
# values. For many devices the initial value is set from the bg argument of
# the device, and for the rest it is normally "white".
#
> unlist(par("fg","bg"))
      fg          bg
    "black" "transparent"
> par(bg="thistle",fg="blue")
> hist(rnorm(30),main="bg=\"thistle\"", fg="blue")
```

**bg="thistle", fg="blue"**

