

- [Tutorial](#)
- [Exercícios](#)
- [Apostila](#)

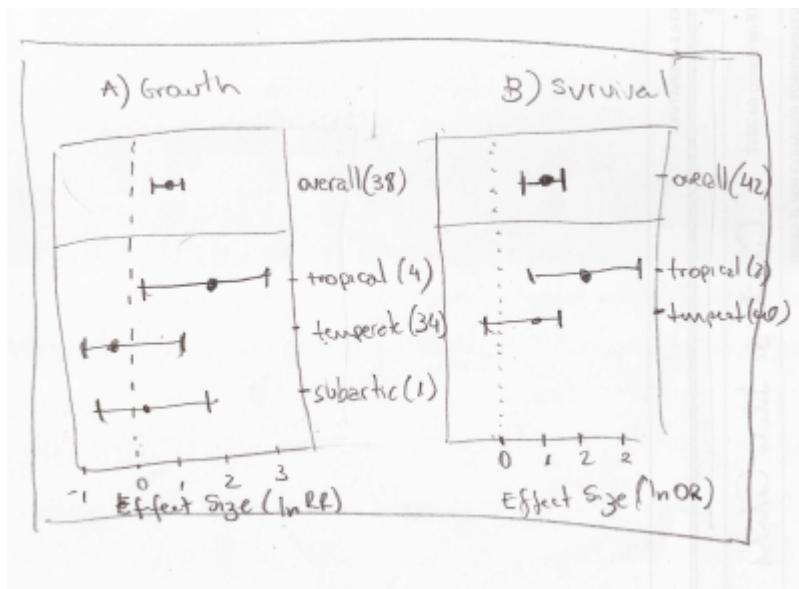
5b. Gráficos II: um procedimento



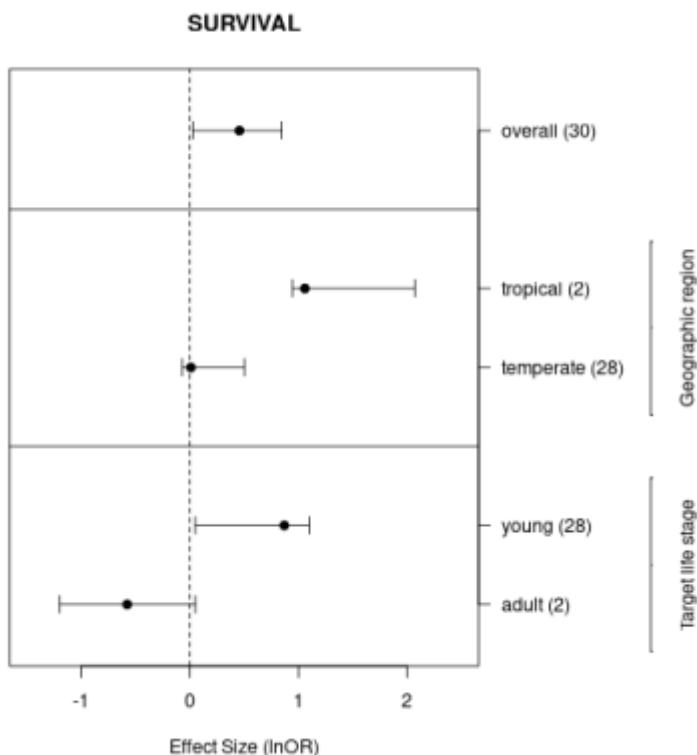
[Video](#)

Nesse tutorial apresentamos um procedimento para a construção de gráficos no R, incluindo cada elemento separadamente. O objetivo é mostrar como é possível editar e incluir elementos gráficos da forma que desejar, o que permite a construção de gráficos muito complexos. Utilizaremos o gráfico publicado por uma de nossas primeiras alunas para exemplificar o procedimento adotado para produzir gráficos no R.

O procedimento tem início com um esboço do gráfico. Nesse caso, fizemos um esboço no guardanapo de papel em um boteco. Essa é a parte mais complicada, imaginar algo que represente os nossos dados de forma sintética, de fácil leitura, esteticamente agradável e adequado à revista em que será publicado.



Para simplificar nosso procedimento neste tutorial, vamos produzir apenas uma variação de um dos painéis. Ou seja, queremos algo como:



Apesar de ser um gráfico simples, apresenta algumas complicações inerentes ao tipo de gráfico que não é padrão.

Abrindo o dispositivo

Quando o dispositivo de tela é aberto pelas funções de alto nível, os parâmetros que serão abertos são aqueles definidos pelo valores que estão no par. Um parâmetro importante é a dimensão do dispositivo. Dependendo do tamanho e da razão entre altura e largura, os elementos podem ter sua posição relativa muito diferente. Inclusive, sobrepondo elementos. Para evitar esse tipo de complicaçāo é importante que a janela gráfica de tela tenha as mesmas dimensões e razão da figura final que deverá ser construída em um dispositivo de arquivo. Quando possível é interessante também trabalhar com a mesma resolução da imagem, associada ao tamanho da janela e o tamanho do pixel. Entretanto, esse último é mais complicado de ser definido no dispositivo de tela.

O tamanho padrão do dispositivo de tela é 7 polegadas de altura e de largura. Para controlar essas dimensões abrimos o dispositivo de tela com os argumentos `width` e `height`, como a seguir:

```
X11(width = 10, height = 10)
```

Quando submeter os exercícios no notaR, **não abra o dispositivo gráfico!** Deixe as funções de alto nível abrirem o dispositivo com o tamanho padrão.

Criando o layout

Como os elementos da legenda do eixo à direita são complexos, vamos tratá-los como um painel à parte. No [primeiro tutorial de gráficos](#) utilizamos o parâmetro `mfrow` para dividir o dispositivo gráfico em partes simétricas. A função `layout` é mais flexível e permite criar painéis de diferentes tamanhos no dispositivo gráfico. No nosso caso, vamos criar duas colunas, a esquerda com 80% da largura total e a direita com 20% restante. O primeiro argumento da função é uma matriz com a sequência com que os painéis irão ser desenhados ¹⁾.

```
layout(matrix(c(1, 2), ncol = 2, nrow = 1), width = c(8, 2))
layout.show(2) # mostra o layout dos dois painéis
```

Iniciando o gráfico

A primeira coisa a fazer depois de definir o layout do gráfico é ajustar os parâmetros gráficos globais do primeiro painel. Em seguida, construímos o espaço de coordenadas sem nenhum elemento utilizando uma função de alto nível, como o `plot`. Note que o espaço de coordenadas deve estar relacionado às amplitudes dos dados que serão grafados.

```
par (mar = c(5, 1, 4, 5))
plot(x = NULL, y = NULL, xlim = c(-1.5, 2.5), ylim = c(0.5, 7.5), type =
  "n", yaxt = "n", xlab = "Effect Size (lnOR)", ylab = "", main = "SURVIVAL")
```

Linhas guias e eixo

Em seguida, continuamos inserindo elementos. Abaixo utilizamos a função `abline`, a função que desenha linhas de regressão ($y = a + bx$) utilizando os parâmetros `v` para linha vertical e `h` para linha horizontal. O parâmetro `lty` define o tipo de linha, no caso 2 é a linha tracejada. Outras funções também fazem essa tarefa, como por exemplo `segments` que usaremos mais a frente.

```
abline (v = 0, lty = 2)
abline (h = c(3,6))
axis(side = 4, at = c(1,2,4,5,7), labels=c("adult (2)", "young (28)",
  "temperate (28)", "tropical (2)", "overall (30)"), las=2 )
```

Utilizamos no código acima a função `axis` para construir o eixo e seus elementos. O lado do eixo é definido pela posição iniciando pelo eixo x como 1 e seguindo no sentido horário. A posição 4 indica o eixo à direita.

Inserindo os dados

A lógica desse método é incluir cada elemento separadamente para ter controle total na elaboração do gráfico. Abaixo inserimos os resultados para os adultos:

```
# ADULT
```

```
points(x = -0.577, y = 1, pch = 19) # pch: tipo de simbolo
points(x = -1.2, y = 1, pch = "|")
points(x = 0.05, y = 1, pch = "|")
segments(x0 = -1.2, y0 = 1, x1 = 0.05, y1 = 1) # um segmento
```

Agora os resultados para os outros grupos:

```
#YOUNG
points(x = 0.87, y = 2, pch = 19)
points(x = -0.05, y = 2, pch = "|")
points(x = 1.1, y = 2, pch = "|")
segments(x = 1.1, y0 = 2, x1 = -0.05, y1 = 2)
#TEMPERATE
points(x = 0.01, y = 4, pch = 19)
points(x = -0.07, y = 4, pch = "|")
points(x = 0.5, y = 4, pch = "|")
segments(x = -0.07, y0 = 4, x1 = 0.5, y1 = 4)
#TROPICAL
points(x = 1.06, y = 5, pch = 19)
points(x = 0.946, y = 5, pch = "|")
points(x = 2.073, y = 5, pch = "|")
segments(x = 2.073, y0 = 5, x1 = 0.946, y1 = 5)
#OVERALL
points(x = 0.457, y = 7, pch = 19)
points(x = 0.025, y = 7, pch = "|")
points(x = 0.847, y = 7, pch = "|")
segments(x = 0.025, y0 = 7, x1 = 0.847, y1 = 7)
```

Segundo painel

Esse painel foi criado para acrescentar a legenda da direita com suas particularidades. As margens do gráfico não permitem edições complexas. Caso o gráfico tenha elementos nas margens, uma solução é tratar a margem como um painel. Como fizemos no painel anterior, primeiro ajustamos os parâmetros globais e, em seguida, criamos um espaço de coordenadas cartesianas para posicionar os elementos em um espaço vazio.

```
par (mar=c(5,2,4,3)) #controla tamanhos das margens
plot(x=NULL,y=NULL, xlim=c(0, 2), ylim=c(0.5, 7.5), type="n", xaxt="n",
yaxt="n", xlab="", ylab="", bty="n")
```

Em seguida, acrescentamos os elementos e legendas de eixos. Entretanto, diferente do que fizemos no painel anterior, incluindo cada elemento isoladamente, podemos colocar elementos de mesmo tipo juntos, tirando proveito da operação vetorizada das funções e tornando o código mais sintético.

```
points(x=rep(c(0.5),4), y=c(0.4, 2.6, 3.4, 5.6), pch="-")
segments(x0=c(0.5, 0.5), y0=c(0.4, 3.4), x1=c(0.5, 0.5), y1=c(2.6, 5.6))
axis(side=4, at=1.5, labels= "Target life stage", lwd.ticks=0, cex.axis =
1.5)
axis(side=4, at=4.5, labels= "Geographic region", lwd.ticks=0, cex.axis =
```

1.5)

Salvando o gráfico

Para salvar o gráfico em um arquivo, pode-se utilizar a seguinte função.

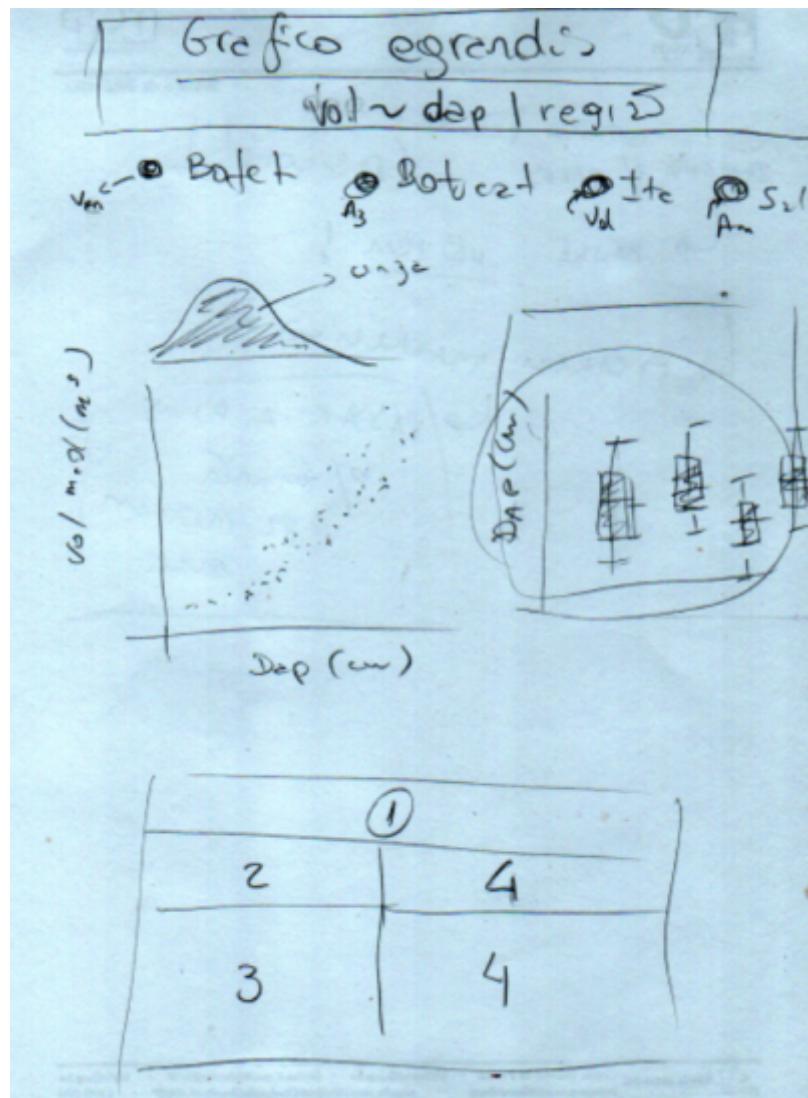
`savePlot`

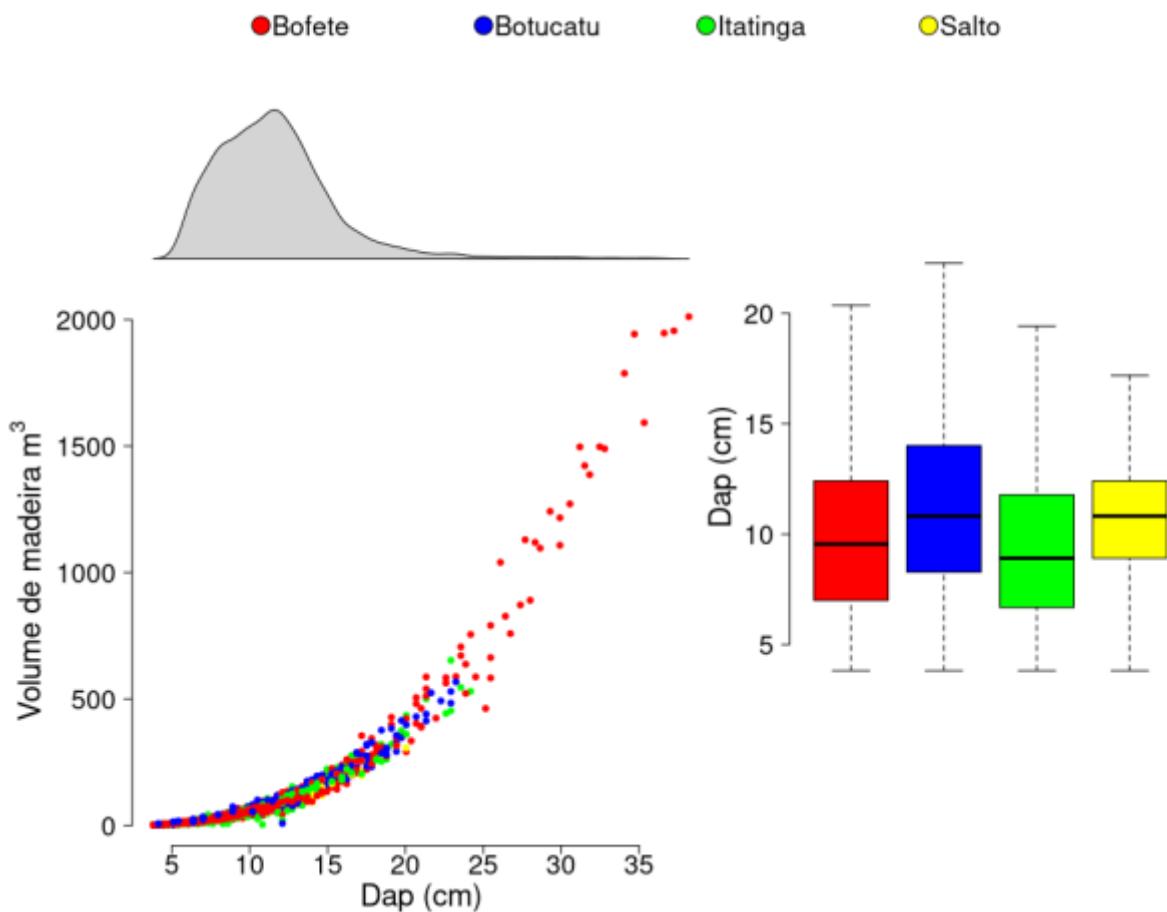
Caso esteja usando o RStudio, antes, inicie uma janela gráfica com a função `X11()`²⁾. Para ter mais controle da qualidade gráfica é necessário usar um dispositivo de arquivo (funções `tiff`, `jpeg`, `png`, por exemplo).

```
savePlot("metaGraf.png", type = "png")
```

Gráfico diâmetro e volume de eucaliptos

Agora vamos reproduzir o código apresentado em aula que cria o gráfico com os dados de *Eucaliptus grandis*. O planejamento do gráfico foi feito a partir do esboço e ele deve ficar como a figura que segue.





Siga as instruções para chegar ao gráfico como na figura acima:

1. Leia o arquivo [egrandis.csv](#) a partir do seu diretório de trabalho em uma sessão do R, nomeie o objeto como `egrandis`.
2. Garanta que as variáveis foram lidas corretamente e crie o objeto `tabReg` com a contagem de quantas observações (`arvore`) tem em cada uma das `regiao`.
3. Crie um objeto chamado `cores` com as quatro cores (vermelho, azul, verde e amarelo) que representarão as localidades. Essas cores devem ser criadas com a função `rgb` sem transparência e com valores máximos das cores.
4. Abra um dispositivo de janela com largura de 10 e altura de 8 polegadas. **Essa linha de comando precisa ficar comentada antes de submeter ao notaR, ele não aceita abertura de dispositivos de tela. Mas, é importante manter o tamanho para comparar com a figura enquanto produz o gráfico.** Não modifique o tamanho da janela ao longo do processo, assim as proporções não se alteram.
5. Mude os parâmetros globais do dispositivo gráfico para ter: aumento de 20% nos símbolos desenhados no gráfico e sem a borda que define a área gráfica.
6. Crie uma estrutura de layout que tenha 4 painéis da seguinte forma: `matrix(c(1, 2, 3, 1, 4, 4), 2, 2)` como no esboço da figura.
 - proporções de altura `c(0.13, 0.20, 0.67)`
 - proporções de largura `c(0.60, 0.40)`
7. Comece colocando a legenda de localidades no painel superior que tem 13% de altura e toda a largura da figura. Modifique o parâmetro global do painel para que fique sem margens `mar = c(0, 0, 0, 0)` e, desta forma, usar toda a área de plotagem.
8. Nessa primeiro painel inicie o gráfico com a função `plot` sem nenhum elemento e com as coordenadas cartesianas definidas com o eixo x e o eixo y indo de 0 até 1.
9. Coloque os nomes das localidades em ordem alfabética nas posições em `x c(0.20, 0.40,`

1. Poste o código e a figura do gráfico *Eucaliptus grandis* no formulário google

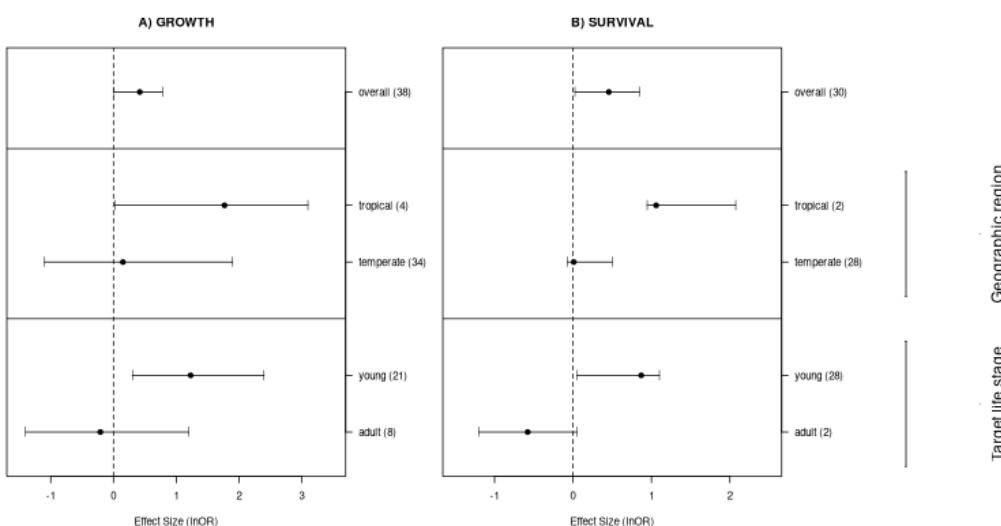
Last update: 2025/09/23 19:34

02_tutoriais:tutorial5b:start http://ecor.ib.usp.br/doku.php?id=02_tutoriais:tutorial5b:start

0.60, 0.80) e bem no meio da altura ($y = 0.5$), garanta que a posição do texto em relação às 2. A continuação do nosso tutorial está no exercício do notar [Finalizando o gráfico](#) coordenadas é ajustado à esquerda e no meio da altura".

10. Coloque os símbolos de círculo, preenchido pela cor correspondente de cada localidade, usando

9, 0.59, 0.79)



do a função

nites do x e do y

intermediário
ermelho, verde e

L).

nites de máximo e
o egrandis,

17. Insira os eixos com as respectivas escalas nos eixos x e y, com a função `axis` com os

3. Em seguida o desenho aparece aumentando em 100% o tamanho dos dígitos das escalas.

18. Coloque a legenda do eixo x e y como aparece na figura, com a distância da legenda para o

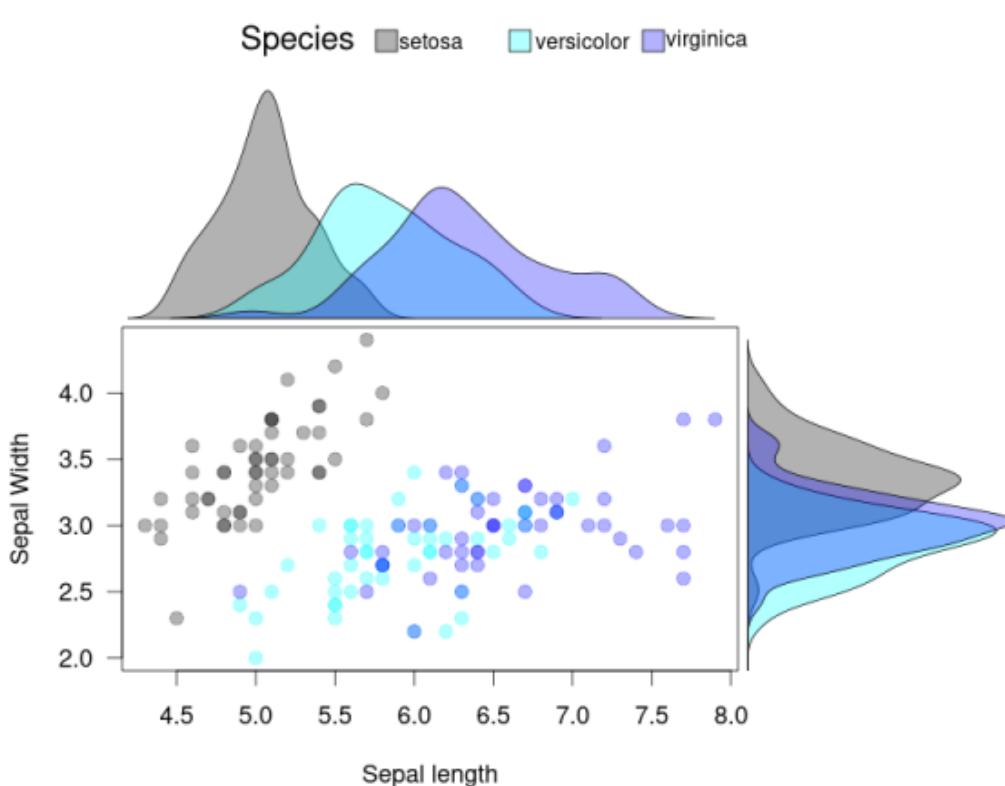
50%.⁴⁾

ágio, use `pch=19`.
`c(15, 4, 10,`

ocalidades, lembra-

imentando em

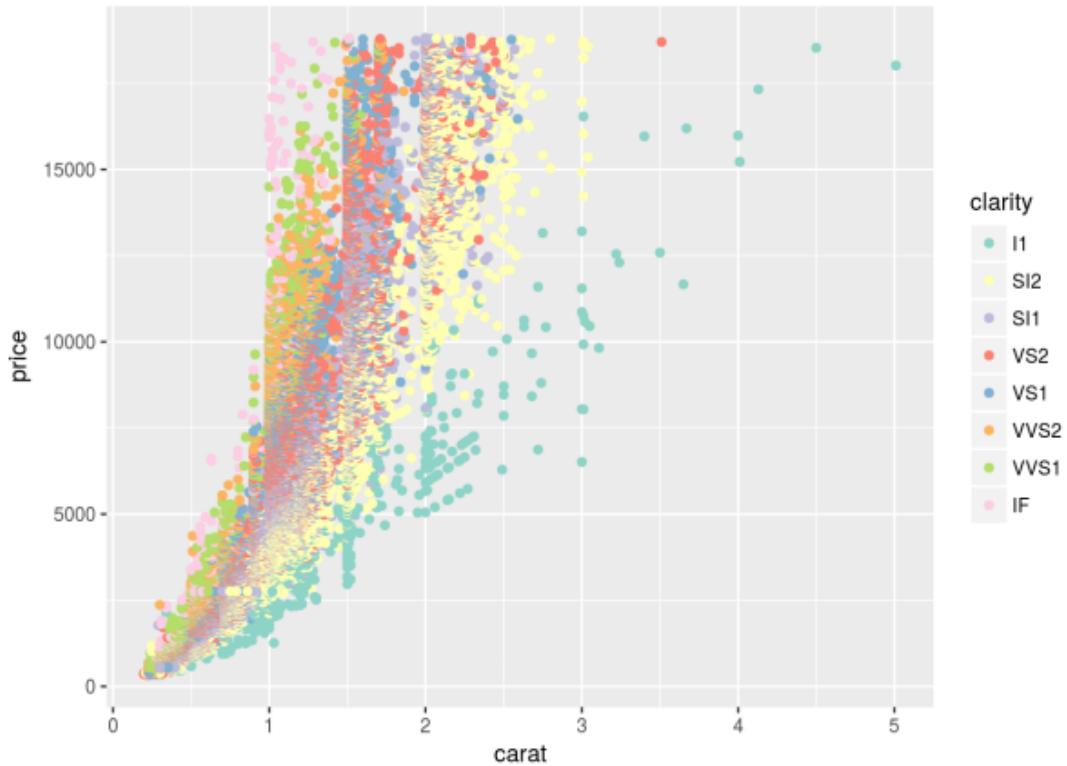
eres e a 3.5 linhas



4. Nesse wiki focamos no uso das ferramentas básicas do R e nesse tutorial no pacote `graphics` carregado por padrão na sessão do R. O pacote para elaboração de gráficos chamado `ggplot2` vem se tornando muito popular nos últimos anos, mas apresenta uma sintaxe muito diferente da usual no R, o que nos parece não ser muito efetivo para o aprendizado da linguagem. Por essa razão, preferimos deixá-lo de fora do nosso material.

Existem muitos bons tutoriais sobre o `ggplot2`, inclusive um ótimo feito pelo colaborador da disciplina Gustavo Burin Ferreira, caso tenha interesse acesse:

- <https://blog.gburin.com/tutorial-de-ggplot2>



- 1) Para juntar painéis em um único elemento basta indicar o mesmo valor na matriz
- 2) no windows pode usar a função windows() e no MacOSx pode precisar usar a função quartz()
- 3)

```
adj = c(0, 0.5)
```
- 4) a função expression pode ajudar com a junção de caracteres e fórmulas matemáticas para textos de legendas em gráficos

From:

<http://ecor.ib.usp.br/> - ecoR

Permanent link:

http://ecor.ib.usp.br/doku.php?id=02_tutoriais:tutorial5b:start

Last update: **2025/09/23 19:34**

