

## Table of Contents

<b>Função plot()</b> .....	1
<b>Dispositivos de Tela</b> .....	1
<b>Listar e Controlar Dispositivos</b> .....	2
<b>Dispositivos de Arquivos</b> .....	3
Figuras Vetoriais PDF ou POSTSCRIPT .....	3
Figuras Raster .....	4
Video aula .....	5



# Dispositivos Gráficos

No existe existem dois principais tipos de **dispositivos (Devices) gráficos**, que basicamente significam **onde** você irá imprimir um gráfico ou figura, podendo ser:

1. Na tela (monitor) - ou seja, em janelas do R ou do RStudio onde você visualiza gráficos;
2. Num arquivo em formato pdf, jpeg, tiff, png, eps, etc.

## Função plot()


- **plot()** ou **plot.default()** - é a principal função genérica para gerar gráficos. Iremos ver isso melhor em [Funções de Alto Nível](#), mas é necessário rapidamente introduzir isso para facilitar as demais explicações.
- Aqui entenda que com essa função você gera gráficos e vamos apenas ver um exemplo muito simples

```
#o objeto R do iris como exemplo
str(iris)
#plotando comprimento de sépala vs. comprimento de pétalas e colorindo os
pontos de acordo com as espécies
vcl = c("yellow", 'green', 'blue')[as.numeric(iris$Species)] #cria um vetor de
cores para cada linha em Iris segundo especie
#plota a figura
plot(iris$Sepal.Length, iris$Petal.Length, type='p', pch=21, bg=vcl, cex=0.8)
```

- Muitos argumentos da função plot() são [parâmetros gráficos](#).

## Dispositivos de Tela

Quando você usa funções para gerar um gráfico, o R ou RStudio usa automaticamente um dispositivo de tela. No entanto, você pode abrir novas janelas com as seguintes funções:

- **X11()** ou **x11()** - funciona no Mac, Linux e acho que Windows
- **quartz()** - apenas se estiver usando MacOSX
- **windows()** - apenas se estiver usando Windows (esta função continua disponível na versão Windows do R? )

```
?device #veja o help da função e as opções de devices
X11() #irá abrir uma janela nova
#vamos plotar o mesmo gráfico do exercício anterior
vcl = c("yellow", 'green', 'blue')[as.numeric(iris$Species)] #cria um vetor de
cores para cada linha em Iris segundo especie
#plota a figura
plot(iris$Sepal.Length, iris$Petal.Length, type='p', pch=21, bg=vcl, cex=0.8)

#irá fechar essa nova janela
dev.off()
```

```
quartz() #num mac abre uma janela sem dar nenhum aviso (funciona melhor que
X11() no Mac)
vcl = c("yellow", 'green', 'blue')[as.numeric(iris$Species)] #cria um vetor de
cores para cada linha em Iris segundo especie
#plota a figura
plot(iris$Sepal.Length, iris$Petal.Length, type='p', pch=21, bg=vcl, cex=0.8)

dev.off() #fecha
```

## Listar e Controlar Dispositivos

Existem funções que permitem trabalhar com vários dispositivos ao mesmo tempo, controlando o uso dos dispositivos abertos. Pode ter, por exemplo, várias janelas diferentes mostrando gráficos diferentes durante um trabalho. As principais funções para trabalhar com dispositivos são:

- **dev.list()** - lista todos os dispositivos abertos no momento (geralmente o dispositivo padrão, que é sempre o número 1, é ignorado na lista), retornando o número (ordem de abertura) e o nome de cada um
- **dev.cur()** - mostra qual o dispositivo que está ativo (current). Se há vários abertos, tem um que está ativo no momento e se você enviar uma figura vai sair neste dispositivo.
- **dev.set()** - torna ativo um determinado dispositivo
- **dev.off()** - fechar o dispositivo atual ou vários. Esta é a função dessa lista que é mais usada na prática.

```
#vamos primeiro fechar todos os dispositivos
dev.off(which=dev.list())
dev.list() #vai retornar NULL por não há dispositivos abertos
#vamos abrir vários dispositivos
X11() #primeira janela extra
X11() #segunda janela e mesma figura com cores diferentes
X11() #terceira janela e mesma figura com cores diferentes

#coloque e redimensione as janelas para ter as tres visiveis na tua tela e
volte aqui.

#veja os dispositivos abertos
dev.list()

#qual o atual
dev.cur() #o ultimo que abrimos, né

dev.set(2)
#vamos mudar para o segundo e plotar algo
plot(iris$Sepal.Length, iris$Petal.Length, type='p', cex=0.8)

#vamos mudar para o terceiro e plotar outra coisa
dev.set(3)
dev.cur() #deve responder 3
```

```

vcl = c("yellow", 'green', 'blue')[as.numeric(iris$Species)]
plot(iris$Sepal.Length, iris$Petal.Length, type='p', pch=21, bg=vcl, cex=0.8)

#vamos mudar para o quarto e plotar a mesma coisa com outras cores
dev.set(4)
dev.cur() #deve responder 4
vcl = c("red", 'blue', 'white')[as.numeric(iris$Species)]
plot(iris$Sepal.Length, iris$Petal.Length, type='p', pch=21, bg=vcl, cex=0.8)

#vamos fechar o dispositivo atual
dev.cur()
dev.off()
dev.cur() #mudou automaticamente porque voce fechou o 4
dev.off() #mesma coisa fechou o ativo
dev.cur() #sobrou o 3
#fechar este também
dev.off()
dev.list() #nao tem mais nenhum aberto

```

## Dispositivos de Arquivos

Há vários dispositivos para gerar arquivos com imagens. As funções em geral tem o nome do tipo de arquivo gerado. Vamos ver dois exemplos apenas, mas a mesma lógica se aplica a qualquer um dos dispositivos listados em dispositivos (**help('device')**).

- Essas funções são ótimas ferramentas para incluir nos seus scripts a produção de figuras para uma publicação, basta ver na revista as especificações e através dos parâmetros das funções que abrem os dispositivos e dos **parâmetros gráficos**, você pode gerar figuras na especificação exata da revista selecionada.

## Figuras Vetoriais PDF ou POSTSCRIPT

- **pdf()** e **postscript()** - são na nossa opinião as funções mais importantes, porque elas geram arquivos de imagens de ótima qualidade porque são vetorias. Não precisa definir resolução, porque nesse tipo de imagem isso não existe.

```

#plotando a figura na tela é o que fazemos normalmente
vcl = c("yellow", 'green', 'blue')[as.numeric(iris$Species)]
plot(iris$Sepal.Length, iris$Petal.Length, type='p', pch=21, bg=vcl, cex=0.8)

#salvando a figura num PDF
?pdf #veja o help dessa função e seus argumentos, especialmente width e height
#abre o dispositivo para pdf
pdf(file='meupdf.pdf', width=lcm(15), height=lcm(10))
#lcm() apenas pega valores em cm e converte em polegadas que é a especificação padrão da funcao pdf. Portanto largura e altura de cada página do PDF é definida por esses argumentos

```

```
#plota a figura
plot(iris$Sepal.Length,iris$Petal.Length,type='p',pch=21,bg=vcl,cex=0.8)
dev.off() #fecha o pdf
#só será possível ver o pdf se tiver fechado ele.
getwd() #o pdf foi gerado nessa pasta

#se voce incluir vários gráficos. O pdf irá gerar várias páginas ao mesmo
tempo.
#abre o dispositivo para pdf
pdf(file='meupdf2.pdf',paper='a4')
#tamanho papel A4
#plota a figura 10 vezes
for(i in 1:10) {
  plot(iris$Sepal.Length,iris$Petal.Length,type='p',pch=21,bg=vcl,cex=0.8)
}
dev.off() #fecha o pdf
#veja que o arquivo 2 tem várias páginas
```

## Figuras Raster

- **jpeg(), png(), tiff()**,etc. - dispositivos que geram imagens em pixels, cuja qualidade depende muito da definição da resolução.

```
#formato jpeg sem controlar a resolucao (usando units='px' ou pixel)
#abre o dispositivo no formato desejado
jpeg(filename="meujpeg.jpg",width=600,height=400,units='px')

#plota alguma coisa
vcl = c("yellow",'green','blue')[as.numeric(iris$Species)]
plot(iris$Sepal.Length,iris$Petal.Length,type='p',pch=21,bg=vcl,cex=0.8)

#fecha o jpeg
dev.off()

#ABRA O ARQUIVO E FAÇA UM ZOOM GRANDE
#note que os pixels são super visíveis

#AGORA MELHORANDO A RESOLUCAO
#formato jpeg com 300dpi
#abre o dispositivo no formato desejado
jpeg(filename="meujpeg2.jpg",width=15,height=10,units='cm',res=300)

#plota alguma coisa
vcl = c("yellow",'green','blue')[as.numeric(iris$Species)]
plot(iris$Sepal.Length,iris$Petal.Length,type='p',pch=21,bg=vcl,cex=0.8)

#fecha o jpeg
dev.off()
```

- **ABRA OS TRES ARQUIVOS E COMPARE A RESOLUÇÃO FAZENDO UM ZOOM ATÉ VISUALIZAR O PIXEL. NOTE QUE NO PDF VOCE NUNCA VE O PIXEL, POR ISSO TRABALHE COM pdf().**

## Video aula

[video01\\_bot89-2020-04-09\\_18.22.40.mp4](#)

From:

<http://www.botanicaamazonica.wiki.br/labotam/> - **Ecologia e Evolução de Plantas Amazônicas**

Permanent link:

<http://www.botanicaamazonica.wiki.br/labotam/doku.php?id=bot89:precurso:1plot:inicio>

Last update: **09/43/2020 19:43**