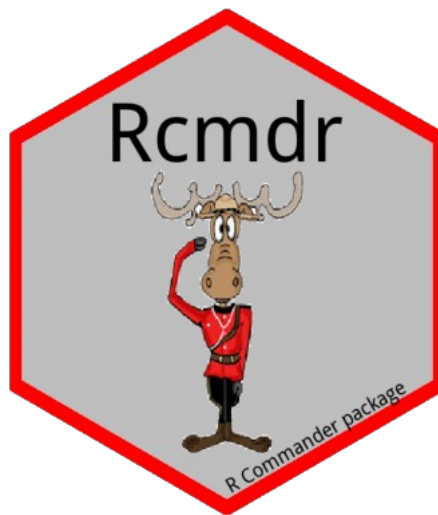


R Commander: um pouco além dos menus gráficos

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



Felipe Rafael Ribeiro Melo
Departamento de Métodos Quantitativos
Universidade Federal do Estado do Rio de Janeiro - UNIRIO

Rio de Janeiro, Brasil
Setembro de 2024

Sumário

1	Introdução	4
2	Linhas de comando	5
2.1	Submetendo linha de comando	6
2.2	Submetendo linha de comando modificada	7
3	Cores disponíveis no R	8
4	Modificando a estrutura de gráficos gerados pelo R Commander	11
4.1	Gráfico de barras simples	11
4.1.1	Mudando a cor das barras: apenas uma cor	12
4.1.2	Mudando a cor das barras: duas ou mais cores	12
4.1.3	Espaçamento entre as barras	13
4.1.4	Editando nomes abaixo das barras	13
4.1.5	Tamanho das fontes	14
4.1.6	Alongando o eixo vertical	15
4.1.7	Gráfico de barras na horizontal	15
4.1.8	Frequências e porcentagens dos atributos nas barras	16
4.2	Gráfico de pizza	18
4.2.1	Mudando as cores das “fatias”	19
4.2.2	Tamanho do gráfico de pizza	19
4.2.3	Tamanho da fonte: nomes dos atributos	20
4.2.4	Editando nomes dos atributos ao lado das fatias	20
4.2.5	Porcentagens em gráficos de pizza	21
4.3	Histograma	23
4.3.1	Modificando cores das barras	23
4.3.2	Tamanho das fontes	24
4.3.3	Alongando o eixo vertical	25
4.3.4	Listar os valores que delimitam as classes	25
4.3.5	Modificar os valores que delimitam cada classe	26
4.3.6	Frequências ou porcentagens nas barras de um histograma	28
4.3.7	Polígono de frequências	29

4.3.8	Histograma de um único grupo do conjunto de dados	30
4.4	Boxplot	32
4.4.1	Mudando a cor da caixa e do traçado do boxplot	32
4.4.2	Mudando as cores da caixas e dos traçados de boxplots em grupos .	33
4.4.3	Editando nomes dos atributos em boxplots por grupos	33
4.4.4	Tamanho das fontes	34
4.4.5	Alongando (ou encurtando) o eixo vertical	35
4.4.6	Caixa do boxplot “amassada” na altura da mediana	35
4.4.7	Valores discrepantes indicados por círculos preenchidos	36
4.4.8	Valores discrepantes identificados pelos respectivos valores da variável	36
4.5	Gráfico de barras múltiplas	38
4.5.1	Posição da legenda	39
4.5.2	Edições na caixa da legenda	40
4.6	Diagrama de dispersão	42
4.6.1	Edição de cores	43
4.6.2	Espessura da reta de regressão	44
4.6.3	Inibição da grade (<i>grid</i>) no fundo do gráfico	45
4.6.4	Edições em diagramas de dispersão por grupos	45
4.6.5	[<i>Bônus</i>] Matriz de correlação em um determinado grupo	47
	Referências	49
	Anexo I - Lista de cores disponíveis no R	52

1 Introdução

Em diversas áreas do conhecimento, análises estatísticas em conjuntos de dados tem sido cada vez mais utilizadas, com *softwares* específicos lidando com toda a parte computacional de tais análises. Neste contexto, o *software* R (R Core Team, 2024) tem ganhando destaque nos últimos anos, sobretudo no ambiente acadêmico, em virtude de suas diversas vantagens, como ser gratuito, de código aberto, bastante flexível, apoiado por uma grande equipe de desenvolvedores em todo o mundo e frequentemente atualizado com novos pacotes e funcionalidades. Entretanto, este *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 (Fox; Bouchet-Valat, 2024), 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* (Melo, 2024), em particular a sua quarta seção, que trata de representações gráficas. Apesar dos menus de construção de gráficos no R Commander possuírem diversos campos que permitem editar¹ a saída gráfica, estas 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 menu *Gráficos → Gradiente de cores (color palette)*? A resposta é **Sim**, e o principal objetivo deste material é mostrar como fazer estas e algumas outras edições em gráficos gerados pelo R Commander que não são suportadas nos seus respectivos menus de construção. Ou seja, vamos além dos menus geradores de gráficos do R Commander.


Por comodidade, **usamos ao longo de todo este material apenas o conjunto de dados *Bussab* utilizado na Seção 4 em (Melo, 2024)**, na forma como o mesmo está disposto ao início desta seção, ou seja, com suas seis colunas originais e cinco colunas adicionadas na Subseção 3.8 em (Melo, 2024). **Para baixar o conjunto de dados *Bussab***

¹O verbo *editar* e suas conjugações, bem como o substantivo *edição* e suas variações, são empregados ao longo deste material no sentido de “preparar ou alterar para publicação”.

“**atualizado**”, acesse github.com/feliperafaelrm/Rcmdr/raw/main/Bussab.RData, de modo a não ser necessário realizar (ou refazer) a importação do conjunto de dados presente no arquivo *Bussab.xlsx* e modificações de variáveis no conjunto de dados importado.

2 Linhas de comando

Cada comando executado no R Commander por meio de menus gera, na janela *R Script* do R Commander, uma 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.

Para começarmos a explorar esse assunto, **primeiramente faça o download do arquivo *Bussab.RData*** no [link](#) exposto nas últimas linhas da Seção 1. Abra o R, digite **require(Rcmdr)** ou **library(Rcmdr)** na janela *R Console* e aperte  para abrir a janela *R Commander*. Uma vez nesta janela, siga os passos abaixo:

1. Carregue o conjunto de dados *Bussab* acessando o menu **Dados → Carregar conjunto de dados** e procurando pelo arquivo *Bussab.RData*.
2. Acesse o menu **Gráficos → Gráfico de Barras**, selecione *Proced* e dê OK.
3. Acesse o menu **Gráficos → Gráfico de Pizza**, selecione *Instrução*, digite *Distribuição do nível de instrução* no campo *Título do gráfico* e clique em OK.

A Figura 1 mostra como fica a janela *R Commander* após a realização dos passos acima. Foque na janela *R Script*, na qual 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 do pacote *colorspace*, o qual fornece a paleta de cores padrão para as “fatias” de gráficos de pizza gerados no R Commander, e esta linha é escrita assim que clicamos no menu para geração de gráficos de pizza pela primeira vez após carregarmos o R Commander. Portanto, as três

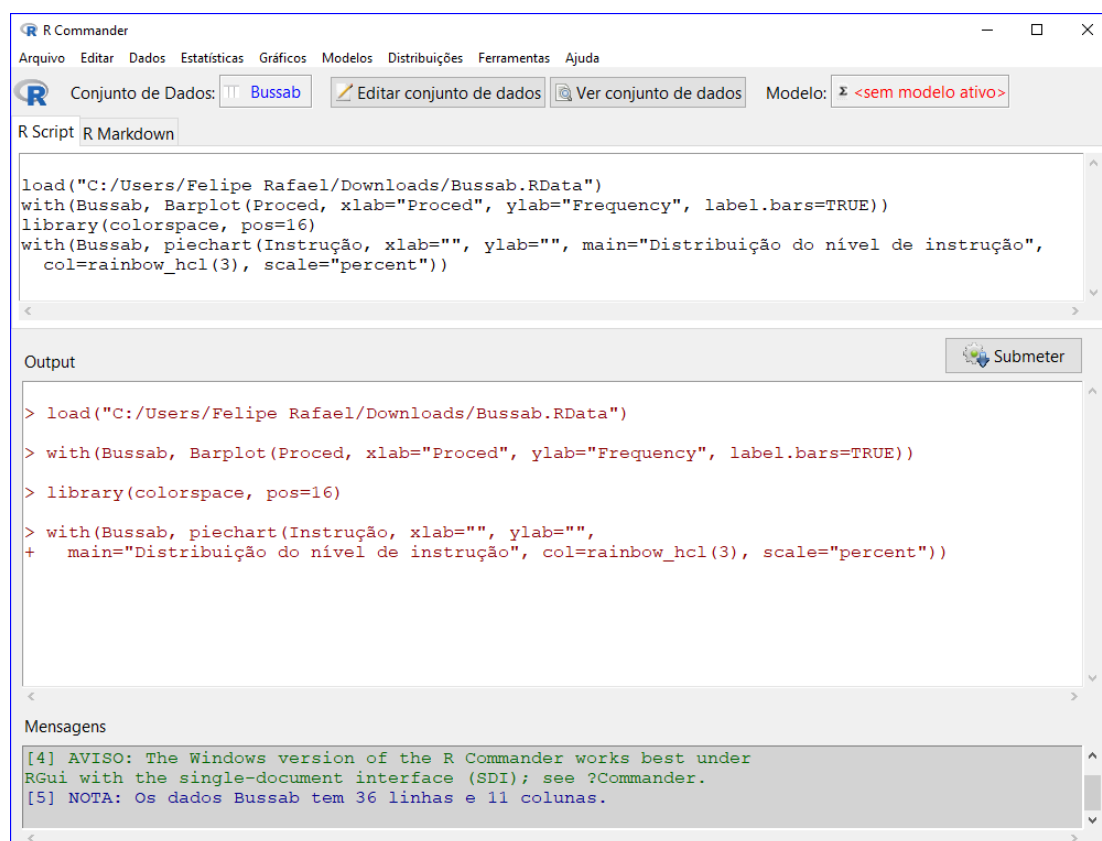


Figura 1: Apresentando linhas de comando na janela *R Script* do R Commander.

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”,

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

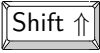


foi gerada ao executarmos o Passo 2 (geração do gráfico de barras da variável *Proced*), e

```
load("C:/Users/Felipe Rafael/Downloads/Bussab.RData")
```

foi gerada ao executarmos o Passo 1 (carregamento do conjunto de dados *Bussab*, presente no arquivo *Bussab.RData*).



2.1 Submetendo linha de comando


As linhas de comando exibidas na janela *R Script* do R Commander não são apenas “decorações”. Podemos **submeter** 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.

Para **submeter** (ou **executar**) uma linha de comando na janela *R Script*, selecione toda a linha de comando com o *mouse* (ou com as setas do teclado, mantendo a tecla  pressionada) e clique no botão *Submeter* (localizado entre as janelas *R Script* e *Output*, do lado direito) ou use o atalho do teclado  + . Ainda, é possível submeter duas ou mais linhas de comando na janela *R Script* de uma só vez, selecionando todas elas e clicando em *Submeter*.

Acessando a janela *R Gui*, observe 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, faça o procedimento acima com a linha de comando que vai de `with(Bussab, Barplot` até `label.bars=TRUE))`. 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, piechart` até `scale="percent"))` com o *mouse* e clicando em *Submeter* logo após.

Observação 1. Quando a linha de comando que desejamos submeter 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, clicando com o *mouse* em qualquer lugar desta linha ou conduzindo o cursor até ela com as setas do teclado. Ainda com as setas do teclado e as teclas  e/ou , podemos fazer com que uma linha de comando que inicialmente ocupe duas ou mais linhas da janela *R Script* passe a ocupar uma linha só nesta janela, e dessa forma ela pode ser submetida (com ou sem alterações) apenas com o cursor sobre ela.

Observação 2. Submeter linha de comando na janela *R Script* retorna o mesmo resultado que digitar ou colar linha de comando na janela *R Console* e apertar .

2.2 Submetendo linha de comando modificada

A grande motivação em submeter linhas de comando exibidas na janela *R Script* reside no fato de **podermos modificá-las antes de submetê-las**. Observe, por exemplo, a linha de comando associada ao gráfico de barras da variável *Proced*. Nela, está

escrito `ylab="Frequency"`, que define o rótulo do eixo-y. Substituindo `"Frequency"` por `"Frequência"` e submetendo esta linha de comando, surge um novo gráfico na janela *R Graphics*, com a palavra *Frequência* escrita no rótulo do eixo-y em vez de *Frequency*.

Em particular, a modificação acima também poderia ser feita no 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 gráficos de barras. Isto procede não apenas para gráficos de barras, mas para vários resultados que o R Commander fornece por meio de seus menus. Em particular, estamos interessados em editar saídas gráficas, e isto é 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**, apenas selecionando essa variável e clicando em OK. A linha de comando


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

é impressa na janela *R Script*. Atente ao termo `col="darkgray"` na linha acima, no qual *col* se refere a cor (uma abreviação de *color*) e *darkgray* significa cinza escuro, que é a cor padrão 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, o resultado é um histograma com barras de outra cor (seguindo o exemplo: barras vermelhas).

Mas como saber quais cores o R disponibiliza?

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

```
colors()
```

e aperte  para que os nomes das 657 cores disponíveis no R por algum nome sejam exibidos na janela *R Console*. Tal lista também segue no Anexo I (página 52), com o nome de cada cor escrito com a respectiva cor.

Observação 3. Também é possível visualizar os nomes de todas as cores disponíveis no R 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()`, em consonância com as Observações 1 e 2. Dessa forma, os nomes das cores são impressos na janela *Output* do R Commander.

As cores configuradas no menu **Gráficos → Gradiente de cores (color palette)** podem ser utilizadas por meio de rotulação numérica. A cor do primeiro retângulo da paleta é rotulada pelo número 1, a cor do segundo retângulo da paleta é rotulada pelo número 2, e assim por diante, até a cor do oitavo (e último) retângulo da paleta, rotulada pelo número 8. Além disso, o número 0 rotula a cor branca. A Figura 2 exibe a paleta padrão. Observe que, no interior de cada retângulo colorido, há um código composto por uma trilha seguida de seis dígitos alfanuméricos, que representa o padrão hexadecimal (código HEX) da respectiva cor, que descreve fielmente a cor de cada retângulo. Os nomes abaixo de cada retângulo fazem parte das 657 cores listadas pela execução de `colors()`, mas não necessariamente representam a cor exata do retângulo. Dito isto e considerando a paleta padrão conforme a Figura 2, sem qualquer tipo de modificação:

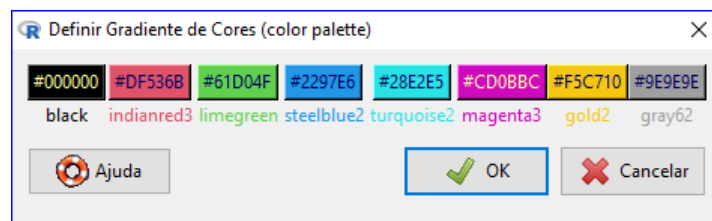


Figura 2: Gradiente de cores original do R Commander.

- 0 equivale a #FFFFFF ("white"),
- 1 equivale a #000000 ("black"),
- 2 equivale a #DF536B (algo próximo a "indianred3"),
- 3 equivale a #61D04F (algo próximo a "limegreen"),
- 4 equivale a #2297E6 (algo próximo a "steelblue2"),
- 5 equivale a #28E2E5 (algo próximo a "turquoise2"),
- 6 equivale a #CD0BBC (algo próximo a "magenta3"),
- 7 equivale a #F5C710 (algo próximo a "gold2"),
- 8 equivale a #9E9E9E ("gray62").

Da lista acima, concluímos que, do segundo até o sétimo retângulo colorido da Figura 2, o nome abaixo de cada um deles é uma “aproximação” da cor exata do respectivo retângulo. Por exemplo, as expressões

```
col="indianred3"
```

e

```
col=2
```

retornam resultados ligeiramente diferentes dentro de uma linha de comando no tocante à cor. O mesmo vale para vetor de cores (um conceito definido com maior clareza na Subseção 4.1.2): por exemplo, as expressões

```
col=c("steelblue2","gold2")
```

e

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

geram resultados ligeiramente diferentes dentro de uma linha de comando no tocante à cor. A título de ilustração e tomando por base a linha de comando explicitada no começo desta seção, associada ao histograma da variável *Renda*, submeta as duas linhas de comando abaixo (uma de cada vez) e veja a ligeira diferença das cores de uma figura para a outra.

```
with(Bussab, Hist(Renda, scale="frequency", breaks="Sturges",
  col=c("steelblue2","gold2")))
with(Bussab, Hist(Renda, scale="frequency", breaks="Sturges", col=c(4,7)))
```

Observação 4. A paleta de cores padrão do R a partir de sua versão 4.0.0 é denominada "R4". Nomes de outras paletas pré-configuradas podem ser verificados ao digitar e submeter, na janela *R Script*, a linha de comando `palette.pals()`. Para mudar a paleta tomada como padrão (por exemplo, de "R4" para "R3"), digite e submeta, na janela *R Script*, `palette("R3")`. Para conferir a nova paleta escolhida, acesse o menu *Gráficos → Gradiente de cores (color palette)*. A rotulação numérica para cores passa a seguir a nova paleta escolhida. Em particular, a paleta "R3" (padrão do R antes da versão 4.0.0) traz os nomes abaixo dos retângulos representando fielmente as cores dos mesmos.

4 Modificando a estrutura de gráficos gerados pelo R Commander

Nesta seção, que se alonga até o encerramento deste material em diversas subseções, vamos modificar a estrutura original de gráficos gerados pelo R Commander. Ou seja, após gerar um gráfico em algum submenu específico do menu **Gráficos**, **editamos 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 aborda 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 no começo das subseções supracitadas 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 seguem válidas.

De forma a acompanhar, de forma integral, as subseções da Seção 4, admitimos que o(a) leitor(a) deste material esteja ciente que:

- é necessário carregar o arquivo *Bussab.RData* e, conseqüentemente, ter o conjunto de dados *Bussab* como conjunto de dados ativo;
- qualquer sugestão de modificação de linha de comando deve ser feita na janela *R Script* do R Commander;
- linhas de comando explicitadas neste material com sugestão de submissão devem ser copiadas daqui e coladas em linha(s) em branco da janela *R Script*;
- cada resultado desejado só é alcançado após submissão/execução da referida linha de comando modificada.

4.1 Gráfico de barras simples

Vamos começar com edições em gráficos de barras simples. Para ilustrarmos alterações 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",
  label.bars=TRUE))
```

Exceto pela Subseção 4.1.8, todas as subseções da Subseção 4.1 tomam, 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 na janela *R Script* deve ficar na forma

```
with(Bussab, Barplot(Proced, xlab="Proced", ylab="Frequency",
  label.bars=TRUE,col="red"))
```

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

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`. Por exemplo, submeter a linha de comando abaixo gera um gráfico com barras na cor pêssogo (*peachpuff*) e bordas vermelhas:

```
with(Bussab, Barplot(Proced, xlab="Proced", ylab="Frequency",
  label.bars=TRUE,col="peachpuff",border="red"))
```

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 a variável *Proced* possui três atributos, seu gráfico de barras é composto de três barras. Logo, este vetor de cores é algo do tipo

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

Como ilustração, escreva a expressão `,col=c("blue","pink","green")` no lugar de `,col="red"` na linha de comando explicitada na Subseção 4.1.1 – observe que, com esta modificação, a linha de comando termina com três “fecha parênteses”. Seleccionando toda a linha de comando e clicando em *Submeter*, o gráfico de barras gerado tem primeira barra na cor azul (*blue*), segunda barra na cor rosa (*pink*) e terceira barra na cor verde (*green*).

Observação 6. 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, tal como nas duas linhas de comando sugeridas no final da Seção 3 para o histograma da variável *Renda*.

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 e, conseqüentemente, a largura das mesmas. Quanto maior o `número_de_sua_preferência`, maior o espaçamento entre as barras. O padrão é 0.2, portanto qualquer número maior que 0.2 faz com que as barras fiquem mais espaçadas umas das outras (conseqüentemente, mais finas), e qualquer número positivo menor que 0.2 faz com que as barras fiquem mais próximas umas das outras (conseqüentemente, mais largas). Em particular, `space=0` faz com que as barras fiquem coladas umas às outras, assemelhando-se à 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 *Outra*), 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").
```

Para satisfazer a sugestão 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 7. É possível usar o comando `\n` para quebra de linha em qualquer um dos novos nomes das barras, o que é recomendável para nomes muito compridos.

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 estas 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 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 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 *Outra* ter altura 13 (isto é, 13 indivíduos incluídos). Para refazer o gráfico de barras, mas com o eixo vertical traçado, 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)
```

de modo que a linha de comando a ser submetida fique da forma

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

4.1.7 Gráfico de barras na horizontal

Alternativamente, um gráfico de barras pode ser exibido com barras na horizontal – neste caso, 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, observe 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 contornar este equívoco, 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",  
  label.bars=TRUE,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 de orientação dos nomes no eixo vertical. Para remover a expressão *Proced* do rótulo do eixo-y, substitua `ylab="Proced"` por `ylab=""` e submeta-a novamente.

Observação 8. 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`.

Observação 9. Ao horizontalizarmos um gráfico de barras, as frequências (ou porcentagens) não são mais impressas dentro das barras. Uma alternativa para contornar esta limitação é abordada na Subseção 4.1.8, mas precisamente na Observação 10.

4.1.8 Frequências e porcentagens dos atributos nas barras

Por meio do menu *Gráficos → Gráfico de Barras*, é possível imprimir as frequências ou porcentagens associadas a cada atributo nas barras de um gráfico de barras, marcando a caixinha *Show counts or percentages in bars* na aba *Opções*. Na linha de comando gerada, ter marcado esta caixinha retorna `label.bars=TRUE`, enquanto ter deixado ela desmarcada retorna `label.bars=FALSE`. De todo modo, esta funcionalidade não “mescla” frequências e porcentagens, imprimindo apenas as frequências quando o eixo vertical está na escala de frequências, e imprimindo apenas as porcentagens quando o eixo vertical está na escala de porcentagens.

Mostramos nessa subseção que, por meio da execução de algumas linhas de comando, é possível imprimir frequências e porcentagens juntas em cada barra, o que pode ser bem informativo. Primeiramente, segue um passo a passo de como fazer isto no gráfico de barras da variável *Proced*, para logo após generalizarmos a forma de obtermos gráficos desta forma 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$Proced)
porcentagem <- round(100*contagem/sum(contagem),2)
rotulo <- paste0(contagem," (",porcentagem,"%",")")
k <- 1:length(levels(Bussab$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 edições de interesse, seja no menu e/ou na linha de comando gerada (só não altere o espaçamento entre as barras

e não as coloque na horizontal, por enquanto). É necessário desmarcar a caixinha *Show counts or percentages in bars*, ou seja, é necessário que a linha de comando gerada tenha a expressão `label.bars=FALSE`.

3. Se a **escala do eixo** do gráfico gerado acima está em **contagem de frequências**, copie a seguinte linha de comando e cole-a numa linha em branco da janela *R Script*:

```
text(b, contagem, rotulo, pos=1, col="black")
```

Se a **escala do eixo** do gráfico gerado acima está em **percentagens**, copie a seguinte linha de comando e cole-a numa linha em branco da janela *R Script*:

```
text(b, porcentagem, rotulo, pos=1, col="black")
```

Submeta a linha de comando colada para “carimbar” as frequências e porcentagens nas barras.

Observação 10. Algumas ponderações sobre o passo a passo acima.

- 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.
- 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 Passo 3, a expressão `col="black"` define a cor da fonte do texto “carimbado”, que pode ser alterada para qualquer outra cor de sua preferência.
- É possível também modificar o tamanho da fonte do texto “carimbado” adicionando `,cex=número_de_sua_preferência` antes do “fecha parênteses” das linhas de comando do Passo 3.
- Ainda sobre o Passo 3, a expressão `pos=1` presente nas suas linhas de comando faz com que frequências e porcentagens sejam escritas dentro das barras, no topo. Se esta expressão for alterada para `pos=3`, as frequências e porcentagens passam a se localizar acima das barras, uma vez submetida uma destas linhas de comando.

- No caso de gráficos de barras na horizontal, é necessário trocar **b** e **contagem** (ou **b** e **porcentagem**) de posição nas linhas 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.

Observação 11. Uma alternativa para que seja usada vírgula ao invés de ponto como separador de casa decimal é incluir

```
options(OutDec=",")
```

antes da segunda linha de comando do Passo 1, ou seja, refazer o passo a passo, mas submetendo a sequência de linhas de comando abaixo em vez das indicadas no Passo 1:

```
contagem <- table(Bussab$Proced)
options(OutDec=",")
porcentagem <- round(100*contagem/sum(contagem),2)
rotulo <- paste0(contagem, " (", porcentagem, "%", ")")
k <- 1:length(levels(Bussab$Proced))
b <- k - 0.5 + k*0.2
```

Para que os próximos resultados da atual sessão do R Commander voltem a ter ponto como separador de casa decimal ao invés de vírgula, submeta, na janela *R Script*,

```
options(OutDec=".")
```

4.2 Gráfico de pizza

Nesta subseção, trabalhamos com a variável *Instrução*, em que pese um gráfico de pizza não ser uma representação gráfica recomendada para uma variável qualitativa ordinal. Por comodidade, simplesmente acesse o menu **Gráficos** → **Gráfico de pizza**, selecione a variável *Instrução* e dê OK para que a linha de comando

```
with(Bussab, piechart(Instrução, xlab="", ylab="", main="Instrução",
  col=rainbow_hcl(3), scale="percent"))
```

surja na janela *R Script*. Caso seja o primeiro acesso ao menu de construção de gráficos de pizza na atual sessão do R Commander, também é gerada a linha `library(colorspace, pos=16)`, conforme abordado na Seção 2, mas ela não é útil para nossos propósitos. Por outro lado, a linha de comando em destaque refere-se ao gráfico de pizza gerado, e é nela que são feitas as edições nas Subseções 4.2.1 a 4.2.4.

4.2.1 Mudando as cores das “fatias”

Para modificar as cores no gráfico de pizza da variável *Instrução*, devemos substituir

```
rainbow_hcl(3)
```

por um **vetor de cores** cuja dimensão 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"),
```

cujas ordenação deve seguir a ordem dos atributos da variável em questão. Como visto em (Melo, 2024), os atributos da variável *Instrução* estão ordenados em ordem alfabética: *Fundamental*, *Médio* e *Superior*. Portanto, ao substituírmos a expressão

```
rainbow_hcl(3)
```

na linha de comando associada ao gráfico de pizza da variável *Instrução* por, digamos,

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

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

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. Por exemplo, submeter

```
with(Bussab, piechart(Instrução, xlab="", ylab="", main="Instrução",
  col=c("orange","lightblue","violet"), scale="percent",radius=0.5))
```

na janela *R Script* resulta em um gráfico com as cores propostas na Subseção 4.2.1 e com raio reduzido em relação ao tamanho original.

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

Conforme lembrado na Subseção 4.2.1, os atributos da variável *Instrução* estão ordenados em ordem alfabética: *Fundamental*, *Médio* e *Superior*. Para substituir os nomes *Fundamental*, *Médio* e *Superior* por, digamos, *E.F.*, *E.M.* e *E.S.*, respectivamente, no gráfico de pizza, essa ordenação deve ser obedecida. Todavia, tal modificação necessita, inicialmente, de **uma alteração do código-fonte da função `piechart`**, conforme explicado abaixo.

1. Submeta `piechart <- edit(piechart)` na janela *R Script* para abrir uma nova janela denominada *piechart - Editor R*, com o código-fonte da função `piechart`.

2. No final da primeira linha, digite `labels=levels(x)`, e, na linha composta apenas por `labels <- levels(x)`, digite `#` antes desta expressão, ficando a definição da função `piechart` da seguinte forma:

```
function (x, scale=c("percent", "frequency", "none"), col=rainbow_hcl(nlevels(x)), labels=levels(x),
  ...)
{
  scale <- match.arg(scale)
  if (!is.factor(x))
    x <- as.factor(x)
  #labels <- levels(x)
  tab <- table(x)
  labels <- if (scale == "percent") {
    tab <- 100 * tab/sum(tab)
    paste0(labels, " (", round(tab), "%)")
  }
  else if (scale == "frequency")
    paste0(labels, " (", tab, ")")
  else labels
  pie(tab, labels = labels, col = col, ...)
}
```

3. Feche a janela *piechart - Editor R* e clique em *Sim* para salvar as alterações feitas (ou clique no ícone de disquete no topo da janela *R Gui* antes de fechá-la).

Uma vez realizadas e salvas estas modificações, adicione

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

antes do penúltimo “fecha parênteses” da linha de comando associada ao gráfico de pizza e submeta-a logo após. Cabe ressaltar que este procedimento também funcionaria se tivéssemos escolhido *Contagens de frequências* ou *Neither* no campo *Include in Segment Labels* do menu de construção do gráfico.

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). O menu *Gráficos → Gráfico de pizza* tem, como padrão, a geração de gráficos com esses percentuais impressos, porém aproximados para um percentual inteiro. Nesta subseção, é mostrada uma forma de serem levadas em conta uma ou mais casas decimais de cada percentual no gráfico de pizza. Para tal, é necessário **alterar o**

código-fonte da função `piechart`, assim como foi feito na Subseção 4.2.4. O passo a passo abaixo detalha como proceder para que os percentuais em gráficos de pizza sejam exibidos com precisão de 2 casas decimais.

1. Submeta `piechart <- edit(piechart)` na janela *R Script* para abrir uma nova janela denominada *piechart - Editor R*, com o código-fonte da função `piechart`.
2. Na janela *piechart - Editor R*, procure pela expressão `round(tab)` e substitua ela por `round(tab,2)` para que os percentuais em gráficos de pizza sejam exibidos com precisão de 2 casas decimais.
3. Feche a janela *piechart - Editor R* e clique em *Sim* para salvar as alterações feitas (ou clique no ícone de disquete no topo da janela *R Gui* antes de fechar a janela *piechart - Editor R*).

Uma vez realizadas e salvas estas modificações, qualquer gráfico de pizza na atual sessão do R Commander (gerado pelo menu **Gráficos → Gráfico de pizza** ou por edições em sua linha de comando) explicita os percentuais com duas casas decimais, desde que seja mantida a opção *Percentagens* no campo *Include in Segment Labels* no menu de construção do gráfico (ou, equivalentemente, manter a expressão `scale="percent"` em sua linha de comando).

Exercício 3. Repita o passo a passo acima, porém substituindo

```
paste0(labels, " (", round(tab), "%")
```

ou

```
paste0(labels, " (", round(tab,2), "%")
```

por

```
paste0(labels, " (", round(tab,2), "%","\n",table(x)," indivíduos")
```

no código-fonte da função `piechart`. Em seguida, faça um gráfico de pizza qualquer via menu **Gráficos → Gráfico de pizza** (mantendo o campo *Include in Segment Labels* na opção *Percentagens*) ou submetendo qualquer linha de comando associada a gráfico de pizza (desde que contenha `scale="percent"`).

Observação 12. Para que os percentuais (com uma mais casas decimais) expostos em um gráfico de pizza considerem vírgula ao invés de ponto como separador de casa decimal, submeta `options(OutDec=",")` antes de gerar o gráfico. Para retornar ao padrão de separador de casa decimal do R, submeta `options(OutDec=".")`.

4.3 Histograma

O conjunto de dados *Bussab*, com o qual estamos trabalhando neste material, possui três variáveis quantitativas: *Filhos*, *Idade* e *Renda*.² Trabalhamos nesta subseção com edições em histogramas tomando por base o histograma da variável *Idade*, exceto na Subseção 4.3.5, na qual usamos a variável *Filhos* para geração de histogramas. Portanto, acesse o menu **Gráficos → Histograma** e, por comodidade, simplesmente selecione a variável *Idade* e clique em OK. **A linha de comando gerada, fruto de alterações nas próximas subseções (exceto na Subseção 4.3.5), é a seguinte:**

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

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("skyblue", "firebrick3")` e submeta tal linha de comando modificada para verificar o resultado. Logo após, substitua o vetor de (duas) cores acima por

```
c("skyblue", "firebrick3", "black", "white", "springgreen4", "brown4")
```

²As variáveis *Casado* e *Instr*, apesar de compreendidas pelo R Commander como quantitativas, expressam, na verdade, rótulos numéricos de variáveis qualitativas.

e submeta tal linha de comando modificada.

Observação 13. Assim como visto na Observação 5 para gráficos de barras, é possível modificar a cor padrão (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("green","yellow"), border="blue"))
```

4.3.2 Tamanho das fontes

Em um histograma, é possível modificar o tamanho da fonte dos rótulos dos eixos e dos números em ambos os eixos.

- Adicionar, antes do penúltimo “fecha parênteses” da linha de comando de um histograma, 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 de um histograma, a expressão `,cex.axis=número_de_sua_preferência` faz com que o tamanho da fonte dos **números** nos eixos horizontal e vertical seja modificado (com a submissão da referida linha de comando).

Em qualquer uma das situações acima, maior é o tamanho da fonte quanto maior for o `número_de_sua_preferência`. 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.

4.3.3 Alongando o eixo vertical

No histograma construído para a variável *Idade*, o eixo vertical, representando as frequências, vai de 0 até 10. Para refazer este gráfico com uma folga maior entre o “final” do eixo e o topo da barra mais alta, podemos adicionar, antes do penúltimo “fecha parênteses” da linha de comando do histograma da variável *Idade*, a expressão

```
,ylim=c(0,12)
```

e submeter esta linha de comando.

Observação 14. Para alongar o eixo vertical de um histograma no qual foi escolhido *Percentagens* no campo *Escala do eixo*, devemos continuar nos baseando na escala das frequências. Por exemplo, o histograma da variável *Idade*, na escala percentual, tem a barra mais alta alcançando algo em torno de 28%. Porém, usar `,ylim=c(0,30)` ou `,ylim=c(0,0.30)` para alongar ligeiramente o eixo vertical não surte o efeito desejado. Devemos continuar usando `,ylim=c(0,12)` ou algo próximo.

4.3.4 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: $20 \vdash 25$; $25 \vdash 30$; $30 \vdash 35$; $35 \vdash 40$; $40 \vdash 45$; e $45 \vdash 50$.

Observação 15. 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` ou `$counts`.

Exercício 4. Faça um histograma da variável *Renda* e, logo após, verifique a frequência em cada classe pelo do método descrito na Observação 15.

Observação 16. 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*, marcando a caixinha *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>`.

4.3.5 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 $-0,5 \vdash 0,5$ (contém apenas o valor 0 dentre os possíveis para número de filhos); classe $0,5 \vdash 1,5$ (contém apenas o valor 1 dentre os possíveis para número de filhos); classe $1,5 \vdash 2,5$ (contém apenas o valor 2 dentre os possíveis para número de filhos); e assim sucessivamente até a classe $4,5 \vdash 5,5$, 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 e clique em *Submeter* para termos como saída o histograma da forma desejada. Observe que o único valor possível em cada classe é exatamente o seu respectivo ponto médio.

Exercício 5. 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 17. Caso desejemos classes com amplitude diferente de 1, devemos usar expressões do tipo

```
breaks=seq(início_da_primeira_classe,final_da_última_classe,amplitude)
```

no lugar de expressões como

```
breaks=início_da_primeira_classe:final_da_última_classe.
```

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 delas com amplitude igual a 2 e contendo dois possíveis valores para número de filhos.

Exercício 6. Após construir o histograma sugerido na Observação 17, acesse o menu **Estatísticas** → **Resumos** → **Resumos Numéricos**. Selecione a variável *Filhos* na aba *Dados* e, na aba *Estatísticas*, deixe apenas a caixinha *Frequency Counts* marcada. Ao clicar em OK, é gerada a linha de comando

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

Nesta linha de comando acima, substitua `discreteCounts` por `binnedCounts` e insira a expressão `,breaks=seq(-0.5,5.5,2)` entre o “fecha colchetes” e o “fecha parênteses”. Feito isto e submetendo esta linha de comando, a distribuição de frequências por classes exibida na janela *Output* é condizente com o histograma construído na Observação 17.

4.3.6 Frequências ou porcentagens nas barras de um histograma

Esta subseção apresenta uma forma simples de imprimir frequências ou porcentagens nas barras de um histograma, de forma semelhante ao que a caixinha *Show counts or percentages in bars* do menu **Gráficos** → **Gráfico de Barras** possibilita para gráficos de barras. O passo a passo abaixo retorna ao uso da variável *Idade* para ilustrar esse procedimento.

1. Construa o histograma de interesse (no qual frequências ou porcentagens serão “carimbadas”) no menu **Gráficos** → **Histograma** ou por meio de submissão de linha de comando na janela *R Script*.
2. Na linha de comando do gráfico gerado, escreva `h=` antes do seu início e submeta-a. Por exemplo, a linha de comando base da Subseção 4.3

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

deve ficar na forma

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

3. Submeta a primeira linha de comando abaixo, se o interesse for imprimir **frequências** nas barras; ou submeta a segunda linha de comando abaixo, se o interesse for imprimir **porcentagens** (com precisão de 2 casas decimais) nas barras.

```
text(h$mids, h$counts, h$counts, pos=1, col="black")
```

```
text(h$mids, h$counts, paste0(round(100*prop.table(h$counts),2), "%"),
  pos=1, col="black")
```

Observação 18. Cabe ressaltar que este passo a passo também funcionaria se tivéssemos escolhido *Porcentagens* no campo *Escala do eixo* do menu de construção do gráfico. Porém, se tivéssemos escolhido *Densidades* neste campo, seria necessário substituir `text(h$mids, h$counts,` por `text(h$mids, h$density,` nas linhas de comando do Passo 3.

Observação 19. Nas linhas de comando do Passo 3, é possível configurar outra cor de fonte (diferente de "black") para as frequências ou porcentagens impressas no histograma e/ou trocar `pos=1` por `pos=3`, que faz as frequências ou porcentagens ficarem acima de cada barra em vez de dentro delas, no topo de cada uma. Além disso, é possível usar um vetor de números em vez de 1 ou 3. E na segunda linha de comando do Passo 3, o número 2 se refere à quantidade de casas decimais nas porcentagens – poderia ser utilizado outro número inteiro maior ou igual a zero no lugar de 2, como ilustrado no exercício a seguir.

Exercício 7. Faça um histograma da variável *Filhos* usando `breaks=-0.5:5.5`, conforme detalhado na Subseção 4.3.5, e use essa linha de comando no passo a passo para a impressão de frequências ou porcentagens, porém substituindo `pos=1` no Passo 3 por `pos=c(1,1,1,1,3,1)`.

Observação 20. De maneira análoga ao exposto nas Observações 11 e 12, submeter `options(OutDec=",")` antes do Passo 3 faz com que vírgulas sejam impressas no lugar de pontos como separadoras de casas decimais, e submeter `options(OutDec=".")` reverte a mudança de padrão do R no tocante ao caractere separador de casas decimais.

4.3.7 Polígono de frequências

Esta subseção apresenta uma forma de gerar um polígono de frequências associado a uma distribuição de frequências por classes sobreposto a um histograma também associado a esta distribuição. Histograma e polígono de frequências associado para a variável *Idade* seguem no meio do passo a passo abaixo, 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 modificações eventualmente realizadas em subseções anteriores) é acessar o menu **Gráficos → Histograma**, clicar em *Resetar*, selecionar a variável *Idade* e clicar em OK. Feito isto, é 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`, tal como no passo a passo da Subseção 4.3.6. 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)
```

4. Opcionalmente, a submissão da linha de comando abaixo reforça os pontos médios das classe no topo de cada barra do histograma:

```
points(c(min(h$breaks),h$mids,max(h$breaks)), c(0,h$counts,0),
       col="red", pch=19, cex=1.25)
```

Observação 21. A cor vermelha (*red*) na linhas de comando dos Passos 3 e 4 pode ser substituída pela cor de sua preferência. A expressão `lwd=2` na linha de comando do Passo 3 remete à espessura das linhas do polígono de frequências (quanto maior o número após `lwd=`, mais espessas ficam as linhas que formam o polígono de frequências). Raciocínio análogo se aplica à expressão `cex=1.25` na linha de comando do Passo 4, que regula o tamanho dos pontos no topo de cada barra.

Observação 22. O passo a passo acima também funcionaria caso o histograma gerado no Passo 1 fosse construído com porcentagens em vez de contagens de frequência 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` nas linhas de comando dos Passos 3 e 4.

4.3.8 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, desde que a 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, a linha de comando do histograma da variável *Idade* começa com `with(Bussab, Hist(Idade,`. Para a geração do histograma de interesse, digite a seguinte expressão entre *Idade* e a vírgula que a sucede:

```
[which(Casado_fator=="Sim")]
```

Ao submeter a linha de comando modificada (conforme exibida abaixo), é traçado 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"))
```

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, remetendo às 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.

4.4 Boxplot

Assim como ocorre com histogramas, representações gráficas por meio de boxplots também são úteis para variáveis quantitativas. Nesta subseção, são abordadas edições em boxplots da variável *Renda*, exceto nas Subseções 4.4.7 e 4.4.8, nas quais trabalhamos 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 clique em 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 adicionamos 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 do traçado do boxplot

É possível modificar a cor (originalmente cinza) 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) do boxplot propriamente dito 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="cyan")
```

gera o boxplot da variável *Renda* com caixa preenchida na cor ciano (*cyan*) e o boxplot propriamente dito na cor original preta, ao passo que a linha de comando

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

após submetida, produz o boxplot da variável *Renda* na cor coral (e interior da caixa na cor original cinza). Para editar, em um mesmo boxplot, cor da caixa e do traçado do boxplot, 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="cyan",border="coral")
```

Submetendo a linha de comando acima na janela *R Script*, é gerado o boxplot da variável *Renda* na cor coral, com a cor ciano no interior da caixa.

4.4.2 Mudando as cores da caixas e dos traçados 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 os traçados da mesma cor. Para cores diferentes em cada boxplot (sejam nas caixas ou nos gráficos propriamente ditos), o procedimento é análogo ao abordado 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("yellow","royalblue"),border=c("purple","red")
```

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 na cor roxa com caixa preenchida de amarelo, e o boxplot seguinte na cor vermelha com caixa preenchida de azul royal.

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 4.4.2), porém com *Casados* no lugar de *Sim* e *Solteiros* no lugar de *Não*. 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.

Observação 23. 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 por meio de `\n`.

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 vertical (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 vertical 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, maior (menor) é o tamanho da fonte quanto maior (menor) for o `número_de_sua_preferência`. 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 Alongando (ou encurtando) o eixo vertical

O eixo vertical na janela que comporta um ou mais boxplots 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, submeter a linha de comando

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

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

Observação 24. Em boxplots com valores discrepantes exageradamente distantes do limite inferior e/ou do limite superior, encurtar o eixo horizontal pode ser interessante no sentido de visualizar mais precisamente o comportamento da variável a menos dos valores discrepantes, agindo como uma “lupa”.

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 visualmente mais interessantes para destacar os 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"
```

em seu lugar para obter o mesmo resultado. Contudo, este método alternativo permite escolher qualquer cor disponível no R para o preenchimento dos círculos, com a cor do contorno do círculo definida pela cor da borda do traçado do(s) boxplot(s), cuja forma de edição foi descrita nas Subseções 4.4.1 e 4.4.2. As quatro linhas de comando abaixo são edições de linhas de comando geradas por boxplots por grupos da variável *Idade*, com *Instrução* como variável de grupo. Submeta-as uma de cada vez para verificar o resultado de cada uma delas.

```
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", cex=1.75)
```

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 se resume 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 um valor

discrepante, identificado por 35. Como, no conjunto de dados *Bussab*, a identificação de cada observação resume-se no número da linha, tal valor representa simplesmente que este indivíduo com idade discrepante (dentre aqueles que tem ensino médio com nível de instrução) está na 35ª linha do conjunto de dados – a propósito, ele não tem 35 anos! Tais identificações podem ser omitidas quando é selecionada a opção *Não* no campo *Identificar outliers* (aba *Opções*) no menu de construção de boxplots. Entretanto, abordamos nesta subseção uma alternativa interessante, que consiste em identificar valor(es) discrepante(s) pelo(s) respectivo(s) valor(es) que a variável em questão assume.

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, tomamos 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. A marcação 35 ao lado do único valor discrepante é substituída por 48, que é a idade (em anos completos) do indivíduo que ocupa a 35ª linha do conjunto de dados *Bussab*.

Exercício 8. Na linha de comando dos boxplots das idades pelo nível de instrução, substitua `method="y"` (ou `labels=Bussab$Idade`) por

```
labels=paste(Bussab$Idade, "anos")
```

e submeta-a para que a marcação 48 ao lado do valor discrepante seja substituída por 48 anos.

Observação 25. 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")
```

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 abordadas na Subseção 4.1 para gráficos de barras simples são aplicáveis (e razoáveis) em gráficos de barras múltiplas, seguindo os mesmos procedimentos apresentados 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 agrupamento de barras (conforme Subseção 4.1.4);
- tamanho das fontes (conforme Subseção 4.1.5);
- alongamento 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*, esta última escolhida como variável de grupo, 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", label.bars=TRUE))
```

Em toda a Subseção 4.5, a linha de comando acima é usada 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. Observe 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 `"above"`, `"topright"` ou `"top"` em vez de `"topleft"`, respectivamente. 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 maioria destas seis posições listadas não é aconselhável, sobretudo a posição *bottomleft*, pois ela faz 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", label.bars=TRUE))
```

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

1. Rótulo do eixo-x (*Proced*) e rótulo do eixo-y (*Frequency*) mantiveram-se no local original, mas precisam trocar de lugar;
2. Frequências outrora impressas nas barras estão soltas e desconexas no novo gráfico.
3. Legenda está se sobrepondo às barras associadas à região de procedência *Outra*.

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. O segundo problema é resolvido removendo as frequências ao substituir `label.bars=TRUE` por `label.bars=FALSE`. Em relação à posição da legenda, note que o canto inferior direito parece propício para posicioná-la. Portanto, substitua `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",
  label.bars=FALSE, 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 Observação 8, usamos `xlim` ao invés de `ylim` para alongamento do eixo das frequências. Para esticar o eixo-x (originalmente exibido de 0 até 7) para, digamos, de 0 até 8, adicione a expressão `,xlim=c(0,8)` 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. No gráfico base da Subseção 4.5, 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", label.bars=TRUE,
  col=c("olivedrab", "salmon", "turquoise")))
```

que é a linha de comando tomada como referência nesta Subseção 4.5 com a expressão `,col=c("olivedrab", "salmon", "turquoise")` adicionada antes do penúltimo “fecha parênteses”. Note que as barras associadas aos indivíduos com ensino fundamental como nível de instrução foram pintadas de verde oliva (*olivedrab*), e a referência de cor na legenda também é modificada para verde oliva de forma automática. O mesmo ocorre para o nível de instrução *Médio* (cor salmão) e para o nível de instrução *Superior* (cor turquesa).

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", label.bars=TRUE,
  col=c("olivedrab", "salmon", "turquoise")))
```

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", inset=0.05, title="Instrução", c("E.F", "E.M", "E.S"),
  fill=c("olivedrab", "salmon", "turquoise"))
```

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

- `"topleft"` : posição da legenda, que pode ser qualquer posição dentre as apresentadas na Subseção 4.5.1);
- `inset=0.05` : para que a legenda não fique “colada” no eixo vertical (outro número positivo pode ser usado no lugar de 0.05);
- `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("olivedrab", "salmon", "turquoise")` : cores na legenda, que devem ser as mesmas e seguir a mesma ordem do vetor de cores adicionado na linha de comando associada ao gráfico, à direita de `col=`).

Observação 26. Para executar edições conforme os passos acima em gráficos de barras múltiplas, porém 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("olivedrab", "salmon", "turquoise")`, uma expressão com a seguinte estrutura:

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

No gráfico que estamos explorando, *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 nesta subseção, tomemos um diagrama de dispersão entre as variáveis *Idade* e *Renda*, 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 seguinte linha de comando na janela *R Script*:

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

Tal como nas edições em boxplots abordadas na Subseção 4.4, as expressões que adicionamos em linhas de comando de diagramas de dispersão devem ser inseridas antes do último “fecha parênteses” da linha de comando. De fato, a linha de comando de referência acima tem apenas um “abre parênteses” e, conseqüentemente, apenas um “fecha parênteses”.

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 sejam coloridos 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`). Por exemplo, para recuperar a cor original da reta de regressão, mas mantendo os pontos na cor preta, uma linha de comando conforme abaixo deve ser submetida na janela *R Script*:

```
scatterplot(Renda~Idade, regLine=list(col="blue"), smooth=FALSE,
            boxplots=FALSE, data=Bussab,col="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 faz com que a reta de regressão fique mais espessa, e qualquer número (positivo) menor que 2 faz 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=4), smooth=FALSE, boxplots=FALSE,
            data=Bussab)
```

Observação 27. 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` 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, continuamos 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 do diagrama de dispersão por grupos 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 em um tom de vermelho (*red2*) e os pontos e reta de regressão dos indivíduos não casados fiquem na cor azul piscina (*dodgerblue*):

```
scatterplot(Renda~Idade | Casado_fator, regLine=TRUE, smooth=FALSE,
  boxplots=FALSE, by.groups=TRUE, data=Bussab,col=c("red2","dodgerblue"))
```

Observação 28. 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 do referido gráfico.

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, pois só é possível definir uma espessura, sendo esta 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, do seguinte modo: 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 ) 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 (de Pearson) entre *Idade* e *Renda* de todos os indivíduos do conjunto de dados *Bussab* é exibido na janela *Output*: 0,3633622.

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

- Bussab, W.; Morettin, P. (2017). *Estatística Básica*. São Paulo: Saraiva, 9ª edição.
- DataCamp (n.d.). Pie charts in R - Pie Chart with Annotated Percentages. Em DataCamp. Disponível em: <https://www.datacamp.com/doc/r/pie>.
- Fox, J. (2005). *The R Commander: A Basic-Statistics Graphical User Interface to R*. Journal of Statistical Software, 14(9), 1–42. <https://doi.org/10.18637/jss.v014.i09>.
- Fox, J.; Weisberg, S. (2019). *An R Companion to Applied Regression*, Third edition. Sage, Thousand Oaks CA. Disponível em: <https://www.john-fox.ca/Companion/index.html>.
- Fox, J. (2022) *Getting started with the R Commander - Version 2.8-0*. Disponível em: <https://cran.r-project.org/web/packages/Rcmdr/vignettes/Getting-Started-with-the-Rcmdr.pdf>.
- Fox, J.; Marquez, M. (2023). *RcmdrMisc: R Commander Miscellaneous Functions*. R package version 2.9-1. Disponível em: <https://CRAN.R-project.org/package=RcmdrMisc>.
- Fox, J.; Bouchet-Valat, M. (2024). *Rcmdr: R Commander*. R package version 2.9-2. Disponível em: <https://cran.r-project.org/web/packages/Rcmdr/index.html>.
- Melo, F.R.R. (2024). *Introdução ao R Commander - Notas de Aula*. Disponível em: <http://gae.uniriotec.br/7/material.html>.
- Mignozzetti, U.G. (2009). *Introdução ao R Commander*. Disponível em: https://www.bessegato.com.br/UFJF/est000_edu/aprcmdr.pdf.
- Priberam (n.d.). Significado de *Editar*. Dicionário Priberam da Língua Portuguesa. Disponível em: <https://dicionario.priberam.org/editar>.
- R Core Team (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Viena, Áustria. Disponível em: <https://www.R-project.org/>.

- R Documentation (n.d.). `barplot`: Bar Plots. Documentation for the `graphics` package, Version 3.6.2. Disponível em: <https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/barplot>.
- R Documentation (n.d.). `boxplot`: Box Plots. Documentation for the `graphics` package, Version 3.6.2. Disponível em: <https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/boxplot>.
- R Documentation (n.d.). `hist`: Histograms. Documentation for the `graphics` package, Version 3.6.2. Disponível em: <https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/hist>.
- R Documentation (n.d.). `pie`: Pie Charts. Documentation for the `graphics` package, Version 3.6.2. Disponível em: <https://www.rdocumentation.org/packages/graphics/versions/3.6.2/topics/pie>.
- R Documentation (2023). `Barplot`: Bar Plots. Documentation for the `RcmdrMisc` package, Version 2.9-1. Disponível em: <https://www.rdocumentation.org/packages/RcmdrMisc/versions/2.9-1/topics/Barplot>.
- R Documentation (2023). `Boxplot`: Boxplots With Point Identification. Documentation for the `car` package, Version 3.1-2. Disponível em: <https://www.rdocumentation.org/packages/car/versions/3.1-2/topics/Boxplot>.
- R Documentation (2023). `Hist`: Plot a Histogram. Documentation for the `RcmdrMisc` package, Version 2.9-1. Disponível em: <https://www.rdocumentation.org/packages/RcmdrMisc/versions/2.9-1/topics/Hist>.
- R Documentation (2023). `piechart`: Draw a Piechart With Percents or Counts in the Labels. Documentation for the `RcmdrMisc` package, Version 2.9-1. Disponível em: <https://www.rdocumentation.org/packages/RcmdrMisc/versions/2.9-1/topics/piechart>.
- R Documentation (2023). `scatterplot`: Enhanced Scatterplots with Marginal Boxplots, Point Marking, Smoothers, and More. Documentation for the `car` package, Version 3.1-2. Disponível em: <https://www.rdocumentation.org/packages/car/versions/3.1-2/topics/scatterplot>.

- Zeileis, A.; Hornik, K.; Murrell, P. (2009). Escaping RGBland: Selecting Colors for Statistical Graphics. *Computational Statistics & Data Analysis*, **53**(9), 3259-3270. <https://doi.org/10.1016/j.csda.2008.11.033>.
- Zeileis, A.; Fisher, J.C.; Hornik, K.; Ihaka, R.; McWhite, C.D.; Murrell, P.; Stauffer, R.; Wilke, C.O. (2020). colorspace: A Toolbox for Manipulating and Assessing Colors and Palettes. *Journal of Statistical Software*, **96**(1), 1-49. <https://doi.org/10.18637/jss.v096.i01>.

Anexo I - Lista de cores disponíveis no R por nome

	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

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

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

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

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

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

	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

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	mincream	mistyrose	mistyrose1

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

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

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			