Aula 3.7 - parametros globais

Elder Sodre

Wednesday, February 25, 2015

#### Preparando os dados

Para a aula de hoje, usaremos como exemplo alguns dados com os quais vocês já devem estar acostumados das últimas aulas.

setwd("C:\\Users\\Elder\\Desktop\\Minicurso Elementos de Linguagem R")  
  
#Dados do ecolagoas de 2008:  
ecolagoas <- read.csv2("ecolagoas2008.csv",na.strings="-",row.names=1)  
  
#Dados de zooplâncton do lago Batata em 1998:  
batata <- read.csv2("zoobatata1998.csv",row.names=1)  
zoo <- batata[,3:53] #apenas as colunas com as densidades  
  
#Variáveis x e y:  
x <- 1:20  
set.seed(635)  
y <- 1:20 + rnorm(20,sd=3)  
  
#Variável a:  
set.seed(60)  
a <- rnorm(123)

A função par() altera parâmetros globais de gráficos. Isto significa que, uma vez alterados, eles afetarão todos os gráficos daquele momento em diante.

A função par(), sem argumentos, retorna uma lista com todos os parâmetros e seus valores atuais:

par()

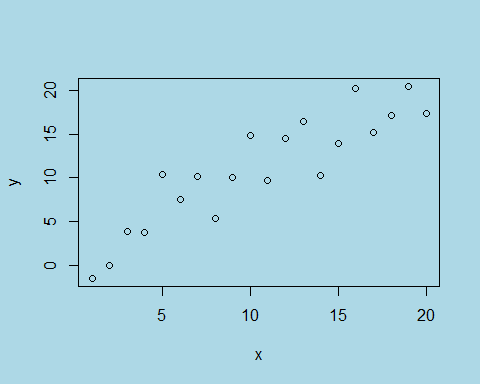
## $xlog  
## [1] FALSE  
##   
## $ylog  
## [1] FALSE  
##   
## $adj  
## [1] 0.5  
##   
## $ann  
## [1] TRUE  
##   
## $ask  
## [1] FALSE  
##   
## $bg  
## [1] "transparent"  
##   
## $bty  
## [1] "o"  
##   
## $cex  
## [1] 1  
##   
## $cex.axis  
## [1] 1  
##   
## $cex.lab  
## [1] 1  
##   
## $cex.main  
## [1] 1.2  
##   
## $cex.sub  
## [1] 1  
##   
## $cin  
## [1] 0.15 0.20  
##   
## $col  
## [1] "black"  
##   
## $col.axis  
## [1] "black"  
##   
## $col.lab  
## [1] "black"  
##   
## $col.main  
## [1] "black"  
##   
## $col.sub  
## [1] "black"  
##   
## $cra  
## [1] 10.8 14.4  
##   
## $crt  
## [1] 0  
##   
## $csi  
## [1] 0.2  
##   
## $cxy  
## [1] 0.03989 0.09259  
##   
## $din  
## [1] 5 4  
##   
## $err  
## [1] 0  
##   
## $family  
## [1] ""  
##   
## $fg  
## [1] "black"  
##   
## $fig  
## [1] 0 1 0 1  
##   
## $fin  
## [1] 5 4  
##   
## $font  
## [1] 1  
##   
## $font.axis  
## [1] 1  
##   
## $font.lab  
## [1] 1  
##   
## $font.main  
## [1] 2  
##   
## $font.sub  
## [1] 1  
##   
## $lab  
## [1] 5 5 7  
##   
## $las  
## [1] 0  
##   
## $lend  
## [1] "round"  
##   
## $lheight  
## [1] 1  
##   
## $ljoin  
## [1] "round"  
##   
## $lmitre  
## [1] 10  
##   
## $lty  
## [1] "solid"  
##   
## $lwd  
## [1] 1  
##   
## $mai  
## [1] 1.02 0.82 0.82 0.42  
##   
## $mar  
## [1] 5.1 4.1 4.1 2.1  
##   
## $mex  
## [1] 1  
##   
## $mfcol  
## [1] 1 1  
##   
## $mfg  
## [1] 1 1 1 1  
##   
## $mfrow  
## [1] 1 1  
##   
## $mgp  
## [1] 3 1 0  
##   
## $mkh  
## [1] 0.001  
##   
## $new  
## [1] FALSE  
##   
## $oma  
## [1] 0 0 0 0  
##   
## $omd  
## [1] 0 1 0 1  
##   
## $omi  
## [1] 0 0 0 0  
##   
## $page  
## [1] TRUE  
##   
## $pch  
## [1] 1  
##   
## $pin  
## [1] 3.76 2.16  
##   
## $plt  
## [1] 0.164 0.916 0.255 0.795  
##   
## $ps  
## [1] 12  
##   
## $pty  
## [1] "m"  
##   
## $smo  
## [1] 1  
##   
## $srt  
## [1] 0  
##   
## $tck  
## [1] NA  
##   
## $tcl  
## [1] -0.5  
##   
## $usr  
## [1] 0 1 0 1  
##   
## $xaxp  
## [1] 0 1 5  
##   
## $xaxs  
## [1] "r"  
##   
## $xaxt  
## [1] "s"  
##   
## $xpd  
## [1] FALSE  
##   
## $yaxp  
## [1] 0 1 5  
##   
## $yaxs  
## [1] "r"  
##   
## $yaxt  
## [1] "s"  
##   
## $ylbias  
## [1] 0.2

?par

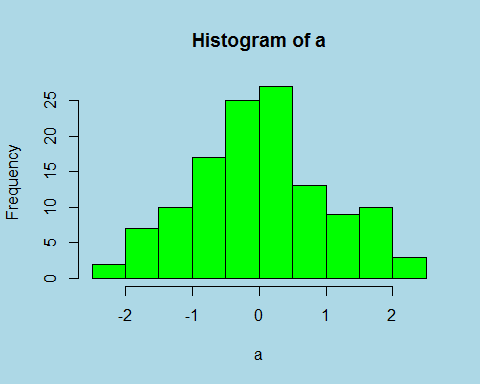
## starting httpd help server ... done

Por exemplo, bg é o fundo do gráfico. Podemos alterar este parâmetro com o comando abaixo:

par(bg="lightblue")  
#De agora em diante, todos os gráficos que fizermos terão um fundo azul claro.  
  
plot(x,y)

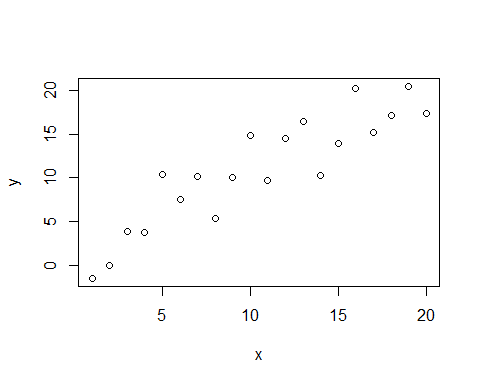


hist(a,col="green")



IMPORTANTE: Não existe nenhuma função que retorne automaticamente os parâmetros globais ao default. Para retornar ao default, você precisa fazer isso manualmente.

par(bg="white")  
plot(x,y)

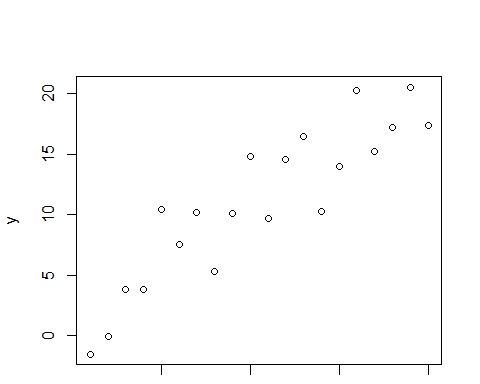


Outro parâmetro global é mar, que especifica o tamanho das margens. Ele é um vetor com 4 elementos, na fórmula: c(embaixo,esquerda,acima,direita).

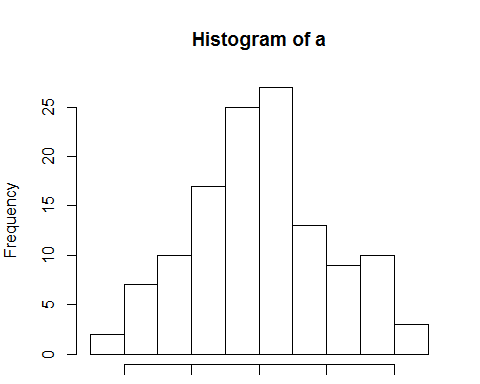
Seu default é c(5, 4, 4, 2) + 0.1.

Vejamos o que acontece quando mexemos nele:

par(mar=c(1,4,4,2)) #Diminuí as margens abaixo do gráfico.  
plot(x,y)

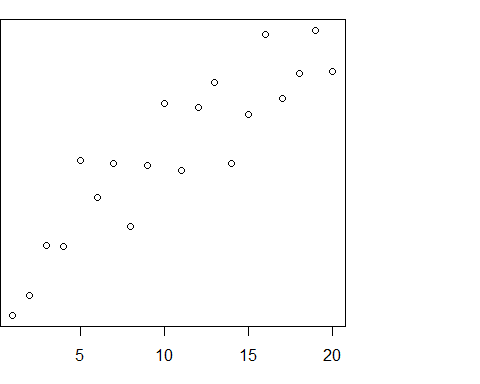


hist(a)

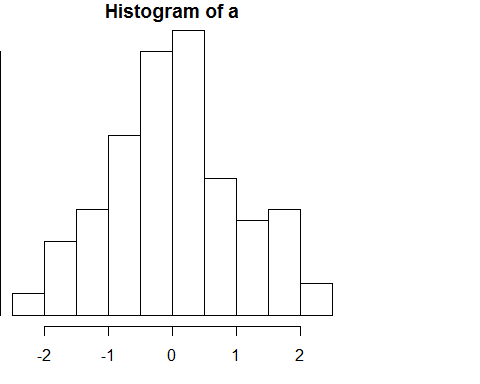


Observe que eixos e títulos podem ser cortados ao diminuir o tamanho das margens.

par(mar=c(3,0,1,7)) #Diminuí as margens abaixo do gráfico.  
plot(x,y)



hist(a)



par(mar=c(5,4,4,2)+0.1)

#### Alterando o número de linhas e colunas

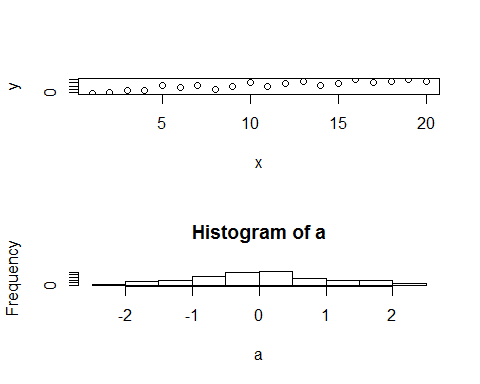
Há muitos parâmetros globais, mas os mais importantes são provavelmente mfrow e mfcol.

mfrow e mfcol fazem exatamente a mesma coisa. São usados : mfrow=c(linhas,colunas)

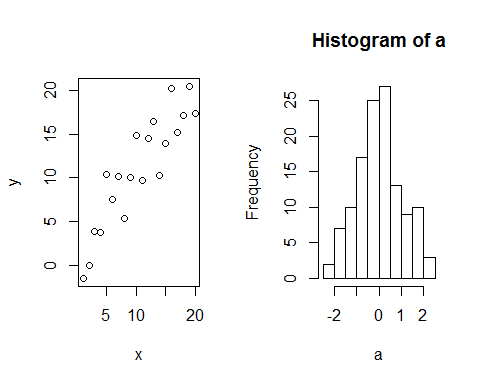
No default, mfrow=c(1,1), ou seja, um gráfico apenas.

Vamos brincar um pouco:

par(mfrow=c(2,1)) #Duas linhas e uma coluna, ou seja, um gráfico em cima do outro.  
plot(x,y)  
hist(a)



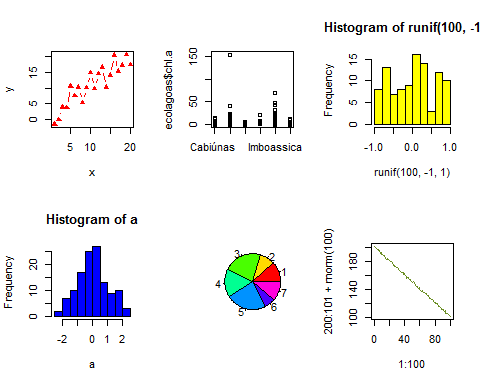
par(mfrow=c(1,2)) #Uma linha e duas colunas, ou seja, um gráfico ao lado do outro.  
plot(x,y)  
hist(a)



Quando o espaço dos gráficos está dividido, cada novo gráfico criado ocupa um espaço disponível, se houver (Em vez de apagar o anterior)

Vejamos mais um exemplo:

par(mfcol=c(2,3)) #Duas linhas e três colunas  
  
plot(x,y,type="b",pch=17,col="red")  
hist(a,col="blue")  
stripchart(ecolagoas$chl.a ~ ecolagoas$lagoa,v=T)  
pie(c(45,32,85,63,90,23,43),col=rainbow(7))  
hist(runif(100,-1,1),col="yellow")  
plot(1:100,200:101+rnorm(100),type="l",col="olivedrab")



No exemplo acima, vemos que os gráficos são plotados pelas colunas; quando uma coluna enche, passa-se para a próxima.

par(mfrow=c(1,1))