Aplicações Computacionais em Exploração e Análise de Dados: Visualização Descritiva

Leandro Vitral Andraos (Bolsista IC, Departamento de Estatística, UFJF)
Marcel de Toledo Vieira (Professor, Departamento de Estatística, UFJF)

1. Introdução

O *R* é um programa estatístico computacional disponível **gratuitamente** através da *internet* sob a *General Public License*. Na verdade, o *R* é mais que um pacote estatístico, é um ambiente de desenvolvimento estatístico bastante flexível, e é também uma linguagem de programação completa. Nesta aula estaremos tratando de conceitos elementares e fazendo uso de alguns exemplos.

Este *software* é **compatível** com as três principais plataformas: *Unix/GNU/Linux/FreeBSD*, *Machintosh* e *Windows*. Além disso, possui diversas funcionalidades, e permite a incorporação de novos comandos e extensões, programadas na linguagem *R* ou em outras linguagens como *C*, *Perl1*, *Python2*, *Lisp3* ou *Tcl/Tk4*.

2. Onde Obter o *R*?

Informações detalhadas sobre o *R* estão **disponíveis** no sítio do *Comprehensive R Archive Network* (CRAN):

http://www.r-project.org/

Um *link* **direto** para o *download* da versão 3.0.1 é:

http://cran.fiocruz.br/bin/windows/base/

Clique em Download R 3.0.1 for Windows. Após o *download* dê um duplo clique no arquivo e siga os passos de **instalação**. Desta maneira, o pacote será instalado. Para executá-lo deve-se acessar o aplicativo *R* a partir do *menu* de Programas.

3. Estatística Descritiva

Gráficos e tabelas são recursos amplamente utilizados para representar resultados de estudos e informações de uma forma organizada e clara. Com estas ferramentas, podemos visualizar informações quantitativas de forma resumida, o que facilita a utilização desses resultados para a tomada de decisões.

A construção de gráficos é, certamente, um dos mais importantes aspectos da **análise exploratória de dados**.

Iremos começar com o exemplo fictício da tabela abaixo:

	Número de Professores	Número de Alunos
Privada	1751	25280
Estadual	1186	21328
Municipal	947	18432
Federal	29	280

Para digitar no R nossa tabela podemos utilizar o seguinte comando:

```
dados<-matrix(c(1751,1186,947,29,25280,21328,18432,280),nrow=4,ncol=2)
rownames(dados)= c("Privada","Estadual","Municipal","Federal")
colnames(dados)=c("Número de Professores", "Número de Alunos")
dados
```

A seguir, faremos gráficos de barras para a tabela acima, utilizando inicialmente somente a primeira coluna através do seguinte comando:

```
barplot(dados[,1])
```

Que tal alterarmos as cores das barras dos gráficos?

```
barplot(dados [,1], col=c("blue"))
barplot(dados [,1], col=c("red","green","blue","hotpink"))
```

Ficou bem melhor, não ficou? Sinta-se a vontade para escolher suas próprias cores!

```
colors()
```

Vamos a seguir adicionar um título ao nosso gráfico? (*digitar em uma mesma linha*)

```
barplot(dados [,1], col=c("red","green","blue","hotpink"), main="Distribuição de
Professores na Rede de Ensino")
```

E quanto à escala? Podemos alterar a do eixo y:

```
barplot(dados [,1], col=c("red","green","blue","hotpink"),main="Distribuição de
Professores na Rede de Ensino",ylim=c(0,3000))
```

Podemos também adicionar *labels* aos eixos x e y da seguinte forma:

```
barplot(dados[,1],col=c("red","green","blue","hotpink"),main="Distribuição de
Professores na Rede de Ensino",ylim=c(0,3000),xlab="Escolas",ylab="Frequencia")
```

E agora adicionar, por exemplo, uma referencia sobre a fonte dos dados:

```
barplot(dados[,1], col=c("red","green","blue","hotpink"),main="Distribuição de
Professores na Rede de Ensino",ylim=c(0,3000),xlab="Escolas",ylab="Frequencia",
sub="Fonte:www.ibge.com.br")
```

Podemos agora hachurar as barras do nosso gráfico:

```
barplot(dados[,1], col=c("red","green","blue","hotpink"),main="Distribuição de
Professores na Rede de Ensino",ylim=c(0,3000),xlab="Escolas",ylab="Frequencia",
sub="Fonte:www.ibge.com.br", density=30)
```

Se quisermos adicionar uma borda laranja em cada barra utilizamos o comando abaixo:

```
barplot(dados[,1], col=c("red","green","blue","hotpink"),main="Distribuição de
Professores na Rede de Ensino",ylim=c(0,3000),xlab="Escolas",ylab="Frequencia",
sub="Fonte:www.ibge.com.br", border="orange")
```

É possível invertermos o gráfico e visualizá-lo na forma horizontal da seguinte forma:

```
barplot(dados[,1], col=c("red","green","blue","hotpink"),main="Distribuição de
Professores na Rede de Ensino",xlim=c(0,3000),ylab="Escolas",xlab="Frequencia",
sub="Fonte:www.ibge.com.br",horiz=T)
```

Perceba que agora invertemos x com y!

3.1 Gráficos com as duas variáveis

A partir do momento em que entendemos como fazer o gráfico para uma variável, adicionaremos uma segunda e refaremos os gráficos:

```
barplot(dados)
```

Na verdade, esse gráfico terá um melhor aproveitamento se for feito da seguinte forma:

```
barplot(dados, beside=TRUE)
```

Alterando as cores, título e eixos, assim como feito anteriormente teremos:

```
barplot(dados[,2:1],beside=TRUE, main="Distribuição
do número de Alunos e Professores",ylab="Frequencia",
col=c("red","green","blue","hotpink"))
```

Perceba que ao colocarmos `dados[,2:1]`, invertemos a ordem do gráfico. Compare com o gráfico anterior e perceba que as barras dos professores e alunos trocaram de lugar.

Que tal se agora adicionarmos uma legenda à figura?

```
barplot(dados[,2:1],beside=TRUE,legend.text=rownames(dados), main="Distribuição
do número de Alunos e Professores",ylab="Frequencia",
col=c("red","green","blue","hotpink"))
```

3.2 Gráfico de Setores

Podemos visualizar nossa tabela anterior como um gráfico como setores, também conhecido como gráfico de pizza ou de torta ☺.

```
pie(dados[,1])
```

Podemos melhorar o gráfico da seguinte forma:

- adicionado um título:

```
title("Professores na Rede de Ensino")
```

- calculando a porcentagem referente a cada categoria:

```
porcentagem<-(dados[,1]*100/sum(dados[,1]))
porcentagem
```

Muitas casas decimais não acham? Vamos arredondar somente para duas.

```
porcentagem<-round(dados[,1]*100/sum(dados[,1]),2) #
porcentagem
```

Podemos agora adicionar as porcentagens ao gráfico, mas primeiro precisamos dizer ao R que esses serão os rótulos do gráfico (sendo separados por aspas no comando abaixo).

```
rotulos<-paste("(",porcentagem,"%)",sep="")

rotulos

pie(dados[,1], main=" Professores na Rede de Ensino ",labels=rotulos,
col=rainbow(7))

legend(1,1,names(dados[,1]),col = rainbow(7),pch=rep(20,6))
```

Se quisermos mudar a angulação do gráfico de pizza podemos girá-lo 180° da seguinte forma:

```
pie(dados[,1], main=" Professores na Rede de Ensino ",labels=rotulos,
col=rainbow(7),init.angle=180)

legend(1.2,1,names(dados[,1]),col = rainbow(7),pch=rep(20,6))
```

Podemos então fazer o gráfico de setores para o número de alunos por Escola e o de número de professores por Escola. Podemos fazer os dois gráficos juntos a partir do comando `mfrow()`.

Mas primeiro vamos calcular novamente as porcentagens, só que dessa vez para os alunos:

```
porcentagem2<-round(dados[,2]*100/sum(dados[,2]),2) #

rotulos2<-paste("(",porcentagem2,"%)",sep="")

par(mfrow=c(1,2))

pie(dados[,1], main=" Professores na Rede de Ensino ",labels=rotulos,
col=rainbow(7))

legend(-1.0,2.1,names(dados[,1]),col = rainbow(7),pch=rep(20,6))

pie(dados[,2], main=" Alunos na Rede de Ensino ",labels=rotulos2, col=rainbow(7))

legend(-1.0,2.1,names(dados[,2]),col = rainbow(7),pch=rep(20,6))
```

3.3 Histograma

O *R* traz em seus arquivos de instalação diversas **bases de dados** interessantes. Verifiquem através do seguinte comando:

```
data()
```

Vamos construir um **histograma** para a variável "volume de vazão do Rio Nilo".

```
data(Nile)

hist(Nile)
```

Podemos alterar o histograma com os parâmetros utilizados acima:

```
hist(Nile, ,density=30,col="blue",border=TRUE,main=" Histograma da variável
Vazão",xlab="Vazão",ylab="Frequencia")
```

É possível ajustarmos uma curva ao nossos dados, mas primeiro instalaremos e carregaremos o pacote **basicStats**:

```
install.packages("fBasics")

library("fBasics")

histPlot(as.timeSeries(Nile))
```

3.4 Ramo e folhas

Podemos fazer um diagrama de ramo e folhas através do seguinte comando:

```
stem(Nile)
```

Ou ainda:

```
stem(Nile,scale=2)
```

3.5 Box Plot

Um *boxplot* pode ser construído da seguinte maneira:

```
boxplot(Nile)
```

Ou ainda pelo pacote fBasics carregado acima:

```
boxPlot(Nile)
```

Para apresentar o gráfico na horizontal fazemos:

```
boxPlot(Nile,horizontal=T)
```

Como sabemos, os *boxplots* podem ser utilizados para a comparação de diferentes grupos (vamos utilizar nosso exemplo anterior):

```
boxPlot(dados,main="Box Plot",title=FALSE)
```

4. Gráfico de dispersão e regressão linear simples

Vamos agora trabalhar com o banco de dados *Orange* que está disponível no R. Com esses dados, criaremos o gráfico de dispersão para observar se há relação entre o tamanho da circunferência da árvore (variável dependente) com sua idade (variável explicativa).

```
plot(Orange[,c(2,3)],col="Red",main="Gráfico de idade x
Circunferencia",xlab="Idade",ylab="Circunferencia")
```

Podemos localizar um ponto qualquer no gráfico da seguinte forma:

```
locator(1)

reg <- lm(circumference ~age,data=Orange)

reg

abline(reg,col="blue")
```

5. Análise de Correspondência

Vamos utilizar uma outra ferramenta estatística que nos fornece um gráfico extremamente interessante e de fácil entendimento. Consideremos a variável *crimes* que inclui informações sobre o número de crimes registrados em diferentes regiões da Noruega.

```
crimes<- matrix(c(395,147,694,2456,152,327,1758,916,1347),ncol=3)
```

Podemos tratar esta matriz como uma tabela. Para isso, daremos nomes as colunas e as linhas.

```
colnames(crimes)<- c("Assalto/roubo", "Fraude", "Vandalismo")
rownames(crimes)<- c("Grande Oslo","Centro","Norte")
crimes
install.packages("ca")
library(ca)
a=ca(crimes)
plot(a, main="Crimes na Noruega por Região")
```

6. Outras opções de gráficos que o R oferece

Podemos obter demonstrações sobre as potencialidades gráficas do R através dos seguintes comandos:

```
install.packages(rgl)
demo(graphics)
demo(rgl)
```

7. Copiando os resultados para o *Microsoft Word*

É possível copiar e **colar** as saídas de funções do *R* caso seja de interesse. Para isso apertamos o botão esquerdo do *mouse* e o arrastamos sobre o texto. O tom azul sobre a tela indica a parte que está sendo selecionada. Em seguida copiamos selecionando no menu “Edit” a opção “Copy”, ou simplesmente apertando sucessivamente a tecla “Ctrl” e a tecla “C”. Outra possibilidade é clicar com o botão direito do *mouse* e escolher “Copy”. Depois disso, pode-se colar o conteúdo no *Word*. Para copiar gráficos, o procedimento é semelhante.

Onde encontrar material sobre o *R* na *internet*? (apenas algumas sugestões...)

Em português:

http://leg.ufpr.br/Rtutorial/
http://leg.ufpr.br/Rpira/Rpira/
http://www.void.cc/r/
http://www.feferraz.net/br/R-mae5704.html

Em inglês:

http://faculty.washington.edu/tlumley/Rcourse/
http://www.cas.lancs.ac.uk/short_courses/intro_r.html

Livros e apostilas sobre o *R*:

Beasley, C. R. (2004) *Bioestatística Usando R*. Universidade Federal do Pará, Bragança. (1)

Dalgaard, P. (2002) *Introductory Statistics with R*. New York, Springer. (2)

Lumley, T. R (2006) *Fundamentals and Programming Techniques*. R Core Development Team, Birmingham. (3)

Maindonald, J. H. (2004) *Using R for Data Analysis and Graphics – Introduction, Code and Commentary*. Centre for Bioinformation Science, Australian National University. (4)

Pacheco, A. G. F., Cunha, G. M. e Andreozzi, V. L. *Aprendendo R*. Escola Nacional de Saúde Pública, FioCruz, Rio de Janeiro. (5)

Paradis, E. *R for Beginners*. Institut des Sciences de l' Evolution. Universite Montpellier II, Montpellier. (6)

Torgo, L. (2006) *Introdução à Programação em R*. Universidade do Porto, Porto. (7)

Verzani, J. *Using R for Introductory Statistics*. (8)

As referências (1), (3) a (8) podem ser disponibilizadas por e-mail ou baixadas pela *internet*. Além disso, o *R* oferece no *menu* "Help → Manuals (in PDF)", seis outros documentos bastante explicativos.