

Introdução ao Software R

Marcos Santos de Oliveira

Departamento de Matemática e Estatística - DEMAT, UFSJ
36307 – 904, São João Del - Rei, MG

mso@ufsj.edu.br

RESUMO

O objetivo desse mini-curso é apresentar um tutorial sobre o software R. O sistema R é um programa computacional estatístico distribuído gratuitamente e de código aberto. Ilustraremos aspectos básicos do sistema com ênfase na compreensão da linguagem, a estrutura e a forma de operar o programa. Tópicos básicos de matemática e estatística serão utilizados para ilustrar o uso da linguagem.

Palavras-chave: Software livre. Computação científica. Estatística.

Cronograma do minicurso

- Introdução
- Uso da Ajuda do R
- Operações Aritméticas
- Vetores
- Sequências
- Valores Lógicos e Índices
- Estatísticas elementares
- Matrizes
- Gráficos
- Integração Numérica

O que é R?

- R é uma linguagem e ambiente para computação estatística e gráficos.
- Roda em \neq plataformas (Unix, Linux, Windows, Macintosh, etc.)

Origem do R

- Iniciou-se com Robert Gentleman e Ross Ihaka (Statistics Department of the University of Auckland - New Zealand) em 1991.
- Equipe principal do R foi formada em 1997.
- Fundação R em 2003.
- A última versão do R (2.12.0) foi disponibilizada em Outubro de 2010.

www.r-projet.org

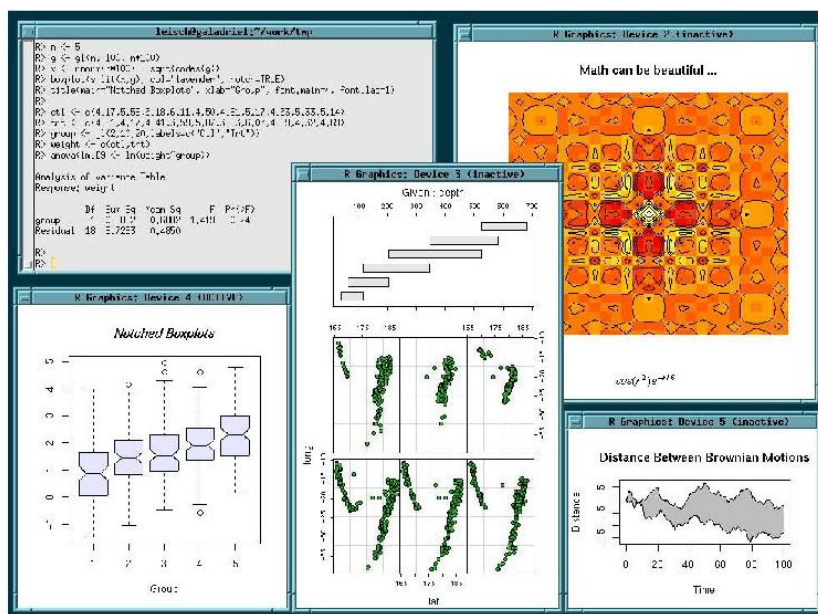
O que é R?

- O R disponibiliza uma grande variedade de métodos estatísticos (modelagem linear e não linear, testes estatísticos clássicos, séries temporais, métodos multivariados, ...) e técnicas gráficas.
- Diferentes autores constroem pacotes (packages) de suas pesquisas e disponibilizam as rotinas em R. Há atualmente **2574** pacotes disponíveis.
- Um dos pontos fortes do R é a facilidade com que gráficos bem delineados e de alta qualidade para impressão podem ser produzidos com possibilidade de inclusão de **fórmulas** e **símbolos matemáticos** quando necessário.

Por que R?

- Software livre e de código aberto.
- Novas funcionalidades são facilmente implementadas e distribuídas.
- Possui interface com outros softwares.
- Instalação no Windows é ~ 32 MB.
- Gráficos de alta definição.

Ilustração de gráficos em R



Rodando o R

R trabalha interativamente usando o modelo de perguntas e respostas:

- Inicie o R
- R providenciará uma linha de comando representado pelo símbolo `>` e aguardará por entradas
- Entre com um comando e pressione enter
- R executará esse comando e imprimirá os resultados
- R aguardará por mais entradas
- Para sair do R usamos a função `q()`

Usando a ajuda (Help) do R

O R tem um sistema de ajuda on-line que permite que a documentação seja exibida em um browser (explorer, netscape, mozilla ou similar). Para iniciar este sistema on-line digite:

```
> options(helphtml=T)
> help.start()
```

Este comando irá inicializar o seu browser e exibir a documentação on-line no browser.

- Clicando em **packages** uma página se abrirá com uma listagem dos pacotes disponíveis.
- Clicando em um dos **pacotes** será aberta uma página com uma lista das funções incluídas no pacotes e uma rápida **descrição** de cada uma delas.
- Clicando na função se abrirá uma janela com a **documentação** específica daquela função.

Usando a ajuda (Help) do R

Dentro do R você pode usar:

```
> help(mean)
```

ou simplement,

 $\bar{y} > ? \text{mean}$

É possível fazer procura por tópicos usando **Search Engine & Keywords** na ajuda on-line.

- Esse mecanismo procura nas descrições de todas as funções do R a palavra chave digitada.
- Por exemplo, localize informações de palavra **mean**.

R como calculadora

Voce pode usar o R para avaliar algumas expressões aritméticas simples.
Por exemplo:

```
> 2+3          somando números
[1] 6
```

```
> 2+3*4        prioridade de operações (multiplicação primeiro)
[1] 14
```

```
> 3/2+1        assim como na divisão
[1] 2.5
```

```
> log(8)        logaritmo natural
[1] 2.079442
```

```
> log(8, base=2) logaritmo na base 2
[1] 3
```

R como calculadora

```
> exp(1)        exponencial
[1] 2.718282
```

```
> 2^4           potências são indicadas por ^ ou **
[1] 16
```

```
> sqrt(2)       raiz quadrada
[1] 1.414214
```

```
> sin(3.14159)  seno(Pi radianos) é zero
[1] 2.65359e-06  a resposta é bem próxima
```

```
> sin(pi)
[1] 1.224606e-16 bem mais próximo de zero
```

Algumas funções aritméticas no R

Nome	Operação
sqrt	raiz quadrada
abs	valor absoluto (positivo)
sin cos tan	funções trigonométricas
asin acos atan	funções trigonométricas inversas
sinh cosh tanh	funções hiperbólicas
asinh acosh atanh	funções hiperbólicas inversas
exp log	exponencial e logaritmo natural
log10	logaritmo base-10
pi	número pi (3,141593)

Armazenando valores em objetos do R

R é uma **linguagem orientada o objeto**: variáveis, dados, matrizes, funções, etc, são armazenados na memória ativa do computador na forma de objetos.

Por exemplo:

```
> x <- 10    # indica que o valor 10 foi armazenado na variável x.
```

Alternativamente pode-se usar o símbolos = ou ->.

As linhas a seguir produzem o mesmo resultado:

```
> x <- sin(pi)
> x = sin(pi)
> sin(pi) -> x
```


Comando repetição rep()

- A função rep() retorna o primeiro argumento repetido o número de vezes indicado pelo segundo argumento.

Exemplos:

```
> rep(1,10)           > rep(c(1,2),5)
[1] 1 1 1 1 1 1 1 1 1 1 [1] 1 2 1 2 1 2 1 2 1 2
```

- Se ambos argumentos tem mais de um elemento então cada elemento do primeiro argumento será associado ao elemento correspondente do segundo argumento.

Veja esses exemplos:

```
> rep(4:1,1:4)         > rep(c(23, 32, 42), c(3, 1, 2))
[1] 4 3 3 2 2 2 1 1 1 1 [1] 23 23 23 32 42 42
```

◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶

Marcos S. Oliveira (UFSJ)

I ERMAC

Nov, 2010

19 / 40

Operações com vetores

- Operações aritméticas em vetores são efetuadas para cada um de seus elementos.

```
> x = 1:10             > x * 2
> x + 2                [1] 2 4 6 8 10 12 14 16 18 20
[1] 3 4 5 6 7 8 9 10 11 12
```

- Operações em dois vetores são também feitos em um elemento de cada vez.

Exemplo:

```
> x = 1:10
> y = 21:30
> x+y
[1] 22 24 26 28 30 32 34 36 38 40
```

◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶ ◀ ▶

Marcos S. Oliveira (UFSJ)

I ERMAC

Nov, 2010

20 / 40

Operações com vetores

Importante característica de vetor

- Vetores com tamanhos diferentes: os elementos do **menor** vetor serão repetidos até atingir o tamanho do **maior** vetor.

Exemplos:

```
> x = 1:10
> y = c(1,2)
> x+y
[1] 2 4 4 6 6 8 8 10 10 12
```

```
> x = 1:10
> y = c(1,2,1)
> x + y
[1] 2 4 4 5 7 7 8 10 10 11
```

Caracteres em R

- O R pode armazenar dados alfanuméricos da mesma forma que armazena dados numéricos.

Exemplos:

```
> c1 = "Marcos"           um caracter como escalar
> c1
[1] "Marcos"
```

```
> c2 = c("Sim", "Talvez", "Não")  um vetor de caracteres
> c2
[1] "Sim" "Talvez" "Não"
```

Fatores em R

- Fatores são usados para armazenar dados categóricos.

Exemplo: Suponha que você tem dados de altura de um grupo de pessoas e quer armazenar a informação sobre o sexo dos indivíduos.

```
> altura =  
c(1.84,1.78,1.69,1.72,1.75,1.81)  
> altura  
[1] 1.84 1.78 1.69 1.72 1.75 1.81  
  
> sexo = c('M','F','F','F','M','M')  
> sexo  
[1] "M" "F" "F" "F" "M" "M"  
  
> fator.sexo = as.factor(sexo)  
> fator.sexo  
[1] M F F F M M  
Levels: F M  
  
> mean(altura)  
[1] 1.765  
  
> by(altura,fator.sexo,mean)  
fator.sexo: F [1] 1.73  
  
fator.sexo: M [1] 1.8
```

Valores lógicos

- O R possibilita computação com valores Booleanos. Estas variáveis podem ter valores TRUE ou FALSE (verdadeiro ou falso).

```
> x = 1:5
> x < 3
[1] TRUE TRUE FALSE FALSE FALSE
```

- É possível realizar operações numéricas com valores lógicos. O R interpreta TRUE como o valor 1 e FALSE como o valor 0. Isto pode ser utilizado, por exemplo, para contar o número de valores em um vetor que obedece a uma determinada condição.

```
> x = 1:6
> y = x > 3
> y
[1] FALSE FALSE FALSE TRUE TRUE TRUE
> sum(y)
[1] 3
```

Operadores lógicos

- Os operadores lógicos são: <, <=, >, >=, == para igualdade exata e != para desigualdade.

```
> x = 36
```

```
> x == 36
```

```
[1] TRUE
```

```
> x != 36
```

```
[1] FALSE
```

```
> x != 18
```

```
[1] TRUE
```

- Você pode fazer operações do tipo “e” e “ou” em vetores usando os símbolos & e |, respectivamente.

```
> x>2 & x<4
```

```
[1] FALSE FALSE TRUE FALSE FALSE
```

```
> x<2 | x>4
```

```
[1] TRUE FALSE FALSE FALSE TRUE
```

Índices

- O R possui diferentes mecanismos de indexação.

Exemplos:

```
> x = 8:1
```

```
> x[5]          5º elemento de x
```

```
[1] 4
```

```
> x[5 : 7]      5º, 6º e 7º elementos de x
```

```
[1] 4 3 2
```

```
> x[c(1,3,8)]  1º, 3º e 8º elementos de x
```

```
[1] 8 6 1
```

```
> x[-2]        Todos os elementos de x, exceto o 2º
```

```
[1] 8 6 5 4 3 2 1
```

```
> x[x > 4]     Todos os elementos de x que são maiores que 4
```

```
[1] 8 7 6 5
```


Matrizes

- Matrizes são geradas no R usando a função `matrix`.

Exemplos:

```
> m = matrix(1:12,ncol=3)
```

```
> m
```

```
      [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
```

Neste exemplo foi construída uma matrix de 4 linhas e 3 colunas usando os números de 1 a 12.

```
> length(m)
```

```
[1] 12
```

```
> dim(m)
```

```
[1] 4 3
```

```
> nrow(m)
```

```
[1] 4
```

```
> ncol(m)
```

```
[1] 3
```

```
> m[1,2]
```

```
[1] 5
```

```
> m[2,2]
```

```
[1] 6
```

```
> m[,2]
```

```
[1] 5 6 7 8
```

```
> m[3,]
```

```
[1] 3 7 11
```

Navigation icons: back, forward, search, etc.

Operações com matrizes

- Todas as funções usuais operam em matrizes da mesma forma que operam com vetores, isto é, elemento por elemento.

Exemplo:

```
> m = matrix(1:4,ncol=2)
```

```
> p = matrix(4:7,ncol=2)
```

```
> m
```

```
      [,1] [,2]
[1,]    1    3
[2,]    2    4
```

```
> p
```

```
      [,1] [,2]
[1,]    4    6
[2,]    5    7
```

```
> m+p
```

```
      [,1] [,2]
[1,]    5    9
[2,]    7   11
```

- Na multiplicação de matrizes usamos o símbolo `%*%` e não `*`.

```
> m%*%p
```

```
      [,1] [,2]
[1,]   19   27
[2,]   28   40
```

```
> m*p
```

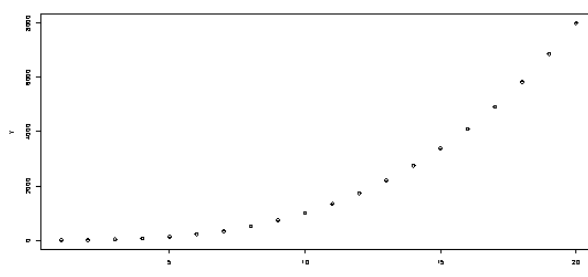
```
      [,1] [,2]
[1,]    4   18
[2,]   10   28
```

Navigation icons: back, forward, search, etc.

Gráficos

- A função `plot()` inicia um novo gráfico. Em sua forma mais simples, a função recebe valores de coordenada x e y .

```
> x = 1:20
> y = x^3
> plot(x,y)
```



Navigation icons: back, forward, search, etc.

Gráficos

- Gráficos com linhas ligando os pontos podem ser obtidos utilizando o argumento opcional `type="l"` na função `plot()`.

```
> plot(x,y,type='l')
```

- Há várias outras opções para os gráficos. Examine estes exemplos:

```
> plot(x,y,type='b')
> plot(x,y,type='o')
> plot(x,y,type='s')
> plot(x,y,type='c')
> plot(x,y,type='h')
```

Navigation icons: back, forward, search, etc.

Editando o gráfico

- Podemos editar um gráfico adicionando determinadas opções:

```
> plot(x,y,col="red")      Mudando a cor
> plot(x,y,pch=2)          Mudando o símbolo a ser plotado
> plot(x,y,type='l',lty=2) Mudando o tipo de linha
> plot(x,y,type='l',lwd=2) Mudando o tamanho da linha
```

- Podemos acrescentar algumas informações ao gráfico, tais como:

```
> plot(x,y)                Construindo o gráfico
> title('Meu 1º gráfico')  Título do gráfico
> points(10,4000)          Adicionando um ponto na posição (10,4000)
> text(5,4000,'Olá')       Adicionando um texto na posição (5,4000)
```

- E ainda escrever uma equação matemática dentro do gráfico:

```
> text(15,2000, expression(hat(y) == alpha + beta*x))
```

Editando o gráfico

- Podemos identificar pontos em um gráfico.

```
> x=rnorm(1000)  Gerando valores da distribuição normal
> plot(x)         Construindo o gráfico
> identify(x,n=4) Identificando 4 pontos do gráfico
```

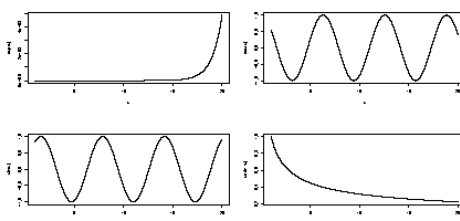
- Podemos adicionar linhas horizontais, verticais e segmentos de reta.

```
> abline(h=c(-2,2))      Adicionando duas linhas horizontais
> abline(v=200)           Adicionando uma linha vertical
> segments(0,-2,1000,2)  Adicionando um segmento de reta
```

Construção de vários gráficos

- Podemos construir vários gráficos em uma mesma janela.

```
> par(mfrow=c(2,2))      Janela gráfica 2x2
> x=seq(1,20,by=0.01)    Valores de x
> plot(x,exp(x),type='l')  Gráfico de f(x) = exp(x)
> plot(x,cos(x),type='l')
> plot(x,sin(x),type='l')
> plot(x,sqrt(1/x),type='l')
```



- Para retornar ao padrão com apenas um gráfico por janela digite

```
> par(mfrow=c(1,1))
```

Navigation icons: back, forward, search, etc.

Gráficos de funções

- Considere a seguinte função:

a) $f(x) = 1 - \frac{1}{x} \sin(x)$ para $0 \leq x \leq 50$

- A idéia básica é criar um vetor com valores das abcissas (valores de x) e calcular o valor da função (valores de f(x)) para cada elemento da função e depois fazer o gráfico unindo os pares de pontos.

```
> x1 = seq(0,50, l=101)
> y1 = 1 - (1/x1) * sin(x1)
> plot(x1, y1, type='l')
```

- Outra forma seria:

```
> plot(function(x) 1 - (1/x) * sin(x), 0, 50)
```

Navigation icons: back, forward, search, etc.

Gráficos de funções

- Considere agora a seguinte função:

$$a) f(x) = \frac{1}{\sqrt{2\pi} \times 5^2} \exp \left[-\frac{1}{2} \left(\frac{x-100}{5} \right)^2 \right] \quad \text{para } 85 \leq x \leq 115$$

```
> x2 = seq(80, 120, l=101)
> y2 = (1/sqrt(50*pi)) * exp(-0.02 * (x2-100)^2)
> plot(x2, y2, type='l')
```

- Note que esta função é a densidade da distribuição normal. O gráfico pode também ser obtido com:

```
> y2 = dnorm(x2, 100, 5)
> plot(x2, y2, type='l')
```

- Ou ainda:

```
> plot(function(x) dnorm(x, 100, 5), 85, 115)
```

Obs: Na janela gráfica escolha: Arquivo ┤ Salvar como ┤ Extensão do arquivo (pdf, png, bmp, jpeg...) para salvar o gráfico.

Gráficos estatísticos

- Composição de setores ("Pizza")

```
> par(mfrow=c(1,2))
> pie(rep(1, 12), col = rainbow(12), radius = 0.4)
> pie(rep(1, 12), col = rainbow(12), radius = 0.9)
```

- Barras

```
> x=rbinom(100,20,0.5)
> barplot(table(x))
```

- Boxplot

```
> y=rnorm(100)
> boxplot(y)
```

- Histograma

```
> y=rnorm(100)
> hist(y)
```

- Gráfico de pontos

```
> dotchart(y)
```

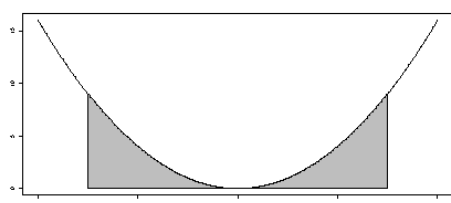
Integração numérica

- A função `integrate` é usada para integração numérica em uma dimensão.

$$a) I = \int_{-3}^3 x^2 dx$$

Para resolver a integral devemos criar uma função no R com a expressão da função que vamos integrar e esta deve ser passada para `integrate` conforme este exemplo:

```
> fx = function(x) x^2
> integrate(fx, -3, 3)
18 with absolute error < 2e-13
```



A integral corresponde à área mostrada no gráfico.

```
> x = seq(-4, 4, l=100)
> x2 = x^2
> plot(x, x2, ty='l')
> x = seq(-3, 3, l=100)
> x2 = x^2
> polygon(rbind(cbind(rev(x),0),
  cbind(x,x2)),col='gray')
```

Integração numérica em Estatística

- Sabemos que para distribuições contínuas de probabilidades a integral está associada a probabilidade em um intervalo. Seja $f(x)$ uma f.d.p. de uma variável contínua, então $P(a < x < b) = \int_a^b f(x) dx$.
- Exemplo: Seja $X \sim N(100, 81)$. Portanto, $f(x) = \frac{1}{9\sqrt{2\pi}} e^{-\frac{1}{162}(x-100)^2}$. A probabilidade $P(85 < x < 105)$ pode ser calculada das três formas diferentes mostradas a seguir.

```
> fx = function(x) {(1/(9*sqrt(2*pi))) * exp(-(1/162)*(x-100)^2)}
> integrate(fx, 85, 105)
0.6629523 with absolute error < 7.4e-15
```

```
> integrate(function(x) dnorm(x, 100, 9), 85, 105)
0.6629523 with absolute error < 7.4e-15
```

```
> pnorm(105, 100, 9) - pnorm(85, 100, 9)
[1] 0.6629523
```