Aula 3.6 - anotações

Elder Sodre

Wednesday, February 25, 2015

Os gráficos do R no pacote básico usam funções diferentes para criar gráficos e para anotar gráficos. Anotar significa adicionar elementos a um gráfico pronto. Esses elementos podem ser textos, linhas, pontos, legendas...

#### Preparando os dados

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

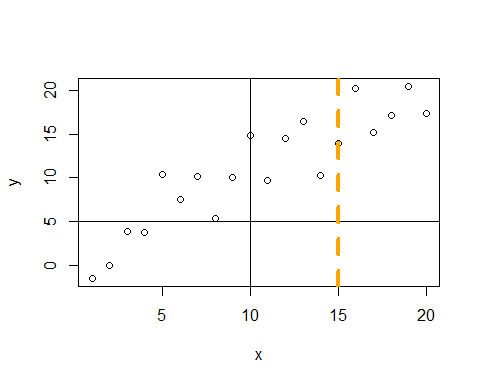
setwd("C:\\Users\\Elder\\Desktop\\Minicurso Elementos de Linguagem R")  
  
#Dados do ecolagoas de 2008:  
ecolagoas <- read.csv2("ecolagoas2008.csv",na.strings="-",row.names=1)  
  
#Dados de zooplâncton do lago Batata em 1998:  
batata <- read.csv2("zoobatata1998.csv",row.names=1)  
zoo <- batata[,3:53] #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)

#### Adicionando linhas

A função abline() acresenta uma linha reta a um gráfico existente. Quatro importantes argumentos desta função são: a,b,h,v.

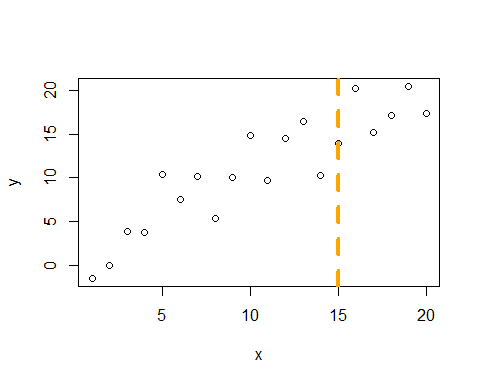
h & v acrescentam uma linha horizontal ou vertical, respectivamente; basta especificar o ponto em que a linha atravessará o eixo x (linha vertical) ou o eixo y (linha horizontal).

plot(x,y)  
abline(h=5) #Acrescentei uma linha horizontal  
abline(v=10) #Acrescentei uma linha vertical  
abline(v=15,col="orange",lty=2,lwd=4) #Outra linha vertical



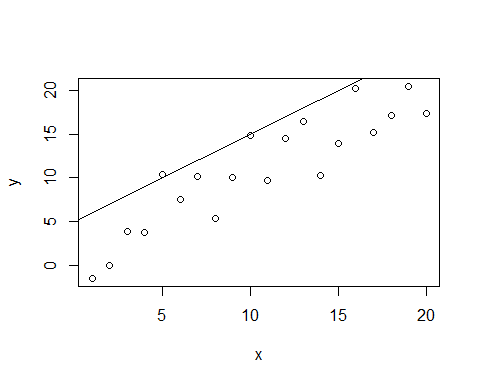
Lembre-se que é possível acrescentar novos elementos a um gráfico, mas não é possível apagar nada de um gráfico. Por exemplo, se no gráfico acima eu quiser manter apenas a linha laranja, eu preciso criar um gráfico novo.

plot(x,y)  
abline(v=15,col="orange",lty=2,lwd=4)

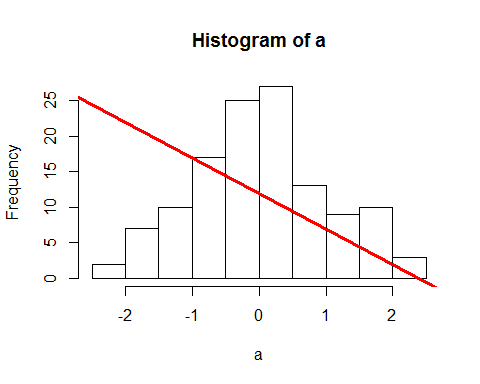


Os argumentos a e b da função abline() acrescentam uma linha reta inclinada ao gráfico, especificando o intercepto e a inclinação da reta. Lembre=se: a equivale ao intercepto da reta (lugar onde ela encontra o eixo y) e b equivale à inclinação (tangente do ângulo formado com o eixo x) (equação da reta: y=a + bx).

plot(x,y)  
abline(a=5,b=1)

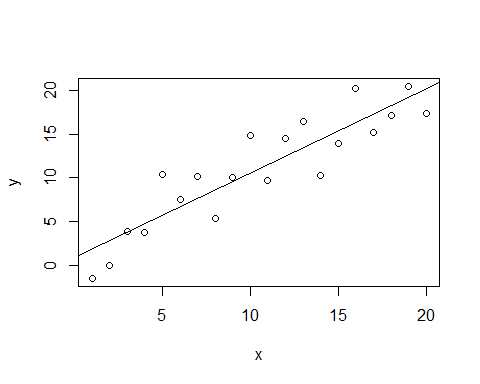


hist(a)  
abline(a=12,b=-5,col="red",lwd=3)



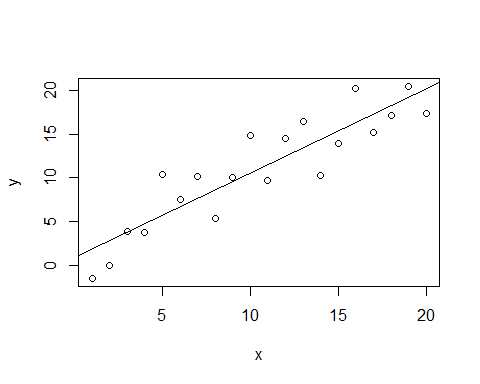
Para adicionar uma linha de tendência ao gráfico de dispersão, basta utilizar a função lm(), que cria um modelo linear. Use a seguinte fórmula:

plot(x,y)  
abline(lm(y~x))



Ou, se preferir, crie um objeto com os resultados de lm() e use este objeto na função abline().

plot(x,y)  
fit <- lm(y~x)  
abline(fit)



#### Acrescentando mais dados ao gráfico

Para exemplificar a adição de novos dados a um gráfico, usaremos os dados de zooplâncton no lago Batata em 1998.

Dados de zooplâncton do lago Batata em 1998:

setwd("C:\\Users\\Elder\\Desktop\\Minicurso Elementos de Linguagem R")  
  
batata <- read.csv2("zoobatata1998.csv",row.names=1)  
zoo <- batata[,3:53] #apenas as colunas com as densidades

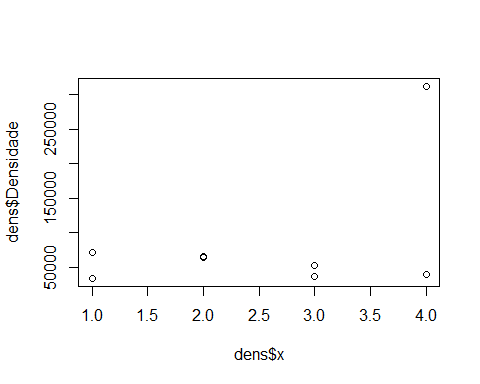
rowSums(zoo) #Densidades

## E08 Enchente E08 Águas Altas E08 Vazante E08 Águas Baixas   
## 72089 65736 36435 40258   
## E10 Enchente E10 Águas Altas E10 Vazante E10 Águas Baixas   
## 34399 64460 53226 311525

dens <- cbind(batata[,1:2],Densidade=rowSums(zoo)) #Dataframe com as densidades totais em cada coleta.  
#O dataframe acima não contém nenhum valor de x para plotar. Vamos criar.  
dens <- cbind(dens,x=rep(1:4,2))

Vamos plotar o gráfico:

plot(dens$Densidade~dens$x)



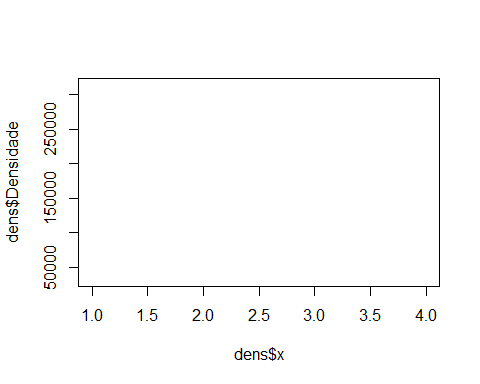
Observe que, nestes dados, a estação de coleta é importante, mas plotar todos os pontos juntos não permite diferenciar entre ambas.

Há duas funções para acrescentar mais dados: points() e lines(). As duas fazem exatamente a mesma coisa, com a diferença que, por default, em points() type="p". Em lines(), type="l".

points() e lines() são semelhantes à função plot(). Porém, plot() cria um gráfico novo e acrescenta os dados, enquanto points() e lines() apenas acrescentam dados a um gráfico preexistente.

Voltando ao caso do lago Batata, uma solução para o nosso problema seria plotar o gráfico vazio e depois acresentar os pontos.

plot(dens$Densidade~dens$x,type="n")



Agora eu preciso separar os dados das duas estações de coleta. Por exemplo, com a função split:

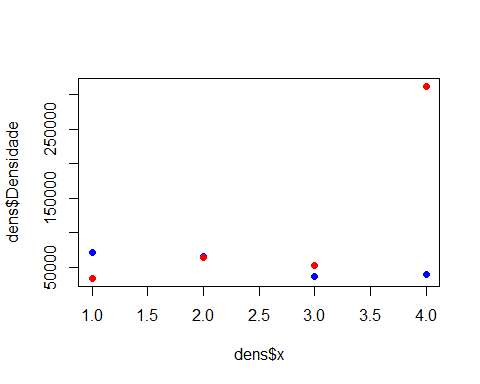
split(dens,f=dens$Estação)

## $E08  
## Estação Período Densidade x  
## E08 Enchente E08 Enchente 72089 1  
## E08 Águas Altas E08 Águas Altas 65736 2  
## E08 Vazante E08 Vazante 36435 3  
## E08 Águas Baixas E08 Águas Baixas 40258 4  
##   
## $E10  
## Estação Período Densidade x  
## E10 Enchente E10 Enchente 34399 1  
## E10 Águas Altas E10 Águas Altas 64460 2  
## E10 Vazante E10 Vazante 53226 3  
## E10 Águas Baixas E10 Águas Baixas 311525 4

d08 <- split(dens,f=dens$Estação)$E08  
d10 <- split(dens,f=dens$Estação)$E10

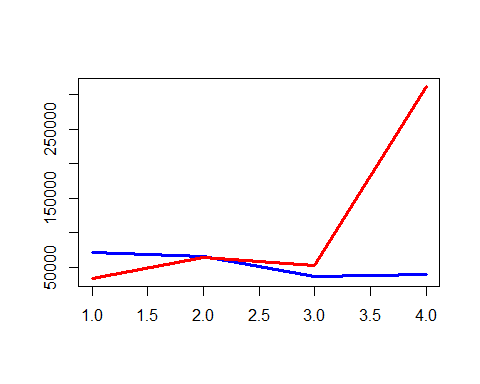
Vamos voltar ao nosso gráfico:

plot(dens$Densidade~dens$x,type="n")  
points(d08$Densidade~d08$x,pch=16,col="blue")  
points(d10$Densidade~d10$x,pch=16,col="red")



Pronto! Agora, como demonstração, vamos fazer a mesma coisa usando a função lines()

plot(dens$Densidade~dens$x,type="n",xlab="",ylab="")  
lines(d08$Densidade~d08$x,lwd=3,col="blue")  
lines(d10$Densidade~d10$x,lwd=3,col="red")



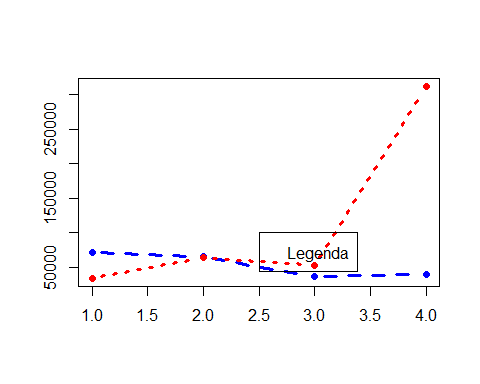
#### Acrescentando legendas

Ainda faltam acrescentar legandas ao gráfico acima, certo?

Isto é feito com a função legend()

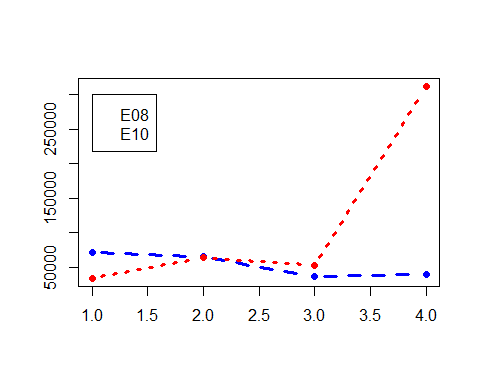
Três argumentos desta função são x, y e legend. x e y especificam as coordenadas da legenda e legend especifica o texto.

plot(dens$Densidade~dens$x,type="n",xlab="",ylab="")  
lines(d08$Densidade~d08$x,type="b",pch=16,lty=2,lwd=3,col="blue")  
lines(d10$Densidade~d10$x,type="b",pch=16,lty=3,lwd=3,col="red")  
  
 legend(x=2.5,y=100000,legend="Legenda")



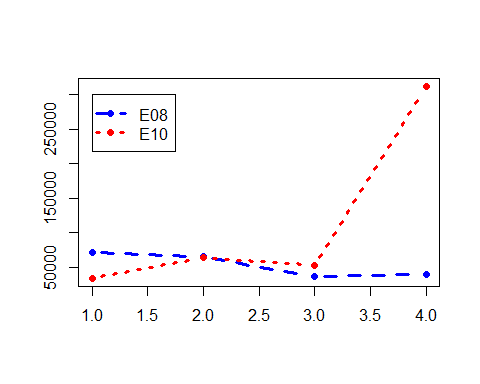
A legenda acima serviu como demonstração, mas ainda não é muito útil.

plot(dens$Densidade~dens$x,type="n",xlab="",ylab="")  
lines(d08$Densidade~d08$x,type="b",pch=16,lty=2,lwd=3,col="blue")  
lines(d10$Densidade~d10$x,type="b",pch=16,lty=3,lwd=3,col="red")  
  
legend(x=1,y=300000,legend=c("E08","E10")) #Agora melhorou, mas falta dizer quem é quem.



O R não calcula automaticamente os parâmentros (pontos, linhas, cores) da legenda. É necessário dizê-lo novamente ao programa.

plot(dens$Densidade~dens$x,type="n",xlab="",ylab="")  
lines(d08$Densidade~d08$x,type="b",pch=16,lty=2,lwd=3,col="blue")  
lines(d10$Densidade~d10$x,type="b",pch=16,lty=3,lwd=3,col="red")  
legend(x=1,y=300000,legend=c("E08","E10"),pch=16,lty=c(2,3),lwd=3,col=c("blue","red"))



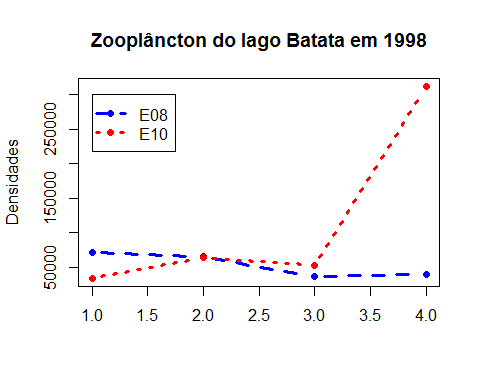
Há ainda outras formas de customizar a legenda. Para maiores informações, confira em ?legend .

#### Acrescentando títulos

Também é possível acrescentar títulos dos gráficos e eixos, com a função title().

A função title() usa os argumentos main, sub, xlab e ylab, que já sabemos como funciona.

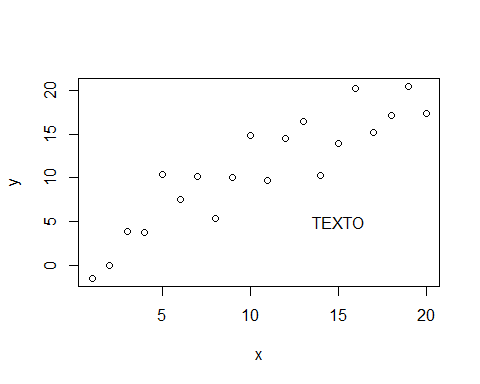
plot(dens$Densidade~dens$x,type="n",xlab="",ylab="")  
lines(d08$Densidade~d08$x,type="b",pch=16,lty=2,lwd=3,col="blue")  
lines(d10$Densidade~d10$x,type="b",pch=16,lty=3,lwd=3,col="red")  
legend(x=1,y=300000,legend=c("E08","E10"),pch=16,lty=c(2,3),lwd=3,col=c("blue","red"))  
  
title(main="Zooplâncton do lago Batata em 1998",ylab="Densidades")



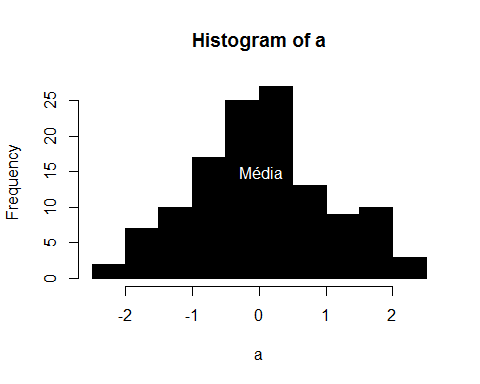
#### Acrescentar texto

A função text() acrescenta texto a um gráfico. Os argumentos x e y equivalem às coordenadas, enquanto label é o texto propriamente dito.

plot(x,y)  
text(x=15,y=5,label="TEXTO")



hist(a,col="black")  
text(x=mean(a),y=15,"Média",col="white")



Mais de um texto pode ser incluído ao mesmo tempo.

plot(dens$Densidade~dens$x,type="n",xlab="",ylab="")  
lines(d08$Densidade~d08$x,type="b",pch=16,lty=2,lwd=3,col="blue")  
lines(d10$Densidade~d10$x,type="b",pch=16,lty=3,lwd=3,col="red")  
legend(x=1,y=300000,legend=c("E08","E10"),pch=16,lty=c(2,3),lwd=3,col=c("blue","red"))  
title(main="Zooplâncton do lago Batata em 1998",ylab="Densidades")  
  
text(x=1:4,y=100000,label=c("enchente","águas altas","vazante","águas baixas"))

