

R Commander: um pouco além dos menus gráficos

Material complementar à apostila
*Introdução ao R Commander - Notas de
Aula*

Felipe Rafael Ribeiro Melo

Departamento de Métodos Quantitativos - UNIRIO

Rio de Janeiro, Brasil

Fevereiro de 2019

Sumário

1	Prefácio	4
2	Linhas de comando	5
2.1	Submetendo linha de comando	7
2.2	Submetendo linha de comando modificada	7
3	Cores disponíveis no R	8
4	Modificando a estrutura de gráficos gerados pelo R Commander	10
4.1	Gráfico de barras simples	10
4.1.1	Mudando a cor das barras: apenas uma cor	11
4.1.2	Mudando a cor das barras: duas ou mais cores	11
4.1.3	Espaçamento entre as barras	12
4.1.4	Editando nomes abaixo das barras	12
4.1.5	Tamanho das fontes	13
4.1.6	Encurtando ou alongando o eixo vertical	14
4.1.7	Gráfico de barras na horizontal	14
4.1.8	Porcentagens dos atributos nas barras	15
4.2	Gráfico de pizza	18
4.2.1	Mudando as cores da fatias	18
4.2.2	Tamanho do gráfico de pizza	19
4.2.3	Tamanho da fonte: nomes dos atributos	19
4.2.4	Editando nomes dos atributos ao lado das fatias	20
4.2.5	Porcentagens em gráficos de pizza	20
4.3	Histograma	22
4.3.1	Modificando cores das barras	23
4.3.2	Listar os valores que delimitam as classes	24
4.3.3	Modificar os valores que delimitam cada classe	25
4.3.4	Histograma de um único grupo do conjunto de dados	26
4.3.5	Polígono de frequências	27
4.4	Boxplot	28
4.4.1	Mudando a cor da caixa e da borda de um boxplot	29

4.4.2	Mudando as cores da caixas e das bordas de boxplots em grupos . .	29
4.4.3	Editando nomes dos atributos em boxplots por grupos	30
4.4.4	Tamanho das fontes	31
4.4.5	Encurtando ou alongando o eixo vertical	32
4.4.6	Caixa do boxplot “amassada” na altura da mediana	32
4.4.7	Valores discrepantes indicados por círculos preenchidos	33
4.4.8	Valores discrepantes identificados pelos respectivos valores da variável	33
4.5	Gráfico de barras múltiplas	35
4.5.1	Posição da legenda	36
4.5.2	Edições na caixa da legenda	38
4.6	Diagrama de dispersão	40
4.6.1	Edição de cores	40
4.6.2	Espessura da reta de regressão	41
4.6.3	Inibição da grade (<i>grid</i>) no fundo do gráfico	42
4.6.4	Edições em diagramas de dispersão por grupos	42
4.6.5	[<i>Bônus</i>] Matriz de correlação em um determinado grupo	44
Referências		45
Anexo I - Lista de cores disponíveis no R		47

1 Prefácio

Em diversas áreas do conhecimento, análises estatísticas em conjuntos de dados tem sido cada vez mais utilizadas, e vários *softwares* específicos cuidam de toda a parte computacional de tais análises. Neste contexto, o *software* R tem ganhando destaque nos últimos anos, sobretudo no ambiente acadêmico, em virtude de suas diversas vantagens, destacando-se, entre elas: ser um *software* gratuito, bastante flexível, apoiado por uma grande equipe de desenvolvedores em todo o mundo e frequentemente atualizado com novos pacotes e funcionalidades. Entretanto, tal *software* não possui uma interface “amigável”: para a obtenção de qualquer resultado, se faz necessária a digitação de linhas de comando, o que pode tornar o seu uso desmotivador. Felizmente, com o advento do pacote *Rcmdr*, é possível executar várias funcionalidades do R, no que tange à análise de conjuntos de dados, por meio de menus, ***sem a necessidade de digitar linhas de comando*** que geram tais funcionalidades.

Este material é um complemento à apostila *Introdução ao R Commander - Notas de Aula*, em particular à Seção 4, que trata de representações gráficas. Apesar dos menus de construção de gráficos no R Commander permitirem editar parcialmente o gráfico de interesse na janela de construção do mesmo, tais edições são, em geral, limitadas, sobretudo em relação à “estética” dos gráficos. Por exemplo, tanto gráficos de barras (simples) como histogramas são, por padrão, impressos na cor cinza. Da mesma forma, existe uma sequência pré-estabelecida de cores para as fatias de um gráfico de pizza. É possível colorir estes gráficos com outras cores sem a necessidade de alterar a paleta de cores do R Commander? Neste material, aprenderemos a fazer edições em gráficos que não são suportadas nos seus respectivos menus de construção. Ou seja, iremos ***além dos menus*** geradores de gráficos do R Commander.


Por comodidade, usaremos neste material o mesmo conjunto de dados utilizado na Seção 4 da apostila *Introdução ao R Commander - Notas de Aula: o conjunto de dados escrito na primeira planilha (Plan1)* do arquivo *Bussab.xlsx* (ver Subseção 2.2 em [1] para importar este arquivo com o R Commander). Além das variáveis originalmente escritas neste arquivo, também vamos utilizar as variáveis *Casado_fator* e *Instrução* (ver Subseção 3.8.1 em [1] para criar estas variáveis no conjunto de dados *Bussab*). Após a importação do conjunto de dados e a criação das “novas” variáveis,

é aconselhável **salvar** este conjunto de dados (menu *Dados → Conjunto de dados ativo → Salvar conjunto de dados ativo*), evitando que seja necessário repetir o procedimento exposto neste parágrafo (para facilitar, dê o nome *Bussab* ao arquivo salvo).

2 Linhas de comando

Como já comentado, nosso intuito com este material é fazer modificações na estrutura de gráficos gerados pelo R Commander, indo além dos menus. Mas como podemos fazer isto?

Cada comando executado no R Commander por meio de menus gera, na janela *R Script* do R Commander, uma linha de comando. Esta linha gerada retornaria a mesma saída do que foi feito por meio de menus, uma vez submetida (ou seja, no jargão de programação, quando “rodamos” esta linha de comando). A última linha de comando na janela *R Script* se refere ao último comando executado, a penúltima linha de comando se refere ao penúltimo comando executado, e assim por diante.

Como exemplo, abra o R, carregue o R Commander (digitando `require(Rcmdr)` na janela *R Gui* e apertando  logo após) e siga os passos abaixo.

1. Carregue o conjunto de dados *Bussab* (salvo no formato *.RData*) no menu *Dados → Carregar conjunto de dados*.
2. Acesse o menu *Gráficos → Gráfico de Barras*, selecione *Proced* e dê OK.
3. Acesse o menu *Gráficos → Gráfico de Pizza*, selecione *Instrução* e dê OK.

Na janela *R Script* da Figura 1 estão expressas as linhas de comando associadas aos comandos executados acima. Note que a última linha **não começa colada à margem, o que significa que ela é continuação da linha imediatamente superior**. Portanto, as duas últimas linhas na janela *R Script* da Figura 1 consistem, na verdade, de apenas uma linha de comando, a qual se refere ao gráfico de pizza da variável *Instrução* (última linha de comando se refere ao último comando executado). Ainda, cabe ressaltar que a linha de comando `library(colorspace, pos=16)` refere-se ao carregamento de um pacote necessário para colorir automaticamente gráficos de pizza (tal linha é gerada assim que clicamos no menu para geração de gráficos de pizza pela primeira vez após carregar o

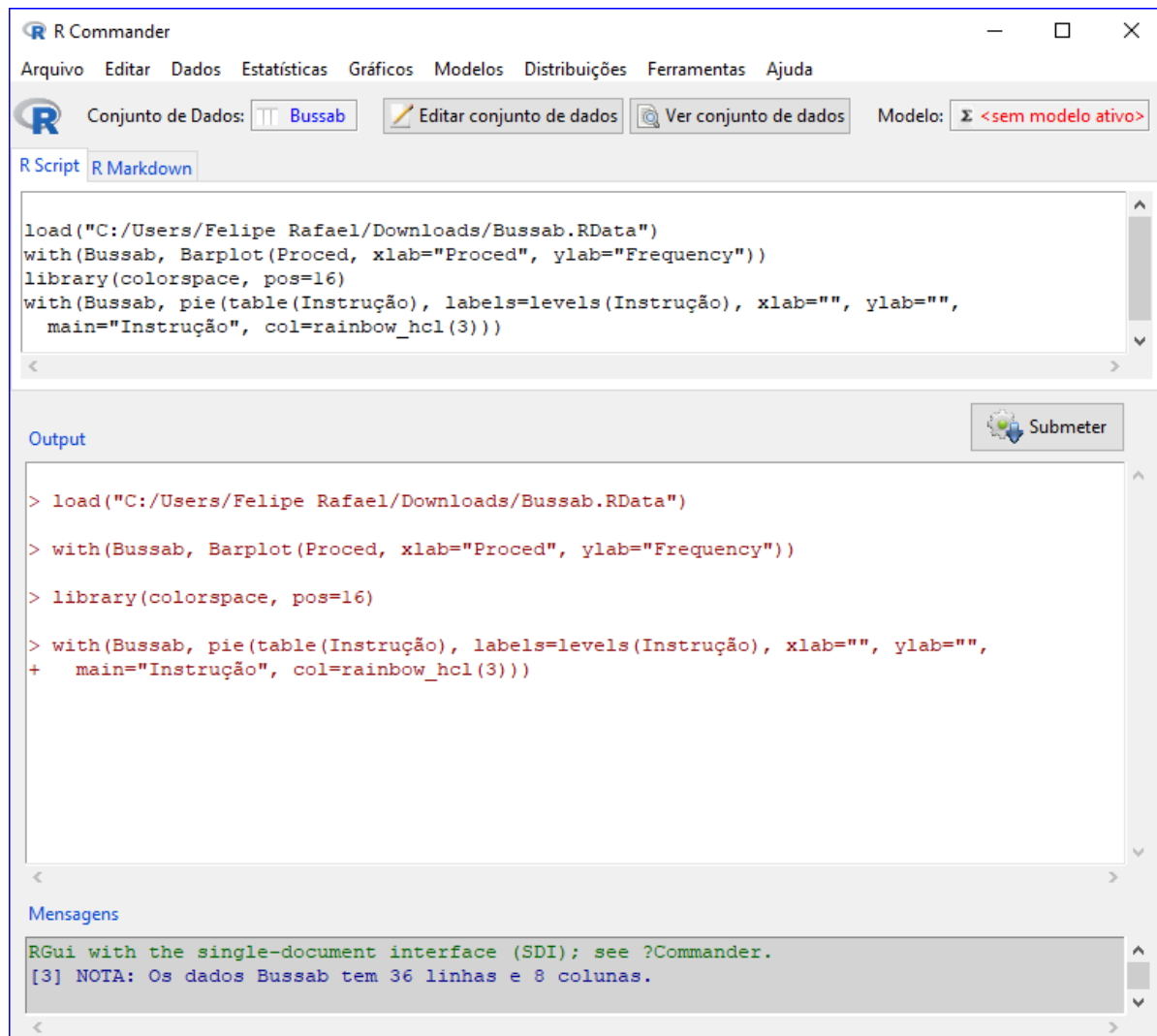


Figura 1: Apresentando linhas de comando na janela *R Script* do R Commander.

R Commander). Portanto, as três últimas linhas na janela *R Script* da Figura 1 são, na verdade, duas linhas de comando geradas pelo Passo 3. Seguindo nosso “caminho de volta”:

- a linha `with(Bussab, Barplot(Proced, xlab="Proced", ylab="Frequency"))` foi gerada ao executarmos o Passo 2 (geração do gráfico de barras da variável *Proced*); e
- a linha `load("C:/Users/Felipe Rafael/Downloads/Bussab.RData")` foi gerada ao executarmos o Passo 1 (carregamento do conjunto de dados *Bussab*, no formato *.RData*).

2.1 Submetendo linha de comando

As linhas de comando exibidas na janela *R Script* do R Commander não são apenas “decorações”. Podemos **submiter** qualquer linha de comando exibida na janela *R Script*, ou seja, re-executar algum comando (já realizado por meio de algum menu) sem precisar retornar ao menu em questão.

Acessando a janela *R Gui*, note que o gráfico atual na janela *R Graphics* é o gráfico de pizza da variável *Instrução*, pois foi o último gráfico a ser executado. Para retomar o gráfico de barras da variável *Proced* sem a necessidade de retornar ao menu de construção do gráfico de barras:

- selecione com o *mouse* toda a linha de comando associada ao gráfico de barras da variável *Proced* (isto é, de `with(Bussab` até `"Frequency"))`); e
- clique no botão *Submiter* (que se localiza entre as janelas *R Script* e *Output*, do lado direito) ou use o atalho do teclado `Ctrl` + `R`.

Feito isto, o gráfico de barras da variável *Proced* é novamente gerado na janela *R Graphics*.

Exercício 1 Refaça o gráfico de pizza da variável *Instrução* por meio da última linha de comando na janela *R Script*, marcando de `with(Bussab, pie` até `col=rainbow_hcl(3)))` com o *mouse* e clicando em *Submiter* logo após.

Observação 1 Quando a linha de comando que desejamos *submiter* ocupa apenas uma linha na janela *R Script*, não é necessário selecionar toda a linha de comando; basta deixar o cursor sobre a linha desejada (clitando com o *mouse* em qualquer lugar desta linha ou conduzindo o cursor até ela com as setas do teclado).

2.2 Submetendo linha de comando modificada

A grande motivação em *submiter* linhas de comando exibidas na janela *R Script* reside no fato de **podermos modificá-las antes de submetê-las** (da mesma forma como

explicado na Subseção 2.1). Observe, por exemplo, a linha de comando associada ao gráfico de barras da variável *Proced*. No seu final, está escrito `ylab="Frequency"`, que define o rótulo do eixo-y. Substitua **Frequency** por **Frequência** e submeta esta linha de comando. Note que, na janela *R Graphics*, surgiu um novo gráfico, com o termo *Frequência* escrito no rótulo do eixo-y em vez de *Frequency*.

Em particular, a modificação acima também poderia ser feita pelo menu *Gráficos → Gráfico de Barras*. Entretanto, há uma gama de possibilidades de edição desta linha de comando que não é suportada pelo menu de construção de gráfico de barras. Isto procede não apenas para gráficos de barras, mas para quase todas os resultados que o R Commander fornece por meio de menus, sejam eles geradores de gráficos ou não. Em particular, estamos interessados em editar saídas gráficas, e isto será amplamente abordado na Seção 4 deste material.

3 Cores disponíveis no R

Uma das modificações mais interessantes que podemos fazer via edição de linhas de comando (presentes na janela *R Script*) é mudar as cores de um gráfico. Como uma ilustração inicial, faça um histograma da variável *Renda* por meio do menu *Gráficos → Histograma*. Uma linha de comando do tipo


```
with(Bussab, Hist(Renda, scale="frequency", breaks="Sturges", col="darkgray"))
```

surgirá na janela *R Script*. Atente ao termo `col="darkgray"` na linha acima, no qual *col* se refere a cor (*col* é uma abreviação do inglês *color*) e *darkgray* significa cinza escuro, que é a cor padrão das barras de histogramas no R Commander. Portanto, se substituirmos *darkgray* pelo nome de outra cor em inglês (por exemplo: *red*) e submetermos tal linha de comando, será gerado um histograma com barras de outra cor (seguindo o exemplo: barras vermelhas).

Mas como saber quantas e quais cores o R disponibiliza?

Na janela *R Console* (dentro da janela *R Gui*), digite


```
colors()
```

e aperte . A lista de todas as cores disponíveis no R surgirá na janela *R Console* (cerca de 650 cores). Tal lista também segue no Anexo I (página 47), com o nome de cada cor escrito com a respectiva cor.

Observação 2 A lista de todas as cores disponíveis no R também pode ser gerada digitando `colors()` em qualquer linha em branco na janela R Script do R Commander e clicando no botão Submeter em seguida (sem retirar o cursor da linha na qual foi digitado `colors()`). A lista de cores surgirá na janela Output do R Commander.

Observação 3 Algumas (poucas) cores no R tem uma rotulação numérica:

0	equivale a	"white" (branco),
1	equivale a	"black" (preto),
2	equivale a	"red" (vermelho),
3	equivale a	"green3" (verde),
4	equivale a	"blue" (azul),
5	equivale a	"cyan" (ciano),
6	equivale a	"magenta" (magenta),
7	equivale a	"yellow" (amarelo),
8	equivale a	"gray" (cinza).

Isso quer dizer que, no lugar de qualquer uma das cores acima entre aspas, basta escrever o número correspondente (sem as aspas) que o resultado será o mesmo. Por exemplo, as expressões

```
col="red"
```

e

```
col=2
```

surtirão o mesmo resultado dentro de uma linha de comando quando submetida. O mesmo vale para vetor de cores: por exemplo, as expressões

```
col=c("blue","yellow")
```

e

```
col=c(4,7)
```

surtirão o mesmo efeito dentro de uma linha de comando. Como exercício, substitua a expressão `col="red"` (ou `col="darkgray"`) na linha de comando do histograma da variável Renda por `col=c(4,7)` ou `col=c("blue","yellow")` e submeta-a. Atente que esta linha de comando editada termina com três “fecha parênteses”.

4 Modificando a estrutura de gráficos gerados pelo R Commander

Nesta seção, veremos como modificar a estrutura original de gráficos gerados pelo R Commander. Ou seja, após gerar um gráfico da forma “limitada” que este pacote permite, ***editaremos a linha de comando associada a este gráfico*** de forma que, ao ser submetida, retorne um gráfico que contorne a limitação dos menus gráficos do R Commander. Cada subseção abordará um tipo diferente de representação gráfica: gráficos de barras (simples), gráficos de pizza, histogramas, boxplots, gráficos de barras múltiplas e diagramas de dispersão. Por comodidade, os gráficos gerados pelos menus do R Commander – cujas linhas de comando associadas serão editadas – no começo das subseções acima citadas estão desprovidos de título, no intuito de não deixar tais linhas de comando muito extensas. Mas mesmo quando modificamos algo na janela de construção dos gráficos (como *rótulo do eixo-x*, *rótulo do eixo-y*, *título do gráfico*, entre outros), as edições listadas na sequência deste material continuam válidas.

4.1 Gráfico de barras simples

Vamos começar com edições em gráficos de barras simples. Para ilustrarmos as edições neste tipo de gráfico, vamos gerar inicialmente um gráfico de barras da variável *Proced* simplesmente acessando o menu *Gráficos → Gráfico de Barras*, clicando em *Proced* e, logo após, em OK. A linha de comando gerada é exatamente a mesma obtida na Seção 2:

```
with(Bussab, Barplot(Proced, xlab="Proced", ylab="Frequency"))
```

Todas as subseções da Subseção 4.1 tomarão como ponto de partida a linha de comando acima.

4.1.1 Mudando a cor das barras: apenas uma cor

Para modificar a cor padrão cinza das barras do gráfico de barras para outra cor (por exemplo: vermelho), adicione antes do penúltimo “fecha parênteses” da linha de comando do gráfico de barras a expressão `,col="red"` (ou a cor de sua preferência no lugar de *red*). No exemplo do gráfico de barras da variável *Proced*, a linha de comando ficará na forma

```
with(Bussab, Barplot(Proced, xlab="Proced", ylab="Frequency",col="red"))
```

Selecione toda a linha e clique em *Submeter* para gerar o gráfico de barras da variável *Proced* com barras vermelhas.

4.1.2 Mudando a cor das barras: duas ou mais cores

Ainda sobre a edição de cores das barras de um gráfico de barras simples, é possível colorirmos cada barra com uma cor diferente. Para tal, devemos escrever, no lugar de `"red"`, um **vetor de cores**. Como no exemplo que estamos seguindo há três barras (variável *Proced* possui três atributos), então este vetor de cores é algo do tipo

```
c("cor_para_1ª_barra","cor_para_2ª_barra","cor_para_3ª_barra")
```

Como ilustração, use a expressão `,col=c("blue","pink","green")` no lugar de `,col="red"` na linha de comando da Subseção 4.1.1 (note que, com esta modificação, a linha de comando terminará com três “fecha parênteses”). Ao selecionarmos toda a linha de comando e clicarmos em *Submeter*, será gerado um gráfico de barras com a primeira barra na cor azul (*blue*), a segunda barra na cor rosa (*pink*) e a terceira barra na cor verde (*green*).

Observação 4 Quando há um número menor de cores (no vetor de cores) do que barras (no gráfico de barras), as cores especificadas se repetem de maneira cíclica.

Observação 5 Também é possível mudar a cor padrão (preta) das bordas das barras de maneira análoga à mudança das cores no interior das barras: basta usar `border` no lugar de `col`. Como exercício, submeta a seguinte linha de comando:

```
with(Bussab, Barplot(Proced,xlab="Proced",ylab="Frequency",col=7,border="red"))
```

Note que `col=7` produz o mesmo efeito de `col="yellow"`, conforme mencionado na Observação 3. De maneira análoga, `border="red"` poderia ser substituído por `border=2` antes da submissão da linha de comando que a saída seria idêntica (gráfico de barras amarelas com bordas vermelhas).

4.1.3 Espaçamento entre as barras

Antes do penúltimo “fecha parênteses” da linha de comando do gráfico de barras, podemos adicionar uma expressão do tipo

```
,space=número_de_sua_preferência
```

para aumentar ou diminuir o espaço entre as barras. Quanto maior o número de sua preferência, maior será o espaçamento entre as barras. O padrão é 0.2, portanto qualquer número maior que 0.2 fará com que as barras fiquem mais espaçadas umas das outras, e qualquer número (positivo) menor que 0.2 fará com que as barras fiquem mais próximas umas das outras. Em particular, `,space=0` fará com que as barras fiquem coladas umas às outras, assemelhando-se a forma de um histograma.

4.1.4 Editando nomes abaixo das barras

Suponha que, em vez dos nomes dos atributos da variável *Proced* abaixo das respectivas barras no gráfico de barras (*Capital*, *Interior* e *Outro*), nós queiramos colocar os nomes *CAP*, *INT* e *OUT*, respectivamente. Para isto, devemos adicionar, antes do penúltimo “fecha parênteses” da linha de comando do gráfico de barras, uma expressão do tipo

```
,names.arg=c("nome_para_1ª_barra","nome_para_2ª_barra","nome_para_3ª_barra")
```

No exemplo acima, a expressão que deve ser adicionada antes do penúltimo “fecha parênteses” da linha de comando do gráfico de barras é

```
,names.arg=c("CAP","INT","OUT").
```

Observação 6 *É possível usar o comando `\n` para quebra de linha em qualquer um dos novos nomes das barras.*

4.1.5 Tamanho das fontes

Em um gráfico de barras, é possível modificar o tamanho da fonte:

- dos nomes abaixo das barras;
- dos rótulos dos eixos; e
- dos números no eixo vertical (frequências ou porcentagens).

As expressões que devem ser adicionadas para tais modificações são bem parecidas e seguem a mesma estrutura.

- Adicionar, antes do penúltimo “fecha parênteses” da linha de comando do gráfico de barras, a expressão

```
,cex.names=número_de_sua_preferência
```

faz com que o tamanho da fonte dos **nomes** abaixo das barras seja modificado (com a submissão da referida linha de comando).

- Adicionar, antes do penúltimo “fecha parênteses” da linha de comando do gráfico de barras, a expressão

```
,cex.lab=número_de_sua_preferência
```

faz com que o tamanho da fonte do **rótulo do eixo-x** e do **rótulo do eixo-y** seja modificado (com a submissão da referida linha de comando).

- Adicionar, antes do penúltimo “fecha parênteses” da linha de comando do gráfico de barras, a expressão

```
,cex.axis=número_de_sua_preferência
```

faz com que o tamanho da fonte dos **números** no eixo vertical seja modificado (com a submissão da referida linha de comando).

Em qualquer uma das três situações acima, quanto maior for o *número_de_sua_preferência*, maior será o tamanho da fonte. O “tamanho padrão” é 1, ou seja, qualquer número maior que 1 aumenta o tamanho da fonte, ao passo que qualquer número (positivo) menor que 1 diminui o tamanho da fonte.

4.1.6 Encurtando ou alongando o eixo vertical

No gráfico de barras da variável *Proced*, note que o eixo vertical (representando as frequências) vai de 0 até 12, apesar da barra relativa a região de procedência *Outro* ter altura 13 (isto é, 13 indivíduos incluídos). Para refazer o gráfico de barras, mas com o eixo indo, por exemplo, de 0 até 14 (em vez de 0 até 12), adicione antes do penúltimo “fecha parênteses” da linha de comando do gráfico de barras a expressão

```
,ylim=c(0,14).
```

A linha de comando ficará como algo do tipo

```
with(Bussab, Barplot(Proced, xlab="Proced", ylab="Frequency",ylim=c(0,14)))
```

Submeta tal linha de comando e veja você mesmo o resultado.

4.1.7 Gráfico de barras na horizontal

Alternativamente, um gráfico de barras pode ser exibido com barras na horizontal (o eixo das frequências passa a ser o eixo-x). Para tal, adicione a expressão

```
,horiz=TRUE
```

antes do penúltimo “fecha parênteses” da linha de comando do gráfico de barras e submeta-a.

Após a submissão acima, note que o “rótulo do eixo-x” e o “rótulo do eixo-y” no gráfico gerado não trocaram de lugar, o que, a rigor, deixa tal representação na horizontal equivocada. Para reparar este erro, troque `xlab` e `ylab` de posição na linha de comando do gráfico de barras e submeta-a novamente. Como ilustração, a linha de comando submetida deve ser algo do tipo

```
with(Bussab, Barplot(Proced, ylab="Proced", xlab="Frequency",horiz=TRUE)).
```

Exercício 2 Adicione a expressão `,las=2` logo após `,horiz=TRUE` na linha de comando acima e submeta-a para verificar a diferença. Para retirar a expressão `Proced` do rótulo do eixo-y, substitua `ylab="Proced"` por `ylab=""` e submeta novamente.

Observação 7 Caso desejemos encurtar/alongar o eixo de frequências ou porcentagens (conforme exposto na Subseção 4.1.6) em um gráfico de barras na horizontal, devemos usar `,xlim=` em vez de `,ylim=`.

4.1.8 Porcentagens dos atributos nas barras

É possível “carimbar” os percentuais de cada atributo sobre as respectivas barras no gráfico de barras. Inicialmente, veremos um passo a passo de como fazer isto no gráfico de barras da variável *Proced* (lembrando que o conjunto de dados ao qual pertence esta variável foi nomeado *Bussab*), para logo após generalizarmos a forma de obtermos gráficos assim para qualquer variável (qualitativa) de qualquer conjunto de dados.

1. Copie as seguintes linhas de comando e cole-as em linhas em branco da janela *R Script*:

```
contagem = table(Bussab$Proced)
porcent = round(contagem/sum(contagem)*100,2)
rotulo = paste(porcent,"%",sep="")
k=1:length(levels(Bussab$Proced))
b=k-0.5+k*0.2
```

Feito isto, selecione todas elas com o *mouse* (na janela *R Script*) e clique em *Submeter*.

2. Gere o gráfico de barras da variável *Proced* e faça as as edições de interesse (só não altere o espaçamento entre as barras e não as coloque na horizontal, por enquanto), de forma que o gráfico que permanecerá na janela *R Graphics* ao avançarmos para o próximo passo será o gráfico no qual as porcentagens serão “carimbadas”.
3. Copie a seguinte linha de comando e cole-a numa linha em branco da janela *R Script*:

```
text(b, 0.9*table(Bussab$Proced), rotulo, pos=3, col="white").
```

Submeta esta linha de comando para “carimbar” as porcentagens nas barras.

Observação 8 O número 2 no final da segunda linha de comando do Passo 1 determina o número de casas decimais das porcentagens. Ele pode ser alterado para qualquer outro número inteiro desejado. Já o número 0.2 no final da última linha de comando do Passo 1 se dá pelo fato de não termos editado espaçamento entre as barras – no caso de edição de espaçamento entre barras, 0.2 deve ser substituído pelo número após `,space=` na linha de comando do gráfico de barras. No caso de gráficos de barras na horizontal, é necessário trocar `b` e `0.9*table(Bussab$Proced)` de posição na linha de comando do Passo 3.

Para obter gráfico de barras como acima para qualquer outra variável qualitativa de qualquer conjunto de dados, é necessário substituir `Bussab$Proced` por uma expressão do tipo `nome_do_conjunto_de_dados$nome_da_variavel` na primeira e na quarta linha de comando do Passo 1, bem como na linha de comando do Passo 3. Neste último passo, a expressão `"white"` define a cor da “tinta do carimbo”, que pode ser alterada para qualquer outra cor de sua preferência. É possível também modificar o tamanho da fonte do “carimbo” adicionando `,cex=número_de_sua_preferência` antes o último “fecha parênteses” do Passo 3. Por fim, caso os “carimbos” fiquem abaixo (acima) de

onde se deseja, basta substituir o número 0.9 na linha de comando do Passo 3 por um número maior (menor) e repetir os Passos 2 e 3.

Exercício 3 (“carimbando” frequências em vez de porcentagens) Repita o procedimento para “carimbar” porcentagens nas barras do gráfico de barras da variável *Proced*, porém substitua a terceira linha de comando do Passo 1

```
rotulo=paste(porcent,"%",sep="")
```

por

```
rotulo=paste(contagem,sep="").
```

Exercício 4 (“carimbando” frequências e porcentagens) Repita o procedimento para “carimbar” porcentagens nas barras do gráfico de barras da variável *Proced*, porém substitua a terceira linha de comando do Passo 1

```
rotulo=paste(porcent,"%",sep="")
```

por

```
rotulo=paste(contagem," (",porcent,"%",")",sep="").
```

Observação 9 Uma alternativa para que seja usada vírgula em vez de ponto como separador de casa decimal é realizar todo o procedimento descrito no passo a passo (ou no Exercício 4), porém substituindo `porcent` por

```
trunc(porcent),",",round(porcent%%1,2)*(10^2)
```

na terceira linha de comando do Passo 1. O número 2 (que aparece duas vezes na linha acima) remete à quantidade de casas decimais, e pode ser substituído por outro número inteiro de sua preferência, em consonância com a quantidade de casas decimais definida na segunda linha de comando do Passo 1 para fazer sentido.

4.2 Gráfico de pizza

Nesta subseção, trabalharemos com a variável *Instrução*, ainda que ela seja uma variável qualitativa ordinal. Por comodidade, simplesmente acesse o menu [Gráficos → Gráfico de pizza](#), selecione a variável *Instrução* e dê OK. Note que duas linhas de comando surgiram na janela *R Script*:

```
library(colorspace, pos=16)
```

e

```
with(Bussab, pie(table(Instrução), labels=levels(Instrução), xlab="",  
  ylab="", main="Instrução", col=rainbow_hcl(3)))
```

A primeira delas se refere ao carregamento da paleta de cores padrão que é utilizada para colorir as fatias de gráficos de pizza no R Commander, e podemos ignorá-la. Ainda, cabe destacar que esta linha só será exibida na janela *R Script* caso o menu [Gráficos → Gráfico de pizza](#) não tenha sido acessado anteriormente após o carregamento da janela *R Commander*. ***Já a segunda linha de comando refere-se ao gráfico de pizza gerado, e é nela que serão feitas as edições nas Subseções 4.2.1 a 4.2.5.***

4.2.1 Mudando as cores da fatias

Para modificar as cores das fatias de um gráfico de pizza no R Commander, devemos substituir a expressão

```
rainbow_hcl(3)
```

por um ***vetor de cores*** cuja quantidade de cores deve coincidir com a quantidade de fatias no gráfico. Note que este número está explícito na expressão acima. De fato, no gráfico de pizza da variável *Instrução*, temos três fatias (associadas aos atributos *Fundamental*, *Médio* e *Superior*), portanto o vetor de cores deve ser da forma

```
c("cor_para_1ª_fatia", "cor_para_2ª_fatia", "cor_para_3ª_fatia")
```

Por padrão, o R utiliza a ordem alfabética na ordenação dos atributos. Portanto, ao substituírmos a expressão

```
rainbow_hcl(3)
```

na linha de comando que gera o gráfico de pizza da variável *Instrução* por, digamos,

```
c("orange","lightblue","violet")
```

e submetermos tal linha, será gerado o gráfico de pizza da variável *Instrução* com fatia do ensino fundamental (“1ª fatia”) na cor laranja, fatia do ensino médio (“2ª fatia”) na cor azul claro e fatia do ensino superior (“3ª fatia”) na cor violeta.

4.2.2 Tamanho do gráfico de pizza

Para aumentar ou diminuir o gráfico de pizza (ou seja, aumentar ou diminuir o raio do círculo), devemos adicionar antes do penúltimo “fecha parênteses” da linha de comando do gráfico de pizza uma expressão do tipo

```
,radius=número_de_sua_preferência
```

e submeter a linha editada. O raio padrão é 0.8, portanto valores maiores que 0.8 como *número_de_sua_preferência* aumentam o gráfico, ao passo que valores (positivos) menores que 0.8 como *número_de_sua_preferência* diminuem o gráfico. Como exercício, submeta a seguinte linha de comando:

```
with(Bussab, pie(table(Instrução), labels=levels(Instrução), xlab="", ylab="",  
  main="Instrução", col=c("orange","lightblue","violet"), radius=0.5))
```

4.2.3 Tamanho da fonte: nomes dos atributos

Para modificar o tamanho da fonte dos nomes dos atributos da variável no gráfico de pizza, devemos adicionar antes do penúltimo “fecha parênteses” da linha de comando do gráfico de pizza uma expressão do tipo

```
,cex=número_de_sua_preferência
```

e submeter a linha editada. O tamanho de fonte padrão é 1, portanto valores maiores que 1 como *número_de_sua_preferência* aumentam o tamanho da fonte, ao passo que valores menores que 1 como *número_de_sua_preferência* diminuem o tamanho da fonte.

4.2.4 Editando nomes dos atributos ao lado das fatias

Os atributos da variável *Instrução* estão definidos como *Fundamental*, *Médio* e *Superior*, sendo que o R Commander sempre os lista nesta ordem (ordem alfabética), salvo tenha sido feita reordenação desses atributos no menu *Dados → Modificação de variáveis no conjunto de dados → Reordenar níveis dos fatores*. Para substituir os nomes *Fundamental*, *Médio* e *Superior* por, digamos, *E.F.*, *E.M.* e *E.S.*, respectivamente, no gráfico de pizza, devemos substituir a expressão

```
labels=levels(Instrução)
```

pela expressão

```
labels=c("E.F.", "E.M.", "E.S.")
```

na linha de comando referente ao gráfico de pizza e submetê-la logo após.

4.2.5 Porcentagens em gráficos de pizza

Algo muito interessante em um gráfico de pizza é explicitar a porcentagem que cada fatia ocupa no círculo (isto é, a porcentagem de cada atributo na distribuição de frequências). Inicialmente, veremos um passo a passo de como obter gráfico de pizza com porcentagens da variável *Instrução* (lembrando que o conjunto de dados ao qual pertence esta variável foi nomeado *Bussab*), para logo após generalizar a forma de obter gráficos de pizza com porcentagens para qualquer variável (qualitativa) de qualquer conjunto de dados.

1. Copie as seguintes linhas de comando e cole-as em linhas em branco da janela *R Script*:

```
contagem = table(Bussab$Instrução)
nomes = levels(Bussab$Instrução)
porcent = round(contagem/sum(contagem)*100,2)
rotulo = paste(nomes, "\n", porcent, "%", sep="")
```

Após isto, selecione todas elas com o *mouse* (na janela *R Script*) e clique em *Submeter*.

2. Faça um gráfico de pizza da variável *Instrução*. Note que nada mudou.
3. Na linha de comando do gráfico de pizza que acabamos de fazer, substitua `labels=levels(Instrução)` por `labels=rotulo` na janela *R Script*.
4. Selecione toda esta linha de comando do gráfico de pizza com o *mouse* (ela começa em `with(Bussab, pie` e termina com dois ou três “fecha parênteses”) e clique em *Submeter*. O gráfico está pronto!

Observação 10 O número 2 no final da terceira linha de comando do Passo 1 determina o número de casas decimais das porcentagens. Ele pode ser alterado para qualquer outro número inteiro desejado.

Para obter gráfico de pizza com porcentagens para qualquer outra variável qualitativa de qualquer conjunto de dados, é necessário substituir `Bussab$Instrução` por uma expressão do tipo `nome_do_conjunto_de_dados$nome_da_variavel` tanto na primeira como na segunda linha de comando do Passo 1. Ainda, cabe ressaltar que, no Passo 3, a expressão `labels=rotulo` entrará no lugar da expressão do tipo `labels=levels(nome_da_variavel)`.

Exercício 5 Repita o procedimento para colocar porcentagens no gráfico de pizza da variável *Instrução* do conjunto de dados *Bussab*, porém substituindo a última linha de comando do Passo 1

```
rotulo=paste(nomes, "\n", percent, "%", sep="")
```

por

```
rotulo=paste(nomes, " (", percent, "%", ") ", sep="").
```

Exercício 6 Repita o procedimento para colocar porcentagens no gráfico de pizza da variável *Instrução* do conjunto de dados *Bussab*, porém substituindo a última linha de comando do Passo 1

```
rotulo=paste(nomes, "\n", percent, "%", sep="")
```

por

```
rotulo=paste(nomes,"\n"," ",contagem," indivíduos", " (" ,porcent,"%",") ",sep="").
```

Observação 11 *Uma alternativa para que seja usada vírgula em vez de ponto como separador de casa decimal é realizar todo o procedimento descrito no passo a passo (ou nos dois exercícios propostos acima), porém substituindo*

```
porcent
```

por

```
trunc(porcent),",",round(porcent%%1,2)*(10^2)
```

na última linha de comando do Passo 1. O número 2 (que aparece duas vezes na linha acima) remete à quantidade de casas decimais, e pode ser substituído por outro número inteiro de sua preferência, em consonância com a quantidade de casas decimais definida na terceira linha de comando do Passo 1 para fazer sentido.

4.3 Histograma

O conjunto de dados com o qual estamos trabalhando neste material (aqui nomeado *Bussab*) possui três variáveis quantitativas: *Filhos*, *Idade* e *Renda*.¹ Trabalharemos nesta subseção com edições em histogramas tomando por base o histograma da variável *Idade*, exceto na Subseção 4.3.3, na qual usaremos a variável *Filhos* para geração de histogramas. Portanto, acesse o menu [Gráficos → Histograma](#) e, por comodidade, simplesmente selecione a variável *Idade* e dê OK. **A linha de comando gerada, a qual será editada nas próximas subseções (exceto na Subseção 4.3.3), é a seguinte:**

```
with(Bussab, Hist(Idade, scale="frequency", breaks="Sturges", col="darkgray")).
```

¹As variáveis *Casado* e *Instr*, apesar de compreendidas pelo R Commander como quantitativas, expressam na verdade rótulos numéricos de variáveis qualitativas.

4.3.1 Modificando cores das barras

Conforme já ilustrado no começo da Seção 3, substituir a cor cinza padrão de histogramas no R Commander pela cor de sua preferência passa simplesmente por substituir a expressão `darkgray` na linha de comando do histograma pela cor de sua preferência – dentre as disponibilizadas pelo R – e submeter esta linha modificada.

Para histogramas com barras de cores diferentes umas das outras, é necessário substituir a expressão `"darkgray"` (incluindo as aspas) por um vetor de cores. Como temos seis classes no histograma gerado para a variável *Idade*, devemos usar um vetor de seis cores para que cada barra (associada a uma classe) fique de uma cor diferente. Quando o número de cores no vetor de cores é inferior ao número de classes, as cores se repetem de maneira cíclica. Como exercício, substitua a expressão `"darkgray"` (incluindo as aspas) pelo vetor de cores

```
c("blue","red"),
```

submeta tal linha de comando modificada e veja o resultado. Logo após, substitua o vetor de cores acima pelo vetor de cores

```
c("blue","red","black","white","darkgreen","maroon")
```

e submeta tal linha de comando modificada.

Observação 12 Assim como visto para gráficos de barras, é possível modificar a cor original (preta) das **bordas** das barras de um histograma adicionado, antes do penúltimo “fecha parênteses” da linha de comando de um histograma, uma expressão do tipo

```
,border="cor_de_sua_preferência"
```

(ou ainda, um vetor de cores no lugar de "cor_de_sua_preferência"). A título de ilustração, submeta a seguinte linha de comando:

```
with(Bussab, Hist(Idade, scale="frequency", breaks="Sturges", col=
  c("blue","red"),border="green"))
```

4.3.2 Listar os valores que delimitam as classes

Em alguns histogramas feitos pelo R Commander, as marcações no eixo horizontal não deixam claro onde começam e/ou onde terminam cada classe. No conjunto de dados que estamos utilizando, por exemplo, isto ocorre no histograma da variável **Renda** (com número de classes em *<auto>*). Para obter uma lista dos valores que delimitam as classes, digite `$breaks` entre os dois “fecha parênteses” que encerram a linha de comando do histograma. Marque toda esta linha de comando e clique em *Submeter*. O resultado desejado é exibido na janela *Output*, na cor azul. Por exemplo, para o histograma da variável *Idade*, modifique linha de comando “original”

```
with(Bussab, Hist(Idade, scale="frequency", breaks="Sturges", col="darkgray"))
```

para

```
with(Bussab, Hist(Idade, scale="frequency", breaks="Sturges", col="darkgray"))$breaks)
```

e clique em *Submeter*. Feito isto, a sequência de números 20 25 30 35 40 45 50 surge na janela *Output*, remetendo aos valores que delimitam as classes. Portanto, as classes consideradas neste histograma são: 20 – 25 ; 25 – 30 ; 30 – 35 ; 35 – 40 ; 40 – 45 ; e 45 – 50.

Observação 13 Para exibir a frequência de cada classe na janela *Output*, basta repetir o processo acima, porém digitando `$counts` no lugar de `$breaks`. Ainda, para exibir o ponto médio de cada classe na janela *Output*, basta fazer o processo acima digitando `$mids` no lugar de `$breaks`.

Exercício 7 Faça um histograma da variável *Renda* deste conjunto de dados e, logo após, verifique a frequência em cada classe por meio do método exposto na Observação 13.

Observação 14 Alternativamente, é possível verificar os valores que delimitam cada classe de um histograma, bem como a frequência e percentual em cada classe, por meio do menu [Estatísticas](#) → [Resumos](#) → [Resumos Numéricos](#) (marque a caixinha *Binned Frequency Counts* na aba *Estatísticas* antes de clicar em *OK*). As classes exibidas na janela *Output* são as mesmas consideradas pelo histograma com o número de classes em *<auto>*.

4.3.3 Modificar os valores que delimitam cada classe

Acesse o menu *Gráficos → Histograma* e gere um histograma da variável *Filhos*, mantendo o número de classes em *<auto>*. Note que a primeira das classes (que começa em 0 e termina em 1) computa tanto os quatro indivíduos com nenhum filho como os cinco indivíduos com (exatamente) 1 filho (altura da barra = $4+5 = 9$). Como associar a cada barra apenas um valor na contagem de filhos? Uma solução é criar classes de tamanho 1 com valores que delimitam as classes terminando em “,5”, isto é: classe -0,5 \rightarrow 0,5 (contém apenas o valor 0 dentre os possíveis para número de filhos); classe 0,5 \rightarrow 1,5 (contém apenas o valor 1 dentre os possíveis para número de filhos); classe 1,5 \rightarrow 2,5 (contém apenas o valor 2 dentre os possíveis para número de filhos); e assim sucessivamente até a classe 4,5 \rightarrow 5,5, a qual contém apenas o valor 5 dentre os possíveis para número de filhos, que é o maior valor de número de filhos nos indivíduos avaliados neste conjunto de dados.

Para implementar a sugestão acima no R Commander, substitua `"Sturges"` (incluindo as aspas) por `-0.5:5.5` na linha de comando gerada pelo histograma da variável *Filhos* (na linguagem R, a expressão -0.5:5.5 representa a sequência de números de -0,5 até 5,5 saltando de 1 em 1 unidade, isto é: -0,5 ; 0,5 ; 1,5 ; 2,5 ; 3,5 ; 4,5 ; 5,5). Após feita esta substituição, marque toda a linha de comando (de `with(Bussab, Hist` até os dois “fecha parênteses” que encerram a linha de comando) e clique em *Submeter* para termos como saída o histograma da forma desejada. Note que o único valor possível em cada classe é exatamente o seu respectivo ponto médio.

Exercício 8 Na linha de comando (do histograma da variável *Filhos*) modificada acima, faça com que os pontos médios de cada classe sejam explicitados na janela *Output* digitando `$mids` entre os dois “fecha parênteses” que encerram a linha de comando e submetendo-a em seguida.

Observação 15 Caso desejemos classes com amplitude diferente de 1, devemos usar expressões do tipo `seq(início_da_primeira_classe,final_da_última_classe,amplitude)` no lugar de expressões como `-0.5:5.5`. A título de ilustração, substitua a expressão `-0.5:5.5` (ou `"Sturges"`) na linha de comando do histograma da variável *Filhos* por `seq(-0.5,5.5,2)` para gerar um histograma com apenas três classes (-0,5 \rightarrow 1,5, 1,5 \rightarrow 3,5 e 3,5 \rightarrow 5,5), cada uma contendo dois possíveis valores para número de filhos.

Exercício 9 Após construir o histograma sugerido na Observação 15, acesse o menu [Estatísticas](#) → [Resumos](#) → [Resumos Numéricos](#). Selecione a variável Idade na aba Dados, marque a caixinha Binned Frequency Counts na aba Estatísticas e dê OK. Na última linha de comando gerada, ou seja,

```
boxed(binnedCounts(Bussab[, "Filhos", drop=FALSE]) ,
```

insira a expressão `boxed(, breaks=seq(-0.5, 5.5, 2))` entre o "fecha colchetes" e o "fecha parênteses" e submeta-a. A distribuição de frequências por classes condizente com o histograma da Observação 15 será exibida na janela Output.

4.3.4 Histograma de um único grupo do conjunto de dados

O menu de construção de histogramas no R Commander possibilita a geração de histogramas por grupos conforme uma variável qualitativa do mesmo conjunto de dados: por exemplo, um histograma das idades no grupo de indivíduos casados e um histograma das idades no grupo de indivíduos que não são casados (variável qualitativa *Casado_fator* escolhida como variável de grupo). Entretanto, estes histogramas são exibidos simultaneamente na mesma janela *R Graphics*, resultando em gráficos “encolhidos” e pouco elegantes.

Suponha que só estejamos interessados no histograma das idades no grupo de indivíduos casados, isto é, no grupo de indivíduos com atributo “Sim” para a variável *Casado_fator*. Em particular, para a modificação sugerida nesta subseção, é razoável fazer um histograma com *rótulo do eixo-x* e *Título do gráfico* de acordo com o histograma de interesse antes de realizar a modificação e submissão da linha de comando. Uma sugestão é deixar o *rótulo do eixo-x* em branco e, como título do gráfico, digitar *Idades dos indivíduos casados* (adicionalmente, podemos substituir *<auto>* por *Frequência* em *rótulo do eixo-y* para que a expressão padrão *frequency* seja substituída por *Frequência*). De qualquer forma, note que a linha de comando do histograma da variável *Idade* sempre começa com `boxed(with(Bussab, Hist(Idade, ,`. Para a geração do histograma de interesse, digite a seguinte expressão entre `boxed(Idade` e a vírgula que a sucede:

```
boxed(which(Casado_fator=="Sim"))].
```

Ao submeter tal linha de comando modificada (conforme exibida abaixo), será gerado o histograma de interesse:

```
with(Bussab, Hist(Idade[which(Casado_fator=="Sim")], scale="frequency",
  breaks="Sturges", col="darkgray", xlab="", ylab="Frequência", main=
  "Idades dos indivíduos casados"))
```

Observação 16 De forma análoga, é possível gerar o histograma das idades apenas dos indivíduos deste conjunto de dados que não são casados, isto é, o grupo de indivíduos com atributo “Não” para a variável `Casado_fator`. Isto poderia ser feito apenas substituindo `Sim` por `Não` na linha de comando acima e submetendo-a. Todavia, o título do gráfico permaneceria da forma que fizemos anteriormente (Idades dos indivíduos casados). Felizmente, isto pode ser modificado na própria linha de comando antes de submetê-la, substituindo `main="Idades dos indivíduos casados"` por, digamos, `main="Idades dos indivíduos que não são casados"`, evitando assim a necessidade de acessar novamente o menu de construção de histogramas para modificar o título do gráfico e editar uma nova linha de comando gerada pelo R Commander.

4.3.5 Polígono de frequências

Esta subseção apresenta uma forma de gerar histograma com seu polígono de frequências sobreposto. Faremos aqui o histograma e polígono de frequências da variável *Idade* por meio do passo a passo a seguir, o qual pode ser estendido para qualquer variável quantitativa de qualquer conjunto de dados.

1. Primeiramente, deve ser gerado o histograma que ganhará seu respectivo polígono de frequências sobreposto. Por comodidade, tomemos aqui a mesma linha de comando explicitada no início da Subseção 4.3. Uma forma simples de recuperá-la (após as modificações já realizadas nas subseções anteriores) é acessar o menu [Gráficos → Histograma](#), clicar em *Resetar*, selecionar a variável *Idade* e clicar em OK. Feito isto, será gerada novamente, na janela *R Script*, a linha de comando

```
with(Bussab, Hist(Idade, scale="frequency", breaks="Sturges", col="darkgray").
```

2. Na janela *R Script*, digite `h=` no início desta linha de comando, de forma que a mesma comece com `h=with` em vez de `with`. Selecione toda esta linha de comando modificada e clique em *Submeter*.
3. Copie a linha de comando abaixo e cole-a numa linha em branco da janela *R Script*, submetendo-a em seguida para gerar o polígono de frequências sobre o histograma gerado no Passo 1:

```
lines(c(min(h$breaks),h$mids,max(h$breaks)),c(0,h$counts,0),col="red",lwd=2).
```

Observação 17 A cor vermelha (red) na linha de comando Passo 3 pode ser substituída pela cor de sua preferência. Já a expressão `lwd=2` nesta mesma linha de comando regula a espessura das linhas do polígono de frequências (quanto maior o número após `lwd=`, mais espessas ficarão as linhas que formam o polígono de frequências).

Observação 18 O passo a passo acima também funcionaria caso o histograma traçado no Passo 1 fosse construído com porcentagens (em vez de contagem de frequências) no eixo vertical. Por outro lado, no caso da escolha de Densidades no campo Escala do eixo, seria necessário substituir `h$counts` por `h$density` na linha de comando do Passo 3.

4.4 Boxplot

Assim como os histogramas, representações gráficas por meio de boxplots também são úteis para variáveis quantitativas. Nesta subseção, serão expostas edições em boxplots da variável *Renda*, exceto nas Subseções 4.4.7 e 4.4.8, nas quais trabalharemos com boxplots da variável *Idade* em cada nível de instrução. Portanto, para começarmos, acesse o menu [Gráficos → Boxplot](#) e, por comodidade, apenas selecione a variável *Renda* e dê OK, o que gera uma linha de comando na janela *R Script* conforme abaixo:

```
Boxplot( ~ Renda, data=Bussab, id=list(method="y")).
```

As expressões que adicionaremos em linhas de comando de boxplots devem ser inseridas antes do último “fecha parênteses” da linha de comando, e não mais antes do penúltimo “fecha parênteses”, como foi feito em gráficos de barras, gráficos de pizza e histogramas.

4.4.1 Mudando a cor da caixa e da borda de um boxplot

É possível modificar a cor (originalmente branca) no interior da caixa de um boxplot inserindo a expressão `,col="cor_de_sua_preferência"` antes do “fecha parênteses” que encerra a linha de comando do boxplot, e/ou modificar a cor (originalmente preta) das bordas de um boxplot inserindo a expressão `,border="cor_de_sua_preferência"` antes do “fecha parênteses” que encerra a linha de comando do boxplot. Por exemplo, submeter a linha de comando

```
Boxplot( ~ Renda, data=Bussab, id=list(method="y"),col="blue")
```

gera o boxplot da variável *Renda* com caixa na cor azul (e borda na cor original preta), ao passo que a linha de comando

```
Boxplot( ~ Renda, data=Bussab, id=list(method="y"),border="red"),
```

após submetida, produz o boxplot da variável *Renda* com borda vermelha (e caixa na cor original branca). Para editar, em um mesmo boxplot, cor da caixa e cor da borda, basta incluir as duas expressões que configuram cor antes do “fecha parênteses” que encerra a linha de comando original, isto é, algo do tipo

```
Boxplot( ~ Renda, data=Bussab, id=list(method="y"),col="blue",border="red").
```

Submetendo a linha de comando acima na janela *R Script*, será gerado o boxplot da variável *Renda* com caixa azul e borda vermelha.

4.4.2 Mudando as cores das caixas e das bordas de boxplots em grupos

Ao gerar boxplots por grupos (isto é, por atributos de uma variável qualitativa), mais de um boxplot é exibido como resultado. Realizar modificações como na Subseção 4.4.1 colore todas as caixas da mesma cor e/ou todas as bordas da mesma cor. Para cores diferentes em cada boxplot (sejam nas caixas ou nas bordas), o procedimento é análogo ao exposto na

Subseção 4.4.1, porém usando vetor de cores no lugar de apenas uma cor. Como exemplo, façamos o boxplot da variável *Renda* no menu *Gráficos → Boxplot*, porém clicando na caixa *Gráfico por grupos* e selecionando *Casado_fator* como variável de grupo. Na linha de comando originalmente gerada

```
Boxplot(Renda~Casado_fator, data=Bussab, id=list(method="y")),
```

adicione a expressão

```
,col=c("blue","green"),border=c("red","purple")
```

antes do “fecha parênteses” que encerra a linha de comando original. Ao submeter tal linha modificada, o primeiro boxplot – da esquerda para a direita – é exibido com caixa azul (*blue*) e borda vermelha (*red*), e o boxplot seguinte com caixa verde (*green*) e borda roxa (*purple*).

4.4.3 Editando nomes dos atributos em boxplots por grupos

Em boxplots por grupos, é possível modificar os nomes dos atributos da variável qualitativa escolhida como variável de grupo no “eixo horizontal” dos boxplots. Como ilustração, vamos gerar boxplots das rendas pelos atributos da variável *Casado_Fator* (o mesmo construído na subseção anterior), porém com “*Casados*” no lugar de “*Sim*” e “*Solteiros*” no lugar de “*Não*”. Lembrando que a ordenação original dos atributos da variável *Casado_fator* traz *Não* antes de *Sim*, porém aqui estamos usando tal variável já com seus atributos reordenados conforme feito na Subseção 3.8.2 da apostila ***Introdução ao R Commander - Notas de Aula***, ou seja, *Sim* antes de *Não*. Portanto, em boxplots por grupos com a variável *Casado_fator* escolhida como variável de grupo, o primeiro boxplot (da esquerda para a direita) refere-se ao atributo *Sim* e o segundo boxplot refere-se ao atributo *Não*, e mudança proposta acima é realizada adicionando a expressão

```
,names=c("Casados","Solteiros")
```

antes do “fecha parênteses” que encerra a linha de comando original

```
Boxplot(Renda~Casado_fator, data=Bussab, id=list(method="y")),
```

submetendo-a logo após. Caso os atributos da variável *Casado_fator* estivessem em sua ordenação original, deveríamos usar `,names=c("Solteiros","Casados")` no lugar de

```
,names=c("Casados","Solteiros").
```

Observação 19 *Editar nomes dos atributos em boxplots por grupos é bastante útil quando o nome de alguns atributos da variável qualitativa escolhida como variável de grupo são muito extensos. Ainda, é possível usar quebra de linha com o comando `\n`. Como exercício, refaça a modificação acima sugerida, porém com a expressão*

```
,names=c("Indivíduos\nCasados","Indivíduos\nSolteiros")
```

em vez da expressão

```
,names=c("Casados","Solteiros")
```

.

4.4.4 Tamanho das fontes

Em boxplots, é possível modificar o tamanho da fonte dos rótulos dos eixos e o tamanho da fonte dos números no eixo dos valores da variável (simultaneamente com o tamanho da fonte dos nomes dos atributos da variável de grupo, no caso de boxplots por grupos). As expressões que devem ser adicionadas para tais modificações são parecidas e seguem a mesma estrutura.

- Adicionar, antes do “fecha parênteses” que encerra a linha de comando original do boxplot, a expressão

```
,cex.lab=número_de_sua_preferência
```

faz com que o tamanho da fonte do **rótulo do eixo-x** e do **rótulo do eixo-y** seja modificado (com a submissão da referida linha de comando).

- Adicionar, antes do “fecha parênteses” que encerra a linha de comando original do boxplot, a expressão

```
,cex.axis=número_de_sua_preferência
```

faz com que o tamanho da fonte dos números no eixo dos valores seja modificado (com a submissão da referida linha de comando). Ainda, no caso de boxplots por grupos, tal comando também modifica o tamanho da fonte dos nomes dos atributos da variável de grupo.

Em qualquer uma das duas situações acima, quanto maior o *número_de_sua_preferência*, maior será o tamanho da fonte. O “tamanho padrão” é 1, ou seja, qualquer número maior que 1 aumenta o tamanho da fonte, ao passo que qualquer número (positivo) menor que 1 diminui o tamanho da fonte.

4.4.5 Encurtando ou alongando o eixo vertical

O eixo vertical na janela que comporta boxplot(s) pode ser alongado ou encurtado com um comando do tipo

```
,ylim=c(inicio_do_eixo , final_do_eixo)
```

inserido antes do “fecha parênteses” que encerra a linha de comando do(s) boxplot(s) em questão. Por exemplo, a linha de comando

```
Boxplot(Renda~Casado_fator, data=Bussab, id=list(method="y"),ylim=c(0,25)),
```

após submetida, gera boxplots em uma janela com eixo vertical ligeiramente esticado em relação à janela original anteriormente gerada por meio do menu de construção de boxplots.

4.4.6 Caixa do boxplot “amassada” na altura da mediana

De forma a destacar ainda mais, do ponto de vista gráfico, a posição da mediana em um boxplot, uma alternativa é adicionar a expressão

```
,notch=TRUE
```

antes do “fecha parênteses” que encerra a linha de comando original do boxplot. Ao submeter tal linha de comando após esta modificação, a caixa do boxplot gerado (ou de cada boxplot gerado, em caso de boxplots por grupos) segue a forma de um polígono “amassado” na altura da mediana, como se esta fosse um “cinto apertado”. A título de ilustração, submeta as seguintes linhas de comando:

```
Boxplot( ~ Renda, data=Bussab, id=list(method="y"),notch=TRUE),
```

```
Boxplot(Renda~Casado_fator, data=Bussab, id=list(method="y"),notch=TRUE).
```


4.4.7 Valores discrepantes indicados por círculos preenchidos

Originalmente, valores discrepantes nos boxplots gerados pelo R Commander são indicados por círculos de borda preta e interior branco. Todavia, círculos preenchidos podem ser mais interessantes no sentido de dar um “destaque visual” maior aos valores discrepantes. Isto pode ser feito facilmente adicionando a expressão

```
,pch=19
```

antes do “fecha parênteses” que encerra a linha de comando original do boxplot e submetendo-a. Alternativamente, também poderíamos utilizar a expressão

```
,pch=21,bg="black"
```

no lugar de `,pch=19` para obter o mesmo resultado. A vantagem deste método alternativo é que poderíamos escolher qualquer cor disponível no R para o preenchimento dos círculos no lugar de preto (*black*), com a cor do contorno do círculo definida pela cor da borda do(s) boxplot(s), cuja forma de edição foi exposta nas Subseções 4.4.1 e 4.4.2. As quatro linhas de comando abaixo são edições de linhas de comando geradas de boxplots por grupos da variável *Idade*, com *Instrução* como variável de grupo. Submeta cada uma delas e verifique o resultado.

```
Boxplot(Idade~Instrução, data=Bussab, id=list(method="y"),pch=19) ,
```

```
Boxplot(Idade~Instrução, data=Bussab, id=list(method="y"),pch=21,bg="red") ,
```

```
Boxplot(Idade~Instrução,data=Bussab,id=list(method="y"),pch=21,bg="red",border="red") ,
```

```
Boxplot(Idade~Instrução, data=Bussab, id=list(method="y"),pch=21,bg="black") .
```

4.4.8 Valores discrepantes identificados pelos respectivos valores da variável

Por padrão, boxplots gerados pelo R Commander marcam cada valor discrepante pela sua identificação (disposta na coluna cinza à esquerda do conjunto de dados quando clicamos na caixa *Ver conjunto de dados*). Entretanto, quando a identificação de cada observação resume-se apenas ao número de sua linha no conjunto de dados, tal informação costuma ser irrelevante. Por exemplo, ao gerarmos boxplots das idades pelo nível de instrução, o

boxplot das idades do nível de instrução *Ensino Médio* apresenta três valores discrepantes, identificados por 4, 27 e 35. Como, no conjunto de dados (aqui nomeado de) *Bussab*, a identificação de cada observação resume-se no número da linha, tais valores representam simplesmente que os indivíduos nas 4^a, 27^a e 35^a linhas do conjunto de dados tem idades discrepantes no grupo de indivíduos com ensino médio como nível de instrução. Tais identificações podem ser omitidas quando é selecionada a opção “*Não*” no campo *Identificar outliers* (aba *Opções*). Entretanto, veremos nesta subseção uma alternativa interessante, que consiste em identificar valores discrepantes pelos respectivos valores da variável.

Primeiramente, tome por base uma linha de comando associada ao boxplot de interesse (ou aos boxplots de interesse, no caso de boxplots por grupos). Tal qual na Subseção 4.4.7, tomaremos por base boxplots por grupos da variável *Idade*, com *Instrução* como variável de grupo, cuja linha de comando associada é dada por:

```
Boxplot(Idade~Instrução, data=Bussab, id=list(method="y")).
```

Substitua a expressão `method="y"` por

```
labels=Bussab$Idade
```

e submeta a linha de comando modificada. As marcações 4, 27 e 35 ao lado de cada valor discrepante são substituídas respectivamente por 21, 47 e 49, que são as idades (em anos completos) representadas por cada valor discrepante.

Exercício 10 Na linha de comando editada acima, substitua `labels=Bussab$Idade` por `labels=paste(Bussab$Idade,"anos")` e submeta-a. As marcações 21, 47 e 49) ao lado de cada valor discrepante são substituídas respectivamente por 21 anos, 47 anos e 49 anos.

Observação 20 A generalização deste tipo de edição passa por substituir *Bussab* e *Instrução* pelo nome do conjunto de dados em uso e pelo nome da variável de interesse, respectivamente. Ou seja, substituir a expressão `method="y"` por expressões do tipo

```
labels=nome_do_conjunto_de_dados$nome_da_variável
```

ou

```
labels=paste(nome_do_conjunto_de_dados$nome_da_variável,"unidade_de_medida").
```

4.5 Gráfico de barras múltiplas

Gráficos de barras múltiplas são gerados pelo mesmo menu que gera gráficos de barras simples: *Gráficos → Gráfico de barras*. Consequentemente, várias das edições vistas na Subseção 4.1 para gráficos de barras simples são aplicáveis (e razoáveis) em gráficos de barras múltiplas, *seguindo o mesmo procedimento* apresentado em subseções da Subseção 4.1, tais como:

- mudança de cores das barras e/ou das bordas das barras (conforme Subseção 4.1.2);
- editar nomes dos atributos da variável exibidos abaixo de cada grupo de barras “coladas” (conforme Subseção 4.1.4);
- tamanho das fontes (conforme Subseção 4.1.5);
- alongamento ou encurtamento do eixo vertical (conforme Subseção 4.1.6);
- gráfico com barras na horizontal (conforme Subseção 4.1.7).

Para verificarmos algumas das edições permitidas em gráficos de barras múltiplas, vamos trabalhar com o gráfico de barras múltiplas das variáveis *Proced* e *Instrução* do conjunto de dados *Bussab*, com barras lado a lado e legenda no canto superior esquerdo. Para tal, acesse o menu *Gráficos → Gráfico de barras*. É recomendável clicar em *Resetar* para anular as modificações feitas anteriormente na janela de construção de gráficos de barras na Subseção 4.1. Selecione a variável *Proced*, clique na caixa *Gráfico por grupos*, selecione *Instrução* como variável de grupo e dê OK. Na aba *Opções*, marque “*Lado a lado (paralelo)*” e “*À esquerda*” nos campos *Estilo de barras agrupadas* e *Posição da legenda* para enfim clicar em OK. A linha de comando gerada na janela *R Script* após este procedimento segue abaixo (os campos *rótulo do eixo-x*, *rótulo do eixo-y* e *Título do gráfico* da aba *Opções* foram mantidos em *<auto>* para evitar uma linha de comando ainda mais extensa):

```
with(Bussab, Barplot(Proced, by=Instrução, style="parallel",  
  legend.pos="topleft", xlab="Proced", ylab="Frequency"))
```

Em toda a Subseção 4.5, usaremos a linha de comando acima como referência nas edições.

O gráfico de barras múltiplas gerado (envolvendo as variáveis *Proced* e *Instrução*) pode apresentar uma pequena parte da legenda sobreposta à barra relativa aos indivíduos procedentes da capital e que tem ensino médio como nível de instrução. Uma forma de contornar isto é alongar o eixo das frequências, o que desloca a legenda automaticamente para cima. Note que o eixo das frequências vai de 0 até 7. Vamos portanto colocar este eixo indo de 0 até um valor ligeiramente maior que 7 (por exemplo, de 0 até 8). Para tal, inclua a expressão

```
,ylim=c(0,8)
```

antes do penúltimo “fecha parênteses” da linha de comando original deste gráfico de barras múltiplas e submeta-a. O gráfico gerado não apresenta mais nenhum “choque” entre legenda e barras.

4.5.1 Posição da legenda

Na linha de comando que estamos usando como referência, atente à expressão `legend.pos="topleft"`. Ela configura a posição da legenda no gráfico de barras múltiplas (*topleft* remete à: *no topo, à esquerda*), e está desta forma pelo fato de termos escolhido a opção “À esquerda” no campo *Posição da legenda*. Caso escolhêssemos “Acima do gráfico”, “À direita” ou “Centro” neste campo, teríamos as expressões `legend.pos="above"`, `legend.pos="topright"` ou `legend.pos="top"` em vez de `legend.pos="topleft"`. Ainda, há outras posições de legenda omitidas no menu de construção de gráficos de barras múltiplas:

- `legend.pos="left"` (*legenda à esquerda, no centro do eixo vertical*);
- `legend.pos="center"` (*legenda no centro, no centro do eixo vertical*);
- `legend.pos="right"` (*legenda à direita, no centro do eixo vertical*);
- `legend.pos="bottomleft"` (*legenda no canto inferior esquerdo*);
- `legend.pos="bottom"` (*legenda no centro, parte inferior*);
- `legend.pos="bottomright"` (*legenda no canto inferior direito*).

Na prática, a posição *bottomleft* não é aconselhável, pois fará a legenda sobrepor-se ao primeiro conjunto de barras. As posições *bottom* e *bottomright* podem ser úteis em gráficos de barras múltiplas na horizontal. Como ilustração, adicione antes do penúltimo “fecha parênteses” da linha de comando original

```
with(Bussab, Barplot(Proced, by=Instrução, style="parallel",  
  legend.pos="topleft", xlab="Proced", ylab="Frequency")) ,
```

a expressão `,horiz=TRUE` e submeta esta linha de comando. O gráfico de barras na horizontal gerado apresenta essencialmente dois problemas:

1. Rótulo do eixo-x (*Proced*) e rótulo do eixo-y (*Frequency*) mantiveram-se no local original, mas precisam trocar de lugar;
2. Legenda se sobrepondo às barras associadas à região de procedência *Outro*.

O primeiro problema pode ser facilmente contornado trocando as expressões `xlab` e `ylab` de posição na linha de comando, conforme explicado na Subseção 4.1.7. Em relação à posição da legenda, note que o canto inferior direito parece propício para posicionar a legenda. Portanto, vamos substituir `legend.pos="topleft"` por `legend.pos="bottomright"` na linha de comando, submetendo-a da seguinte forma:

```
with(Bussab, Barplot(Proced, by=Instrução, style="parallel",  
  legend.pos="bottomright", ylab="Proced", xlab="Frequency",horiz=TRUE)) .
```

Caso ainda ocorra sobreposição da legenda em outras barras, isto pode ser facilmente reparado alongando o eixo das frequências. Primeiramente, atente ao fato de que, neste gráfico, o eixo das frequências é o eixo horizontal (eixo-x), e não mais o eixo vertical (eixo-y). Portanto, tal como mencionado na Subseção 4.1.7, usaremos `xlim` ao invés de `ylim` para alongamento de eixo das frequências. Para esticar o eixo-x (originalmente exibido de 0 até 7) para, digamos, de 0 até 10, adicione a expressão `,xlim=c(0,10)` antes do penúltimo “fecha parênteses” da última linha de comando submetida e submeta-a novamente.

4.5.2 Edições na caixa da legenda

O título da legenda (que, por padrão, consiste no nome da variável escolhida como variável de grupo) pode ser facilmente modificado via linha de comando por meio do comando `legend.title`. No nosso gráfico, o título da legenda é *Instrução*. É possível modificar este título para, digamos, *Nível de instrução*, adicionando a expressão

```
,legend.title="Nível de instrução"
```

antes do penúltimo “fecha parênteses” da linha de comando do gráfico e submetendo-a.

Outras alterações no interior da caixa que comporta a legenda de um gráfico de barras múltiplas não são triviais. Com relação às cores referenciadas na legenda, não há problemas, pois a edição de cores das barras é refletida na legenda. Como exemplo, submeta a linha de comando

```
with(Bussab, Barplot(Proced, by=Instrução, style="parallel", legend.pos=
  "topleft", xlab="Proced", ylab="Frequency", col=c("red", "blue", "black")))
```

(linha de comando tomada como referência nesta Subseção 4.5, com a expressão `,col=c("red", "blue", "black")` adicionada antes do penúltimo “fecha parênteses”) e note que as barras associadas aos indivíduos com ensino fundamental como nível de instrução foram pintadas de vermelho (*red*), e a referência de cor na legenda também é modificada para vermelho de forma automática. O mesmo ocorre para o nível de instrução *Médio* (cor azul) e para o nível de instrução *Superior* (cor preta).

Não há um comando específico para alterar os nomes ao lado de cada quadrado colorido na legenda (no nosso exemplo: *Fundamental*, *Médio* e *Superior*) dentro da linha de comando de gráficos de barras múltiplas. Contudo, podemos contornar isso deslocando a legenda original para tão longe de forma que fique de fora da janela *R Graphics* e, em seguida, “carimbar” uma nova legenda com os nomes desejados. O deslocamento da legenda pode ser feito substituindo a posição da legenda (incluindo as aspas) por um número “grande”. Na nossa linha de comando de referência, isto passa por substituir a expressão `legend.pos="topleft"` por `legend.pos=100` (o número 100 é grande o suficiente para fazer a legenda original “sumir” do gráfico). Sendo assim, submeta a linha de comando

```
with(Bussab, Barplot(Proced, by=Instrução, style="parallel", legend.pos=100,
  xlab="Proced", ylab="Frequency", col=c("red", "blue", "black")))
```

e, logo após, submeta a linha de comando abaixo (em uma linha em branco da janela *R Script*) para que seja exibida uma nova legenda com os nomes *Fundamental*, *Médio* e *Superior* substituídos por *E.F*, *E.M* e *E.S*, respectivamente:

```
legend("topleft",title="Instrução",c("E.F","E.M","E.S"),fill=c("red","blue","black"))
```

A estrutura da linha de comando acima (comando `legend`) apresenta os seguintes argumentos (separados por vírgulas):

- `"topleft"` : posição da legenda (pode ser qualquer posição dentre as apresentadas na Subseção 4.5.1);
- `title="Instrução"` : expressão entre aspas se torna o título da legenda;
- `c("E.F","E.M","E.S")` : vetor com os nomes desejados para substituir os nomes originais, respeitando a ordenação dos atributos da variável;
- `fill=c("red","blue","black")` : cores na legenda (o vetor de cores à direita de `fill=` deve ser o mesmo vetor de cores adicionado na linha de comando associada ao gráfico, à direita de `col=`).

Observação 21 Para executar edições conforme os passos acima em gráficos de barras múltiplas sem edição de cores, é necessário submeter a linha de comando

```
library(colorspace, pos=16)
```

antes de rodar a linha de comando que gera a nova legenda e usar, no lugar de `fill=c("red","blue","black")`, uma expressão com a seguinte estrutura:

```
fill=rainbow_hcl(número_de_atributos_da_variável_de_grupo)
```

No nosso exemplo, a variável *Instrução* é a variável de grupo, portanto a expressão acima deve assumir a forma `fill=rainbow_hcl(3)`, uma vez que a variável *Instrução* possui três atributos: *Fundamental*, *Médio* e *Superior*. Alternativamente, poderíamos usar `fill=rainbow_hcl(length(levels(Bussab$Instrução)))`.

4.6 Diagrama de dispersão

O menu de construção de diagramas de dispersão no R Commander permite uma gama considerável de edições, como exibição da reta de regressão sobreposta aos pontos, formato dos pontos, tamanho das fontes, etc. Porém, ainda há edições pertinentes que não são suportadas pelo menu *Gráficos → Diagrama de dispersão*, como mudança de cor dos pontos e da reta de regressão, espessura da reta de regressão e inibição da grade no fundo do gráfico.

Como linha de comando de referência para as edições em diagramas de dispersão, tomemos um diagrama de dispersão entre as variáveis *Idade* e *Renda* (do conjunto de dados *Bussab*) com reta de regressão sobreposta. Portanto, acesse o menu *Gráficos → Diagrama de dispersão*, escolha *Idade* como variável-x e *Renda* como variável-y. Na aba *Opções*, marque a caixinha *Linha de quadrados mínimos* e clique em OK para gerar, além do gráfico solicitado, a linha de comando

```
scatterplot(Renda~Idade, regLine=TRUE, smooth=FALSE, boxplots=FALSE,  
            data=Bussab)
```

na janela *R Script*.

4.6.1 Edição de cores

A inclusão de uma expressão do tipo `,col="cor_de_sua_preferência"` antes do último “fecha parênteses” na linha de comando de um diagrama de dispersão faz com que tanto os pontos como a reta de regressão fiquem com a cor de sua preferência após submissão da linha de comando modificada. Por exemplo, submeter a linha de comando

```
scatterplot(Renda~Idade, regLine=TRUE, smooth=FALSE, boxplots=FALSE,  
            data=Bussab,col="black")
```

resulta em um diagrama de dispersão com pontos e reta de regressão na cor preta. Cabe ressaltar que a forma padrão de pontos em diagramas de dispersão no R Commander consiste em pequenas circunferências (pois mantivemos o campo *Caracteres do gráfico* em *<auto>* na janela de construção do diagrama de dispersão). Portanto, a edição de cores como acima colore estas circunferências, e não todo o círculo. Uma forma de ter todo o

círculo preenchido pela cor escolhida é adicionar a expressão `,pch=19` antes do último “fecha parênteses” da linha de comando, conforme segue abaixo, e submetê-la logo após:

```
scatterplot(Renda~Idade, regLine=TRUE, smooth=FALSE, boxplots=FALSE,  
            data=Bussab,col="black",pch=19)
```

Como já destacado nas edições acima, tanto os pontos como a reta de regressão são coloridos pela mesma cor. Para definir uma cor para a reta de regressão diferente da cor dos pontos, é necessário substituir a expressão `regLine=TRUE` por uma expressão do tipo

```
regLine=list(col="cor_de_sua_preferência")
```

(atente-se ao L maiúsculo em `regLine`). Como exemplo, vamos recuperar a cor original da reta de regressão submetendo a seguinte linha de comando na janela *R Script*:

```
scatterplot(Renda~Idade, regLine=list(col="blue"), smooth=FALSE,  
            boxplots=FALSE, data=Bussab,col="black")
```

O diagrama de dispersão gerado possui reta de regressão azul (*blue*) e pontos pretos (*black*).

4.6.2 Espessura da reta de regressão

Vimos na subseção anterior como modificar a cor da reta de regressão. Além disto, também é possível modificar a sua espessura substituindo a expressão `regLine=TRUE` por uma expressão do tipo

```
regLine=list(lwd=número_de_sua_preferência)
```

(novamente, atente-se ao L maiúsculo em `regLine`). A espessura padrão é dada por `lwd=2`. Portanto, qualquer número maior que 2 fará com que a reta de regressão fique mais espessa, e qualquer número (positivo) menor que 2 fará com que a reta de regressão fique menos espessa. Submeta a linha de comando de referência utilizada nesta Subseção 4.6 e, logo após, submeta a linha de comando abaixo para verificar a diferença:

```
scatterplot(Renda~Idade, regLine=list(lwd=3), smooth=FALSE,  
            boxplots=FALSE, data=Bussab)
```

Observação 22 É possível modificar simultaneamente a cor e a espessura da reta de regressão ao substituir a expressão `regLine=TRUE` por uma expressão do tipo

```
regLine=list(col="cor_de_sua_preferência", lwd=número_de_sua_preferência).
```

Por exemplo, submeter a linha de comando

```
scatterplot(Renda~Idade, regLine=list(col="brown",lwd=1), smooth=FALSE,  
  boxplots=FALSE, data=Bussab)
```

retorna um diagrama de dispersão com reta de regressão na cor marrom (brown) e menos espessa que o padrão.

4.6.3 Inibição da grade (*grid*) no fundo do gráfico

A submissão da linha de comando de um diagrama de dispersão com a expressão `,grid=FALSE` (ou `,grid=F`) adicionada antes do último “fecha parênteses” desta linha de comando faz com que o diagrama de dispersão gerado não apresente a grade de fundo. Tal grade é útil como referência de valores tanto para a variável que está no eixo horizontal (eixo-x) como para a variável que está no eixo vertical (eixo-y). Entretanto, em algumas situações práticas, tal grade pode gerar certa “poluição visual”, sendo nestes casos razoável sua inibição por meio da edição aqui apresentada. Para visualização do resultado, submeta a linha de comando

```
scatterplot(Renda~Idade, regLine=TRUE, smooth=FALSE, boxplots=FALSE,  
  data=Bussab,grid=FALSE)
```

4.6.4 Edições em diagramas de dispersão por grupos

Para edições em diagramas de dispersão por grupos, continuaremos tomando como referência o diagrama de dispersão entre as variáveis *Idade* e *Renda*, porém com a variável *Casado_fator* como variável de grupo. Portanto, acesse o menu [Gráficos → Diagrama de dispersão](#), escolha *Idade* como variável-x, *Renda* como variável-y e clique na caixa *Gráfico*

por grupos para escolher a variável *Casado_fator* como variável de grupo (mantenha a caixinha *Gráfico de linhas por grupo* marcada e dê OK). Na aba *Opções*, marque a caixinha *Linha de quadrados mínimos*, caso não esteja marcada. Clique em OK para gerar, além do gráfico solicitado, a linha de comando

```
scatterplot(Renda~Idade | Casado_fator, regLine=TRUE, smooth=FALSE,
            boxplots=FALSE, by.groups=TRUE, data=Bussab)
```

na janela *R Script*. Note que agora temos duas retas de regressão: a reta na cor azul representa a reta de regressão sobreposta ao diagrama de dispersão entre idades e rendas apenas dos indivíduos que são casados; e, analogamente, a reta na cor magenta representa a reta de regressão sobreposta ao diagrama de dispersão entre idades e rendas apenas dos indivíduos que não são casados. Em diagramas de dispersão por grupos, é razoável que a reta de regressão associada a cada grupo seja da mesma cor dos pontos que representam as observações do respectivo grupo. Neste exemplo, cada círculo com contorno na cor azul representa um indivíduo casado, e cada triângulo com contorno na cor magenta representa um indivíduo que não é casado.

Em diagramas de dispersão por grupos, a **edição de cores para cada grupo** (pontos e respectiva reta de regressão) é realizada por meio de um vetor de cores cujo número de cores é o número de atributos da variável de grupo. No caso da variável *Casado_fator*, há dois atributos: *Sim* e *Não*, configurados nesta ordem (basta checar na legenda do gráfico). Portanto, para a edição de cores neste gráfico, é necessário incluir uma expressão do tipo

```
,col=c("cor_para_o_atributo_Sim","cor_para_o_atributo_Não")
```

antes do último “fecha parênteses” da linha de comando tomada como referência nesta subseção e submetê-la em seguida. Por exemplo, a submissão da linha de comando abaixo faz com que os pontos e reta de regressão dos indivíduos casados fiquem na cor ciano (*cyan*) e os pontos e reta de regressão dos indivíduos não casados fiquem na cor dourado (*gold*):

```
scatterplot(Renda~Idade | Casado_fator, regLine=TRUE, smooth=FALSE,
            boxplots=FALSE, by.groups=TRUE, data=Bussab,col=c("cyan","gold"))
```

Para que círculos e triângulos **sejam preenchidos pelas cores definidas**, basta adicionar a expressão `,pch=c(19,17)` antes do último “fecha parênteses” da linha de

comando acima e submetê-la. Alternativamente, isto não seria necessário se, no menu de construção do diagrama de dispersão por grupos, fosse digitado `19,17` no lugar de `<auto>` no campo *Caracteres do gráfico*, pois isto incluiria a expressão `,pch=c(19,17)` na linha de comando de referência.

Para ***modificar a espessura das retas de regressão de cada grupo***, basta proceder conforme a Subseção 4.6.2, independente do número de retas de regressão que o diagrama de dispersão por grupos possua (só é possível definir uma espessura, que será a mesma para todas as retas de regressão).

O ***título da legenda de um diagrama de dispersão por grupos*** no R Commander pode ser modificado via linha de comando, da seguinte forma: adicione, antes do último “fecha parênteses” da linha de comando de um diagrama de dispersão por grupos, uma expressão do tipo

```
,legend=list(title="Título_de_sua_preferência")
```

e submeta a linha de comando modificada. A título de ilustração, submeta a linha de comando

```
scatterplot(Renda~Idade | Casado_fator, regLine=TRUE, smooth=FALSE,
  boxplots=FALSE, by.groups=TRUE, data=Bussab,legend=list(title="É casado?"))
```

Isto gera um diagrama de dispersão por grupos entre as variáveis *Idade* e *Renda* com *Casado_fator* como variável de grupo, porém o título da legenda deixa de ser o nome da variável de grupo e passa a ser: “*É casado?*”.

4.6.5 [Bônus] Matriz de correlação em um determinado grupo

Apesar do menu *Gráficos → Diagrama de dispersão* do R Commander permitir a construção de diagramas de dispersão por grupos, não há menu no R Commander que retorne o coeficiente de correlação entre duas variáveis quantitativas apenas dentro de algum grupo (definido por algum atributo de uma variável qualitativa do mesmo conjunto de dados). Entretanto, é possível obter informações deste tipo editando linhas de comando associadas à matrizes de correlação. Primeiramente, acesse o menu *Estatísticas → Resumos → Matriz de correlação* e escolha as variáveis de interesse. Por conveniência, escolha apenas *Idade* e *Renda* (com auxílio da tecla `Ctrl`) e clique em OK. A linha de comando

```
cor(Bussab[,c("Idade", "Renda")], use="complete")
```

é gerada na janela *R Script* e o coeficiente de correlação entre *Idade* e *Renda* de todos os indivíduos do conjunto de dados *Bussab* é exibido na janela *Output*: 0,3753518.

Editando a linha de comando acima, é possível determinar, por exemplo, o coeficiente de correlação entre idades e rendas apenas dos indivíduos casados (isto é, os que apresentam atributo *Sim* na variável *Casado_fator*). Para isto, inclua, entre o “abre colchetes” e a primeira vírgula da referida linha de comando, a expressão

```
which(Bussab$Casado_fator=="Sim")
```

e submeta esta linha de comando modificada, a qual ficará da seguinte forma:

```
cor(Bussab[which(Bussab$Casado_fator=="Sim"),c("Idade", "Renda")],  
     use="complete")
```

De maneira análoga, é possível determinar o coeficiente de correlação entre idades e rendas apenas dos indivíduos que não são casados (isto é, os que apresentam atributo *Não* na variável *Casado_fator*) por meio da submissão da seguinte linha de comando:

```
cor(Bussab[which(Bussab$Casado_fator=="Não"),c("Idade", "Renda")],  
     use="complete")
```

No caso de outros conjuntos de dados e/ou outras variáveis, a estrutura da expressão incluída entre o “abre colchetes” e a primeira vírgula da linha de comando original da matriz de correlação é sempre da forma

```
which(Nome_do_conjunto_de_dados$Nome_da_variável_de_grupo=="Nome_do_atributo").
```

Referências

- [1] Ribeiro Melo, F.R. (2019). *Introdução ao R Commander - Notas de Aula*. Disponível em: <http://gae.uniriotec.br/7/material.html>
- [2] Ajuda no R - Boxplots (comando `boxplot`). Disponível em: <https://www.rdocumentation.org/packages/graphics/versions/3.5.2/topics/boxplot>
- [3] Ajuda no R - Boxplots (comando `Boxplot`). Disponível em: <https://www.rdocumentation.org/packages/car/versions/3.0-2/topics/Boxplot>
- [4] Ajuda no R - Diagramas de dispersão (comando `scatterplot`). Disponível em: <https://www.rdocumentation.org/packages/car/versions/3.0-2/topics/scatterplot>
- [5] Ajuda no R - Gráficos de barras (comando `barplot`). Disponível em: <https://www.rdocumentation.org/packages/graphics/versions/3.5.2/topics/barplot>
- [6] Ajuda no R - Gráficos de barras (comando `Barplot`). Disponível em: <https://www.rdocumentation.org/packages/RcmdrMisc/versions/2.5-1/topics/Barplot>
- [7] Ajuda no R - Gráficos de pizza (comando `pie`). Disponível em: <https://www.rdocumentation.org/packages/graphics/versions/3.5.2/topics/pie>
- [8] Ajuda no R - Histogramas (comando `hist`). Disponível em: <https://www.rdocumentation.org/packages/graphics/versions/3.5.2/topics/hist>
- [9] Ajuda no R - Histogramas (comando `Hist`). Disponível em: <https://www.rdocumentation.org/packages/RcmdrMisc/versions/2.5-1/topics/Hist>
- [10] Pie Chart with Annotated Percentages. Disponível em: <http://www.statmethods.net/graphs/pie.html>
- [11] R FAQ “How can I add features or dimensions to my bar plot?”. Institute for Digital Research and Education (UCLA). Disponível em: <https://stats.idre.ucla.edu/r/faq/how-can-i-add-features-or-dimensions-to-my-bar-plot/>

Anexo I – Lista de cores disponíveis no R – página 1 de 11

	aliceblue	antiquewhite	antiquewhite1
antiquewhite2	antiquewhite3	antiquewhite4	aquamarine
aquamarine1	aquamarine2	aquamarine3	aquamarine4
azure	azure1	azure2	azure3
azure4	beige	bisque	bisque1
bisque2	bisque3	bisque4	black
blanchedalmond	blue	blue1	blue2
blue3	blue4	blueviolet	brown
brown1	brown2	brown3	brown4
burlywood	burlywood1	burlywood2	burlywood3
burlywood4	cadetblue	cadetblue1	cadetblue2
cadetblue3	cadetblue4	chartreuse	chartreuse1
chartreuse2	chartreuse3	chartreuse4	chocolate
chocolate1	chocolate2	chocolate3	chocolate4
coral	coral1	coral2	coral3

Anexo I – Lista de cores disponíveis no R – página 2 de 11

coral4	cornflowerblue	cornsilk	cornsilk1
cornsilk2	cornsilk3	cornsilk4	cyan
cyan1	cyan2	cyan3	cyan4
darkblue	darkcyan	darkgoldenrod	darkgoldenrod1
darkgoldenrod2	darkgoldenrod3	darkgoldenrod4	darkgray
darkgreen	darkgrey	darkkhaki	darkmagenta
darkolivegreen	darkolivegreen1	darkolivegreen2	darkolivegreen3
darkolivegreen4	darkorange	darkorange1	darkorange2
darkorange3	darkorange4	darkorchid	darkorchid1
darkorchid2	darkorchid3	darkorchid4	darkred
darksalmon	darkseagreen	darkseagreen1	darkseagreen2
darkseagreen3	darkseagreen4	darkslateblue	darkslategray
darkslategray1	darkslategray2	darkslategray3	darkslategray4
darkslategrey	darkturquoise	darkviolet	deeppink
deeppink1	deeppink2	deeppink3	deeppink4

Anexo I – Lista de cores disponíveis no R – página 3 de 11

deepskyblue	deepskyblue1	deepskyblue2	deepskyblue3
deepskyblue4	dimgray	dimgrey	dodgerblue
dodgerblue1	dodgerblue2	dodgerblue3	dodgerblue4
firebrick	firebrick1	firebrick2	firebrick3
firebrick4	floralwhite	forestgreen	gainsboro
ghostwhite	gold	gold1	gold2
gold3	gold4	goldenrod	goldenrod1
goldenrod2	goldenrod3	goldenrod4	gray
gray0	gray1	gray2	gray3
gray4	gray5	gray6	gray7
gray8	gray9	gray10	gray11
gray12	gray13	gray14	gray15
gray16	gray17	gray18	gray19
gray20	gray21	gray22	gray23
gray24	gray25	gray26	gray27

Anexo I – Lista de cores disponíveis no R – página 4 de 11

gray28	gray29	gray30	gray31
gray32	gray33	gray34	gray35
gray36	gray37	gray38	gray39
gray40	gray41	gray42	gray43
gray44	gray45	gray46	gray47
gray48	gray49	gray50	gray51
gray52	gray53	gray54	gray55
gray56	gray57	gray58	gray59
gray60	gray61	gray62	gray63
gray64	gray65	gray66	gray67
gray68	gray69	gray70	gray71
gray72	gray73	gray74	gray75
gray76	gray77	gray78	gray79
gray80	gray81	gray82	gray83
gray84	gray85	gray86	gray87

Anexo I – Lista de cores disponíveis no R – página 5 de 11

gray88	gray89	gray90	gray91
gray92	gray93	gray94	gray95
gray96	gray97	gray98	gray99
	green	green1	green2
green3	green4	greenyellow	grey
grey0	grey1	grey2	grey3
grey4	grey5	grey6	grey7
grey8	grey9	grey10	grey11
grey12	grey13	grey14	grey15
grey16	grey17	grey18	grey19
grey20	grey21	grey22	grey23
grey24	grey25	grey26	grey27
grey28	grey29	grey30	grey31
grey32	grey33	grey34	grey35
grey36	grey37	grey38	grey39

Anexo I – Lista de cores disponíveis no R – página 6 de 11

grey40	grey41	grey42	grey43
grey44	grey45	grey46	grey47
grey48	grey49	grey50	grey51
grey52	grey53	grey54	grey55
grey56	grey57	grey58	grey59
grey60	grey61	grey62	grey63
grey64	grey65	grey66	grey67
grey68	grey69	grey70	grey71
grey72	grey73	grey74	grey75
grey76	grey77	grey78	grey79
grey80	grey81	grey82	grey83
grey84	grey85	grey86	grey87
grey88	grey89	grey90	grey91
grey92	grey93	grey94	grey95
grey96	grey97	grey98	grey99

Anexo I – Lista de cores disponíveis no R – página 7 de 11

	honeydew	honeydew1	honeydew2
honeydew3	honeydew4	hotpink	hotpink1
hotpink2	hotpink3	hotpink4	indianred
indianred1	indianred2	indianred3	indianred4
ivory	ivory1	ivory2	ivory3
ivory4	khaki	khaki1	khaki2
khaki3	khaki4	lavender	lavenderblush
lavenderblush1	lavenderblush2	lavenderblush3	lavenderblush4
lawngreen	lemonchiffon	lemonchiffon1	lemonchiffon2
lemonchiffon3	lemonchiffon4	lightblue	lightblue1
lightblue2	lightblue3	lightblue4	lightcoral
lightcyan	lightcyan1	lightcyan2	lightcyan3
lightcyan4	lightgoldenrod	lightgoldenrod1	lightgoldenrod2
lightgoldenrod3	lightgoldenrod4	lightgoldenrodyellow	lightgray
lightgreen	lightgrey	lightpink	lightpink1

Anexo I – Lista de cores disponíveis no R – página 8 de 11

lightpink2	lightpink3	lightpink4	lightsalmon
lightsalmon1	lightsalmon2	lightsalmon3	lightsalmon4
lightseagreen	lightskyblue	lightskyblue1	lightskyblue2
lightskyblue3	lightskyblue4	lightslateblue	lightslategray
lightslategray	lightsteelblue	lightsteelblue1	lightsteelblue2
lightsteelblue3	lightsteelblue4	lightyellow	lightyellow1
lightyellow2	lightyellow3	lightyellow4	limegreen
linen	magenta	magenta1	magenta2
magenta3	magenta4	maroon	maroon1
maroon2	maroon3	maroon4	mediumaquamarine
mediumblue	mediumorchid	mediumorchid1	mediumorchid2
mediumorchid3	mediumorchid4	mediumpurple	mediumpurple1
mediumpurple2	mediumpurple3	mediumpurple4	mediumseagreen
mediumslateblue	mediumspringgreen	mediumturquoise	mediumvioletred
midnightblue	mintcream	mistyrose	mistyrose1

Anexo I – Lista de cores disponíveis no R – página 9 de 11

mistyrose2	mistyrose3	mistyrose4	moccasin
navajowhite	navajowhite1	navajowhite2	navajowhite3
navajowhite4	navy	navyblue	oldlace
olivedrab	olivedrab1	olivedrab2	olivedrab3
olivedrab4	orange	orange1	orange2
orange3	orange4	orangered	orangered1
orangered2	orangered3	orangered4	orchid
orchid1	orchid2	orchid3	orchid4
palegoldenrod	palegreen	palegreen1	palegreen2
palegreen3	palegreen4	paleturquoise	paleturquoise1
paleturquoise2	paleturquoise3	paleturquoise4	palevioletred
palevioletred1	palevioletred2	palevioletred3	palevioletred4
papayawhip	peachpuff	peachpuff1	peachpuff2
peachpuff3	peachpuff4	peru	pink
pink1	pink2	pink3	pink4

Anexo I – Lista de cores disponíveis no R – página 10 de 11

plum	plum1	plum2	plum3
plum4	powderblue	purple	purple1
purple2	purple3	purple4	red
red1	red2	red3	red4
rosybrown	rosybrown1	rosybrown2	rosybrown3
rosybrown4	royalblue	royalblue1	royalblue2
royalblue3	royalblue4	saddlebrown	salmon
salmon1	salmon2	salmon3	salmon4
sandybrown	seagreen	seagreen1	seagreen2
seagreen3	seagreen4	seashell	seashell1
seashell2	seashell3	seashell4	sienna
sienna1	sienna2	sienna3	sienna4
skyblue	skyblue1	skyblue2	skyblue3
skyblue4	slateblue	slateblue1	slateblue2
slateblue3	slateblue4	slategray	slategray1

Anexo I – Lista de cores disponíveis no R – página 11 de 11

slategray2	slategray3	slategray4	slategrey
snow	snow1	snow2	snow3
snow4	springgreen	springgreen1	springgreen2
springgreen3	springgreen4	steelblue	steelblue1
steelblue2	steelblue3	steelblue4	tan
tan1	tan2	tan3	tan4
thistle	thistle1	thistle2	thistle3
thistle4	tomato	tomato1	tomato2
tomato3	tomato4	turquoise	turquoise1
turquoise2	turquoise3	turquoise4	violet
violetred	violetred1	violetred2	violetred3
violetred4	wheat	wheat1	wheat2
wheat3	wheat4	whitesmoke	yellow
yellow1	yellow2	yellow3	yellow4
yellowgreen			