



# Utilizando o R

# Introdução ao R

- R é um linguagem (ambiente) de programação para computação estatística e gráfica
  - Baseada na linguagem S (S-Plus)
- O ambiente R é flexível
  - Pode ser estendido através de pacotes
  - Open source e gratuito

# Introdução ao R

- Desenvolva um pacote e torne-se um colaborador do projeto
- Homepage do projeto
  - <http://cran.r-project.org/>
- O R é atualizado pelo menos a cada 6 meses
  - R version 2.7.1 has been released on 2008-06-23.

# Introdução ao R

- Códigos desenvolvidos em versões anteriores podem não funcionar em versões mais recentes
- Existem ferramentas GUI para o R
  - <http://www.sciviews.org/Tinn-R/index.html>

# Instalando o R

- O pacote R e os principais manuais podem ser obtidos a partir
  - <http://cran.r-project.org/>
  - ou em um *mirror*
- Siga as instruções para instalar o pacote base do R
  - Não esqueça de instalar os manuais on-line

# Primeiros passos

- Somando dois valores
  - `2 + 2`
- Atribuindo uma operação a uma variável
  - `a = 2 + 2`
  - `a <- 2 + 2`
- Para ver o resultado, digite a variável
  - `a`
  - `[1] 4`
- ou
  - `print (a)`

# Primeiros passos

- Entrando com dados com “c”
  - valores = c (6,7,4,3,2,0,0,6)
  - valores
- **OBS 1:** Dados são armazenados no “R” como um vetor
  - a=4
  - a[1]
- **OBS 2:** O índice da primeira posição do vetor é 1
- **OBS 3:** Comentários devem ser precedidos por #
- Aplicando funções a um conjunto de dados
  - length (valores)
  - min (valores)
  - max (valores)

# Primeiros passos

- Para sair

- >q ()

- Para ajuda

- >help (<nome do comando>)

- Se não sabe o nome corretamente

- >help.search (“<parte do nome>”)

# Gerando valores

```
vetor <- c(1:9)
```

```
vetor = (1:9)
```

```
vetor = (9:1)
```

```
seq(1, 9, by = 2)
```

```
seq(1, 9, by = pi )
```

```
seq(1, 9, by = 0.5)
```

```
rep(1,10)
```

```
rep(vetor,5)
```

```
rep (vetor, each = 5)
```

# Matrizes

```
m <- matrix(c(1,2,3,11,12,13), nrow = 2,  
ncol=3, byrow=TRUE, dimnames =  
list(c("linha1", "linha2"),  
      c("col1", "col2", "col3"))))
```

- `>m`

- Experimente retirar a cláusula `byrow`

# Matrizes

```
m <- matrix(1:9,nrow=3)
```

```
m
```

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

# Operações matemáticas

- entre números (incluindo constantes e variáveis simples)
  - soma: +
  - subtração: -
  - produto: \*
  - divisão: /
  - potência: ^
  - divisão inteira: % / %
  - mod: %%

# Operações matemáticas

- entre matrizes e números:
- se  $x$  é uma matriz, temos:
  - soma:  $x + 2$
  - subtração:  $x - 2$
  - produto:  $x * 2$
  - divisão:  $x / 2$
  - potência:  $x ^ 2$
  - divisão inteira:  $x \% / \% 2$
  - mod:  $x \% \% 2$

# Operações matemáticas

- entre matrizes:
- se  $x$  e  $y$  são matrizes, temos:
  - soma:  $x + y$
  - subtração:  $x - y$
  - produto:  $x \% * \% y$

# Operadores lógicos

- maior que:  $>$  ,  $\geq$
- menor que:  $<$  ,  $\leq$
- igual a:  $==$
- diferente de:  $\neq$
- para expressões:
  - and:  $\&$
  - or:  $|$
  - not:  $!$

# Operações básicas com vetores

- Somatório
  - `sum(<nome do vetor>)`
- Produtório
  - `prod(<nome do vetor>)`
- Tamanho do vetor
  - `length(<nome do vetor>)`
- Ordenação
  - `sort(<nome do vetor>)`

# Operações básicas

- Valor absoluto

- `abs(<nome da variável ou vetor>)`

- Raíz quadrada

- `sqrt(<nome da variável ou do vetor>)`

# Operações estatísticas

- Média aritmética

- `mean(<nome do vetor>)`

- Mediana

- `median(<nome do vetor>)`

- Desvio padrão

- `sd(<nome do vetor>, na.rm=FALSE)`

- se `na=TRUE`, os *missing values* serão removidos

# Operações estatísticas

## ■ Variância

- `var (<nome do vetor>, na.rm=FALSE)`

## ■ Covariância

- `cov(<nome do vetor 1>, <nome do vetor 2>)`

## ■ Correlação

- `cor(<nome do vetor 1>, <nome do vetor 2>)`

## ■ Regressão simples

- `lm (y~x)`

# *Missing values*

- Algumas operações podem retornar resultados “inexistentes”, chamados “*missing values*”
  - Inf: infinito positivo
  - -Inf: infinito negativo
  - NaN: “Not a Number”
  - NA: “Not Available”

# *Missing values*

- Existem funções para testar se algum desses valores foi retornado
  - `is.finite(x)`
  - `is.infinite(x)`
  - `is.nan(x)`
- onde `x` pode ser um vetor

# Geração de números

- Uniformes

- `runif(n, min, max)`

- Normais

- `rnorm(n, média, desvio padrão)`

- t-Student

- `rt(n, graus de liberdade)`

- Qui-Quadrado

- `rchisq(n, graus de liberdade)`

# Geração de gráficos

- `x=c(1:9)`
- `y=c(1:9)`
- `plot(x,y)`
- `plot(x,y,xlab="valores de x", ylab="valores de y")`
- `plot(x,y,type="l")`

# Geração de gráficos

- `x=rnorm(500,0,1)`
- `mean(x)`
- `median(x)`
- `sd(x)`
- `var(x)`
- `hist(x)`

# Lendo arquivos de dados

- O arquivo datafile.dat é composto pelos seguintes dados:

Tamanho lote de terra	homens/hora necessários para uma determinada atividade (arar, limpar, colher, plantar)
30	73
20	50
60	128
80	170
40	87
50	108
60	135
30	69
70	148
60	132

# Lendo arquivos de dados

- Ajustando o diretório de trabalho
  - `setwd("c:\temp")`
- Para ler arquivos com nomes de colunas na primeira linha, use
  - `dataset <- read.table("datafile.dat", header=TRUE)`
- Para obter os vetores
  - `dataset.$tamanho_lote`
  - `dataset$homens_hora`

# Lendo arquivos – parte II

- Ajustando o diretório de trabalho
  - `setwd("c:\temp")`
- Para ler arquivos com nomes de colunas na primeira linha, use
  - `dataset <- scan("datafile2.dat", what=list(x=0,y=0))`
- Para obter os vetores
  - `dataset$x`
  - `dataset$y`

# Lendo dados – parte III

```
stack.dat<-scan(what=list(tamanho_lote=0,homens_hora=0))
```

```
30          73
20          50
60         128
80         170
40          87
50         108
60         135
```

```
attach (stack.dat)
```

```
stack.dat$tamanho_lote
```

```
stack dat$homens_hora
```

- Os comandos acima podem ser inseridos em um arquivo ou diretamente a partir da linha de comando do R

# Lendo arquivos de dados

- O arquivo datafile3.dat é composto por dados que representam o nome do aluno e a natureza do ensino de nível médio

joao	publico
mario	privado
cristiano	privado
maria	publico
talita	privado
mariana	privado
adelia	publico
monique	publico
carlos	privado
bruno	privado

# Lendo arquivos – parte IV

- Ajustando o diretório de trabalho
  - `setwd("c:\temp")`
- Para ler arquivos com nomes de colunas na primeira linha, use
  - `dataset <- scan("datafile2.dat", what=list(x="",y=""))`
- Para obter os vetores
  - `dataset$x`
  - `dataset$y`

# Usando pacotes

- Carregando um pacote no workspace
  - `library (<nome do pacote>)`
- Site com pacotes de colaboradores
  - <http://cran.r-project.org/web/packages/>
- Instalando pacotes
  - `options (CRAN="http://cran.r-project.org")`
  - `Install.packages ("<nome do pacote>")`

# Conselhos úteis

- Ler os manuais no site do projeto CRAN
- Usar a página wiki do projeto CRAN
  - <http://wiki.r-project.org/>
- Usar <http://www.rseek.org/> ao invés do google
- Aprender com os erros
- ?lm dá uma ajuda sobre a função lm. Ler arquivos de help pode ajudar bastante
- Assine a lista do R
  - (<https://stat.ethz.ch/mailman/listinfo/r-help>)
- Crie seu script de bibliotecas personalizado

# Exercício 1

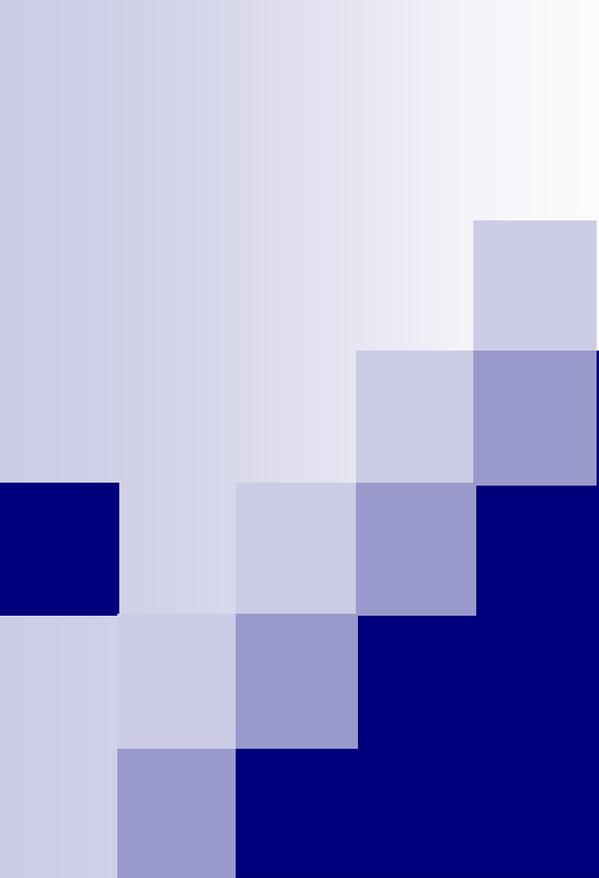
- `setwd(c:/temp)`
- `trabalho=scan("datafile2.dat", what=list (x=0,y=0))`
- `attach (trabalho)`
- `x=trabalho$x`
- `y=trabalho$y`
- `plot (x,y, xlab="valores de x", ylab="valores de y")`
- `ajuste=lm(y~x)`
- `summary (ajuste)`
- `abline (ajuste)`

# Exercício 2 (montgomery)

- Um motor de foguete é fabricado unindo um propelente de ignição a um propelente para manter o foguete em vôo. O poder da força da junção dos propelentes é uma característica de qualidade importante. Suspeita-se que o “poder” dessa força está relacionado com a “idade” do recipiente do propelente. O arquivo de dados datafile4.dat contém os dados relativos a 20 observações da força da junção comparados com a idade do recipiente do propelente.

# Exercício 2 (montgomery)

- Plote o gráfico de dispersão.
- Encontre covariância entre os dois vetores de dados
- Encontre a correlação entre os dois vetores de dados
- Encontre a média dos dois vetores de dados
- Encontre os valores de  $S_{xx}$  e  $S_{xy}$
- Encontre os estimadores  $\beta_0$  e  $\beta_1$
- Dado o modelo  $y = \beta_0 + \beta_1 x$ , encontre os valores ajustados para a variável resposta para cada observação da variável explicativa
- Encontre o resíduo  $e_i$  e verifique se  $\sum e_i = 0$
- Finalmente, aplique os métodos do exercício 1 ao arquivo datafile4.dat e compare os resultados.
- A tabela 2.2 do livro do montgomery apresenta os resultados acima.



# Utilizando o R