

Linguagem R

Aula 1

Arthur Alvim
afma@cin.ufpe.br



Introdução

Linguagem R

Sobre o R

R é uma linguagem e um ambiente para computação e gráficos estatísticos.



- Criada Ross Ihaka e Robert Gentleman na Universidade de Auckland, Nova Zelândia.
- Originada da Linguagem S (que foi desenvolvido nos laboratórios da Bell por John Chambers, entre outros).
- Desenvolvido atualmente pelo "R Development Core Team".

Sobre o R

R é uma linguagem e um ambiente para computação e gráficos estatísticos.



- Criada Ross Ihaka e Robert Gentleman na Universidade de Auckland, Nova Zelândia.
- Originada da Linguagem **S** (que foi desenvolvido nos laboratórios da Bell por John Chambers, entre outros).
- Desenvolvido atualmente pelo "R Development Core Team".

Sobre o R

R é uma linguagem e um ambiente para computação e gráficos estatísticos.



- Criada Ross Ihaka e Robert Gentleman na Universidade de Auckland, Nova Zelândia.
- Originada da Linguagem **S** (que foi desenvolvido nos laboratórios da Bell por John Chambers, entre outros).
- Desenvolvido atualmente pelo "R Development Core Team".

Sobre o R

R é uma linguagem e um ambiente para computação e gráficos estatísticos.



- Já acompanha uma variedade de técnicas estatísticas e gráficos.
- Altamente extensível. (através de pacotes)
- Roda numa grande variedade de sistemas UNIX. (Windows, Linux e Macintosh)
- Objetos do R podem ser acessados através de outras linguagens (C e Java).

Sobre o R

R é uma linguagem e um ambiente para computação e gráficos estatísticos.



- Já acompanha uma variedade de técnicas estatísticas e gráficos.
- Altamente extensível. (através de pacotes)
- Roda numa grande variedade de sistemas UNIX. (Windows, Linux e Macintosh)
- Objetos do R podem ser acessados através de outras linguagens (C e Java).

Sobre o R

R é uma linguagem e um ambiente para computação e gráficos estatísticos.



- Já acompanha uma variedade de técnicas estatísticas e gráficos.
- Altamente extensível. (através de pacotes)
- Roda numa grande variedade de sistemas UNIX. (Windows, Linux e Macintosh)
- Objetos do R podem ser acessados através de outras linguagens (C e Java).

Sobre o R

R é uma linguagem e um ambiente para computação e gráficos estatísticos.



- Já acompanha uma variedade de técnicas estatísticas e gráficos.
- Altamente extensível. (através de pacotes)
- Roda numa grande variedade de sistemas UNIX. (Windows, Linux e Macintosh)
- Objetos do R podem ser acessados através de outras linguagens (C e Java).

Sobre o R

- Site do projeto Comprehensive R Archive Network (CRAN)
<http://cran.r-project.org/>
- Versão Atual
R 2.13.0 (2011-04-13)
- WikiBook do R
http://en.wikibooks.org/wiki/R_Programming
- Livro Introdutório R
<http://cran.r-project.org/doc/manuals/R-intro.pdf>
- O R é atualizado pelo menos a cada 6 meses.
- Códigos desenvolvidos em versões anteriores podem não funcionar em versões mais recentes.

Sobre o R

- Site do projeto Comprehensive R Archive Network (CRAN)
<http://cran.r-project.org/>
- Versão Atual
R 2.13.0 (2011-04-13)
- WikiBook do R
http://en.wikibooks.org/wiki/R_Programming
- Livro Introdutório R
<http://cran.r-project.org/doc/manuals/R-intro.pdf>
- O R é atualizado pelo menos a cada 6 meses.
- Códigos desenvolvidos em versões anteriores podem não funcionar em versões mais recentes.

Sobre o R

- Site do projeto Comprehensive R Archive Network (CRAN)
<http://cran.r-project.org/>
- Versão Atual
R 2.13.0 (2011-04-13)
- WikiBook do R
http://en.wikibooks.org/wiki/R_Programming
- Livro Introdutório R
<http://cran.r-project.org/doc/manuals/R-intro.pdf>
- O R é atualizado pelo menos a cada 6 meses.
- Códigos desenvolvidos em versões anteriores podem não funcionar em versões mais recentes.

Sobre o R

- Site do projeto Comprehensive R Archive Network (CRAN)
<http://cran.r-project.org/>
- Versão Atual
R 2.13.0 (2011-04-13)
- WikiBook do R
http://en.wikibooks.org/wiki/R_Programming
- Livro Introdutório R
<http://cran.r-project.org/doc/manuals/R-intro.pdf>
- O R é atualizado pelo menos a cada 6 meses.
- Códigos desenvolvidos em versões anteriores podem não funcionar em versões mais recentes.

Sobre o R

- Site do projeto Comprehensive R Archive Network (CRAN)
<http://cran.r-project.org/>
- Versão Atual
R 2.13.0 (2011-04-13)
- WikiBook do R
http://en.wikibooks.org/wiki/R_Programming
- Livro Introdutório R
<http://cran.r-project.org/doc/manuals/R-intro.pdf>
- O R é atualizado pelo menos a cada 6 meses.
- Códigos desenvolvidos em versões anteriores podem não funcionar em versões mais recentes.

Sobre o R

- Site do projeto Comprehensive R Archive Network (CRAN)
<http://cran.r-project.org/>
- Versão Atual
R 2.13.0 (2011-04-13)
- WikiBook do R
http://en.wikibooks.org/wiki/R_Programming
- Livro Introdutório R
<http://cran.r-project.org/doc/manuals/R-intro.pdf>
- O R é atualizado pelo menos a cada 6 meses.
- Códigos desenvolvidos em versões anteriores podem não funcionar em versões mais recentes.

Por que usar o R?

- É **Software Livre**. O código fonte segue *GNU General Public License*.
- Principal "concorrente" de outros softwares estatísticos pagos como SPSS, SAS/STAT.
- Bastante documentado e utilizado na comunidade científica
- Roda nos sistemas mais populares atualmente (Windows, Linux e Macintosh).
- É uma boa alternativa para software de computação inteligente pagos como o MatLab.
- Grande comunidade de desenvolvedores.
- Linguagem bem documentada.

Por que usar o R?

- É **Software Livre**. O código fonte segue *GNU General Public License*.
- Principal "concorrente" de outros softwares estatísticos pagos como SPSS, SAS/STAT.
- Bastante documentado e utilizado na comunidade científica
- Roda nos sistemas mais populares atualmente (Windows, Linux e Macintosh).
- É uma boa alternativa para software de computação inteligente pagos como o MatLab.
- Grande comunidade de desenvolvedores.
- Linguagem bem documentada.

Por que usar o R?

- É **Software Livre**. O código fonte segue *GNU General Public License*.
- Principal "concorrente" de outros softwares estatísticos pagos como SPSS, SAS/STAT.
- Bastante documentado e utilizado na comunidade científica
- Roda nos sistemas mais populares atualmente (Windows, Linux e Macintosh).
- É uma boa alternativa para software de computação inteligente pagos como o MatLab.
- Grande comunidade de desenvolvedores.
- Linguagem bem documentada.

Por que usar o R?

- É **Software Livre**. O código fonte segue *GNU General Public License*.
- Principal "concorrente" de outros softwares estatísticos pagos como SPSS, SAS/STAT.
- Bastante documentado e utilizado na comunidade científica
- Roda nos sistemas mais populares atualmente (Windows, Linux e Macintosh).
- É uma boa alternativa para software de computação inteligente pagos como o MatLab.
- Grande comunidade de desenvolvedores.
- Linguagem bem documentada.

Por que usar o R?

- É **Software Livre**. O código fonte segue *GNU General Public License*.
- Principal "concorrente" de outros softwares estatísticos pagos como SPSS, SAS/STAT.
- Bastante documentado e utilizado na comunidade científica
- Roda nos sistemas mais populares atualmente (Windows, Linux e Macintosh).
- É uma boa alternativa para software de computação inteligente pagos como o MatLab.
- Grande comunidade de desenvolvedores.
- Linguagem bem documentada.

Por que usar o R?

- É **Software Livre**. O código fonte segue *GNU General Public License*.
- Principal "concorrente" de outros softwares estatísticos pagos como SPSS, SAS/STAT.
- Bastante documentado e utilizado na comunidade científica
- Roda nos sistemas mais populares atualmente (Windows, Linux e Macintosh).
- É uma boa alternativa para software de computação inteligente pagos como o MatLab.
- Grande comunidade de desenvolvedores.
- Linguagem bem documentada.

Por que usar o R?

- É **Software Livre**. O código fonte segue *GNU General Public License*.
- Principal "concorrente" de outros softwares estatísticos pagos como SPSS, SAS/STAT.
- Bastante documentado e utilizado na comunidade científica
- Roda nos sistemas mais populares atualmente (Windows, Linux e Macintosh).
- É uma boa alternativa para software de computação inteligente pagos como o MatLab.
- Grande comunidade de desenvolvedores.
- Linguagem bem documentada.

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Graphical User Interface para o R:

- JGR (Java Gui for R) (<http://rforge.net/JGR/files>)
- RKWard (<http://rkwad.sourceforge.net/>)
- SciViews-R (<http://www.sciviews.org/SciViews-R/>)
- Rcmdr (R Commander)
(<http://cran.r-project.org/web/packages/Rcmdr/index.html>)
- Tinn-R GUI/Editor
(<http://www.sciviews.org/Tinn-R/index.html>)

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Graphical User Interface para o R:

- JGR (Java Gui for R) (<http://rforge.net/JGR/files>)
- RKWard (<http://rkwad.sourceforge.net/>)
- SciViews-R (<http://www.sciviews.org/SciViews-R/>)
- Rcmdr (R Commander)
(<http://cran.r-project.org/web/packages/Rcmdr/index.html>)
- Tinn-R GUI/Editor
(<http://www.sciviews.org/Tinn-R/index.html>)

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Graphical User Interface para o R:

- JGR (Java Gui for R) (<http://rforge.net/JGR/files>)
- RKWard (<http://rkwad.sourceforge.net/>)
- SciViews-R (<http://www.sciviews.org/SciViews-R/>)
- Rcmdr (R Commander)
(<http://cran.r-project.org/web/packages/Rcmdr/index.html>)
- Tinn-R GUI/Editor
(<http://www.sciviews.org/Tinn-R/index.html>)

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Graphical User Interface para o R:

- JGR (Java Gui for R) (<http://rforge.net/JGR/files>)
- RKWard (<http://rkwad.sourceforge.net/>)
- SciViews-R (<http://www.sciviews.org/SciViews-R/>)
- Rcmdr (R Commander)
(<http://cran.r-project.org/web/packages/Rcmdr/index.html>)
- Tinn-R GUI/Editor
(<http://www.sciviews.org/Tinn-R/index.html>)

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Graphical User Interface para o R:

- JGR (Java Gui for R) (<http://rforge.net/JGR/files>)
- RKWard (<http://rkwad.sourceforge.net/>)
- SciViews-R (<http://www.sciviews.org/SciViews-R/>)
- Rcmdr (R Commander)
(<http://cran.r-project.org/web/packages/Rcmdr/index.html>)
- Tinn-R GUI/Editor
(<http://www.sciviews.org/Tinn-R/index.html>)

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Graphical User Interface para o R:

- JGR (Java Gui for R) (<http://rforge.net/JGR/files>)
- RKWard (<http://rkwad.sourceforge.net/>)
- SciViews-R (<http://www.sciviews.org/SciViews-R/>)
- Rcmdr (R Commander)
(<http://cran.r-project.org/web/packages/Rcmdr/index.html>)
- Tinn-R GUI/Editor
(<http://www.sciviews.org/Tinn-R/index.html>)

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Graphical User Interface para o R:

- JGR (Java Gui for R) (<http://rforge.net/JGR/files>)
- RKWard (<http://rkwad.sourceforge.net/>)
- SciViews-R (<http://www.sciviews.org/SciViews-R/>)
- Rcmdr (R Commander)
(<http://cran.r-project.org/web/packages/Rcmdr/index.html>)
- Tinn-R GUI/Editor
(<http://www.sciviews.org/Tinn-R/index.html>)

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Pode-se utilizar qualquer editor de texto: Bluefish, Crimson Editor, ConTEXT, Eclipse, Emacs, VIM, Tinn-R, Geany, jEdit, Kate, TextMate, TextWrangler, gEdit, SciTE, WinEdt (R Package RWinEdt), notepad++ e SciViews etc.

Existem plugins para IDE Eclipse e para o VIM (vim-r-plugin2). As funcionalidade do R também são acessíveis a várias linguagens de script como Python (RPy), Perl (Statistics::R) e Ruby (rsruby rubygem).

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Pode-se utilizar qualquer editor de texto: Bluefish, Crimson Editor, ConTEXT, Eclipse, Emacs, VIM, Tinn-R, Geany, jEdit, Kate, TextMate, TextWrangler, gEdit, SciTE, WinEdt (R Package RWinEdt), notepad++ e SciViews etc.

Existem plugins para IDE Eclipse e para o VIM (vim-r-plugin2)
As funcionalidade do R também são acessíveis a várias linguagens de script como Python (RPy), Perl (Statistics::R) e Ruby (rsruby rubygem).

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Pode-se utilizar qualquer editor de texto: Bluefish, Crimson Editor, ConTEXT, Eclipse, Emacs, VIM, Tinn-R, Geany, jEdit, Kate, TextMate, TextWrangler, gEdit, SciTE, WinEdt (R Package RWinEdt), notepad++ e SciViews etc.

Existem plugins para IDE Eclipse e para o VIM (vim-r-plugin2)

As funcionalidade do R também são acessíveis a várias linguagens de script como Python (RPy), Perl (Statistics::R) e Ruby (rsruby rubygem).

Ferramentas

R utiliza uma interface de linha de comando. O usuário digita comandos em um prompt (>) e cada comando é executado por vez.

Pode-se utilizar qualquer editor de texto: Bluefish, Crimson Editor, ConTEXT, Eclipse, Emacs, VIM, Tinn-R, Geany, jEdit, Kate, TextMate, TextWrangler, gEdit, SciTE, WinEdt (R Package RWinEdt), notepad++ e SciViews etc.

Existem plugins para IDE Eclipse e para o VIM (vim-r-plugin2)

As funcionalidade do R também são acessíveis a várias linguagens de script como Python (RPy), Perl (Statistics::R) e Ruby (rsruby rubygem).

Instalação

A instalação do R não é nada complicada.

Para sistemas Linux em geral basta baixar o arquivo fonte tar.gz compilar e rodar.

Para o Ubuntu em especial você pode baixar através de repositórios via apt-get:

```
sudo apt-get update # Atualiza listas de pacotes do Ubuntu
sudo apt-get install r-base # Instala o R com seus pacotes básicos
sudo apt-get install r-base-dev # Instala mais pacotes
sudo apt-get install rkwad # Instala uma GUI para R
```

Para rodá-lo no terminal basta digitar R ou procurar o ícone do RKWard nas aplicações.

Instalação

A instalação do R não é nada complicada.

Para sistemas Linux em geral basta baixar o arquivo fonte tar.gz compilar e rodar.

Para o Ubuntu em especial você pode baixar através de repositórios via apt-get:

```
sudo apt-get update # Atualiza listas de pacotes do Ubuntu
sudo apt-get install r-base # Instala o R com seus pacotes básicos
sudo apt-get install r-base-dev # Instala mais pacotes
sudo apt-get install rkwad # Instala uma GUI para R
```

Para rodá-lo no terminal basta digitar R ou procurar o ícone do RKWard nas aplicações.

Instalação

A instalação do R não é nada complicada.

Para sistemas Linux em geral basta baixar o arquivo fonte tar.gz compilar e rodar.

Para o Ubuntu em especial você pode baixar através de repositórios via apt-get:

```
sudo apt-get update # Atualiza listas de pacotes do Ubuntu
sudo apt-get install r-base # Instala o R com seus pacotes básicos
sudo apt-get install r-base-dev # Instala mais pacotes
sudo apt-get install rkwad # Instala uma GUI para R
```

Para rodá-lo no terminal basta digitar R ou procurar o ícone do RKWard nas aplicações.

Instalação

A instalação do R não é nada complicada.

Para sistemas Linux em geral basta baixar o arquivo fonte tar.gz compilar e rodar.

Para o Ubuntu em especial você pode baixar através de repositórios via apt-get:

```
sudo apt-get update # Atualiza listas de pacotes do Ubuntu
sudo apt-get install r-base # Instala o R com seus pacotes básicos
sudo apt-get install r-base-dev # Instala mais pacotes
sudo apt-get install rkwad # Instala uma GUI para R
```

Para rodá-lo no terminal basta digitar R ou procurar o ícone do RKWard nas aplicações.

Instalação

A instalação do R não é nada complicada.

Para sistemas Linux em geral basta baixar o arquivo fonte tar.gz compilar e rodar.

Para o Ubuntu em especial você pode baixar através de repositórios via apt-get:

```
sudo apt-get update # Atualiza listas de pacotes do Ubuntu
sudo apt-get install r-base # Instala o R com seus pacotes básicos
sudo apt-get install r-base-dev # Instala mais pacotes
sudo apt-get install rkwad # Instala uma GUI para R
```

Para rodá-lo no terminal basta digitar R ou procurar o ícone do RKWard nas aplicações.

Instalação

No Windows é uma instalação “NNF” (Next-Next-Finish!).

Baixar

<http://cran.r-project.org/bin/windows/base/R-2.13.0-win.exe>

Ao final pode-se instalar uma GUI como o RKWard ou o Tinn-R.

No Macintosh a instalação é feita através de um pacote (.pkg).

Baixar

<http://cran.r-project.org/bin/macosx/R-2.13.0.pkg>

Este pacote já acompanha uma GUI padrão. Contudo pode-se usar outra GUI como o RKWard ou integração com outros editores de texto.

Instalação

No Windows é uma instalação “NNF” (Next-Next-Finish!).

Baixar

<http://cran.r-project.org/bin/windows/base/R-2.13.0-win.exe>

Ao final pode-se instalar uma GUI como o RKWard ou o Tinn-R.

No Macintosh a instalação é feita através de um pacote (.pkg).

Baixar

<http://cran.r-project.org/bin/macosx/R-2.13.0.pkg>

Este pacote já acompanha uma GUI padrão. Contudo pode-se usar outra GUI como o RKWard ou integração com outros editores de texto.

Instalação

No Windows é uma instalação “NNF” (Next-Next-Finish!).

Baixar

<http://cran.r-project.org/bin/windows/base/R-2.13.0-win.exe>

Ao final pode-se instalar uma GUI como o RKWard ou o Tinn-R.

No Macintosh a instalação é feita através de um pacote (.pkg).

Baixar

<http://cran.r-project.org/bin/macosx/R-2.13.0.pkg>

Este pacote já acompanha uma GUI padrão. Contudo pode-se usar outra GUI como o RKWard ou integração com outros editores de texto.

Instalação

No Windows é uma instalação “NNF” (Next-Next-Finish!).

Baixar

<http://cran.r-project.org/bin/windows/base/R-2.13.0-win.exe>

Ao final pode-se instalar uma GUI como o RKWard ou o Tinn-R.

No Macintosh a instalação é feita através de um pacote (.pkg).

Baixar

<http://cran.r-project.org/bin/macosx/R-2.13.0.pkg>

Este pacote já acompanha uma GUI padrão. Contudo pode-se usar outra GUI como o RKWard ou integração com outros editores de texto.

Primeiros Passos

Primeiros Passos

Primeiros Passos

- É orientada a objeto. Isto significa que tudo pode ser armazenado como um objeto do R.
- Pode ser considerada uma linguagem de matriz.
- Case-sensitive. Logo, objeto \neq OBJETO
- Vários tipos de dados: vetores, matrizes, dataframes e listas.
- O símbolo de atribuição é “<-”. Pode-se utilizar o clássico “=” em alguns casos.

```
> a <- 2
> a = 2
> # isto é um comentário
> # comandos são separados por ';' ou um nova linha
> a <- 1; b <- 2
> q() # para sair do interpretador do R
```

Primeiros Passos

- É orientada a objeto. Isto significa que tudo pode ser armazenado como um objeto do R.
- Pode ser considerada uma linguagem de matriz.
- Case-sensitive. Logo, objeto \neq OBJETO
- Vários tipos de dados: vetores, matrizes, dataframes e listas.
- O símbolo de atribuição é “<-”. Pode-se utilizar o clássico “=” em alguns casos.

```
> a <- 2
> a = 2
> # isto é um comentário
> # comandos são separados por ``;`` ou um nova linha
> a <- 1; b <- 2
> q() # para sair do interpretador do R
```

Primeiros Passos

- É orientada a objeto. Isto significa que tudo pode ser armazenado como um objeto do R.
- Pode ser considerada uma linguagem de matriz.
- Case-sensitive. Logo, objeto \neq OBJETO
- Vários tipos de dados: vetores, matrizes, dataframes e listas.
- O símbolo de atribuição é "<-". Pode-se utilizar o clássico "=" em alguns casos.

```
> a <- 2
> a = 2
> # isto é um comentário
> # comandos são separados por ';' ou um nova linha
> a <- 1; b <- 2
> q() # para sair do interpretador do R
```

Primeiros Passos

- É orientada a objeto. Isto significa que tudo pode ser armazenado como um objeto do R.
- Pode ser considerada uma linguagem de matriz.
- Case-sensitive. Logo, objeto \neq OBJETO
- Vários tipos de dados: vetores, matrizes, dataframes e listas.
- O símbolo de atribuição é " $<-$ ". Pode-se utilizar o clássico " $=$ " em alguns casos.

```
> a <- 2
> a = 2
> # isto é um comentário
> # comandos são separados por ';' ou um nova linha
> a <- 1; b <- 2
> q() # para sair do interpretador do R
```

Primeiros Passos

- É orientada a objeto. Isto significa que tudo pode ser armazenado como um objeto do R.
- Pode ser considerada uma linguagem de matriz.
- Case-sensitive. Logo, objeto \neq OBJETO
- Vários tipos de dados: vetores, matrizes, dataframes e listas.
- O símbolo de atribuição é “<-”. Pode-se utilizar o clássico “=” em alguns casos.

```
> a <- 2
> a = 2
> # isto é um comentário
> # comandos são separados por ';' ou um nova linha
> a <- 1; b <- 2
> q() # para sair do interpretador do R
```


Primeiros Passos

- É orientada a objeto. Isto significa que tudo pode ser armazenado como um objeto do R.
- Pode ser considerada uma linguagem de matriz.
- Case-sensitive. Logo, objeto \neq OBJETO
- Vários tipos de dados: vetores, matrizes, dataframes e listas.
- O símbolo de atribuição é “<-”. Pode-se utilizar o clássico “=” em alguns casos.

```
> a <- 2
> a = 2
> # isto é um comentário
> # comandos são separados por ';' ou um nova linha
> a <- 1; b <- 2
> q() # para sair do interpretador do R
```

Pedindo Ajuda

Pedindo Ajuda no R

```
help.start()      # Ajuda em geral
help(exemplo)    # Ajuda sobre a função exemplo
?exemplo         # mesma coisa
apropos("exemplo") # Lista todas funções com a string "exemplo"
example(foo)     # show an example of function foo
```

Pedindo Ajuda

Pedindo Ajuda no R

```
help.start()      # Ajuda em geral
help(exemplo)    # Ajuda sobre a função exemplo
?exemplo # mesma coisa
apropos("exemplo") # Lista todas funções com a string "exemplo"
example(foo)     # show an example of function foo
```

Rodando o R em Batch

Vamos criar um script qualquer. Por exemplo:

```
a = 2
b = 4
a + b # resultado da soma de a + b
```

Vamos salvar como **script.r** .
Em seguida para rodar o script podemos:

```
# No Linux e Macintosh
R CMD BATCH script.r saida.txt

# No Windows (ajuste o caminho para R.exe)
"C:\Path to\bin\R.exe" CMD BATCH
  --vanilla --slave "C:\Path to\script.R"
```

Rodando o R em Batch

Vamos criar um script qualquer. Por exemplo:

```
a = 2
b = 4
a + b # resultado da soma de a + b
```

Vamos salvar como **script.r** .
Em seguida para rodar o script podemos:

```
# No Linux e Macintosh
R CMD BATCH script.r saida.txt

# No Windows (ajuste o caminho para R.exe)
"C:\Path to\bin\R.exe" CMD BATCH
  --vanilla --slave "C:\Path to\script.R"
```

Rodando o R em Batch

Vamos criar um script qualquer. Por exemplo:

```
a = 2
b = 4
a + b # resultado da soma de a + b
```

Vamos salvar como **script.r** .
Em seguida para rodar o script podemos:

```
# No Linux e Macintosh
R CMD BATCH script.r saida.txt

# No Windows (ajuste o caminho para R.exe)
"C:\Path to\bin\R.exe" CMD BATCH
  --vanilla --slave "C:\Path to\script.R"
```

Área de Trabalho

Conhecendo a sua área de trabalho

```
getwd() # imprime o diretório de trabalho atual - cwd

setwd("c:/docs/mydir") # mudar o diretório de trabalho

dir() # Lista arquivos do cwd.

ls(all=TRUE) # lista objetos todos objetos

rm() # remove objetos

history() # Mostra os 25 últimos comandos

savehistory(file="historico") # Salva seu histórico de comandos
# default is ".Rhistory"

loadhistory(file="historico") # Carrega seu histórico de comandos
```

Área de Trabalho

Conhecendo a sua área de trabalho

```
getwd() # imprime o diretório de trabalho atual - cwd

setwd("c:/docs/mydir") # mudar o diretório de trabalho

dir() # Lista arquivos do cwd.

ls(all=TRUE) # lista objetos todos objetos

rm() # remove objetos

history() # Mostra os 25 últimos comandos

savehistory(file="historico") # Salva seu histórico de comandos
# default is ".Rhistory"

loadhistory(file="historico") # Carrega seu histórico de comandos
```


Área de Trabalho

Para salvar sua área de trabalho

```
save.image() # Salva tudo que foi criado em um arquivo .RData no cwd
save(object list,file="objetos.RData") # Para salvar objetos específicos
load("objetos.RData") # Carregar os objetos na sessão corrente
```

Nota Importante para Usuários Windows

O R se confunde se você usar um caminho como abaixo:

“c:\mydocuments\myfile.txt” Isto se deve ao R enxergar a

“\” com outra funcionalidade. Use:

“c:\\mydocuments\\myfile.txt” ou “c:/mydocuments/myfile.txt”

Área de Trabalho

Para salvar sua área de trabalho

```
save.image() # Salva tudo que foi criado em um arquivo .RData no cwd
save(object list,file="objetos.RData") # Para salvar objetos específicos
load("objetos.RData") # Carregar os objetos na sessão corrente
```

Nota Importante para Usuários Windows

O R se confunde se você usar um caminho como abaixo:

“c:\mydocuments\myfile.txt” Isto se deve ao R enxergar a

“\” com outra funcionalidade. Use:

“c:\\mydocuments\\myfile.txt” ou “c:/mydocuments/myfile.txt”

Área de Trabalho

Para salvar sua área de trabalho

```
save.image() # Salva tudo que foi criado em um arquivo .RData no cwd
save(object list,file="objetos.RData") # Para salvar objetos específicos
load("objetos.RData") # Carregar os objetos na sessão corrente
```

Nota Importante para Usuários Windows

O R se confunde se você usar um caminho como abaixo:

“c:\mydocuments\myfile.txt” Isto se deve ao R enxergar a
“\” com outra funcionalidade. Use:

“c:\\mydocuments\\myfile.txt” ou “c:/mydocuments/myfile.txt”

Entrada e Saída Padrão

Para introduzir um script no interpretador:

```
source("script.r") # rodará o script.r
```

Para direcionar as saídas:

```
sink("saida.txt", append=FALSE, split=FALSE) # direciona a saída para  
sink() # retorna a saída padrão para o terminal
```

A opção *append* controla se devemos sobrescrever um arquivo de saída ou continuar escrevendo nele. Já a *split* define se devemos dividir a saída padrão para a tela e o arquivo de saída ou não.

Entrada e Saída Padrão

Para introduzir um script no interpretador:

```
source("script.r") # rodará o script.r
```

Para direcionar as saídas:

```
sink("saida.txt", append=FALSE, split=FALSE) # direciona a saída para  
sink() # retorna a saída padrão para o terminal
```

A opção *append* controla se devemos sobrescrever um arquivo de saída ou continuar escrevendo nele. Já a *split* define se devemos dividir a saída padrão para a tela e o arquivo de saída ou não.

Entrada e Saída Padrão

Para introduzir um script no interpretador:

```
source("script.r") # rodará o script.r
```

Para direcionar as saídas:

```
sink("saida.txt", append=FALSE, split=FALSE) # direciona a saída para  
sink() # retorna a saída padrão para o terminal
```

A opção *append* controla se devemos sobrescrever um arquivo de saída ou continuar escrevendo nele. Já a *split* define se devemos dividir a saída padrão para a tela e o arquivo de saída ou não.

Entrada e Saída Padrão

Para introduzir um script no interpretador:

```
source("script.r") # rodará o script.r
```

Para direcionar as saídas:

```
sink("saida.txt", append=FALSE, split=FALSE) # direciona a saída para  
sink() # retorna a saída padrão para o terminal
```

A opção *append* controla se devemos sobrescrever um arquivo de saída ou continuar escrevendo nele. Já a *split* define se devemos dividir a saída padrão para a tela e o arquivo de saída ou não.

Saída de Gráficos

Para salvar a saída gráfica:

- `pdf("mygraph.pdf")` ⇒ pdf file
- `png("mygraph.png")` ⇒ png file
- `jpeg("mygraph.jpg")` ⇒ jpeg file
- `bmp("mygraph.bmp")` ⇒ bmp file
- `postscript("mygraph.ps")` ⇒ postscript file

```
jpeg("c:/mygraphs/myplot.jpg") # Salvando saída de um gráfico em jpeg.  
plot(x)  
dev.off()
```


Saída de Gráficos

Para salvar a saída gráfica:

- `pdf("mygraph.pdf")` ⇒ pdf file
- `png("mygraph.png")` ⇒ png file
- `jpeg("mygraph.jpg")` ⇒ jpeg file
- `bmp("mygraph.bmp")` ⇒ bmp file
- `postscript("mygraph.ps")` ⇒ postscript file

```
jpeg("c:/mygraphs/myplot.jpg") # Salvando saída de um gráfico em jpeg.  
plot(x)  
dev.off()
```

Saída de Gráficos

Para salvar a saída gráfica:

- `pdf("mygraph.pdf")` ⇒ pdf file
- `png("mygraph.png")` ⇒ png file
- `jpeg("mygraph.jpg")` ⇒ jpeg file
- `bmp("mygraph.bmp")` ⇒ bmp file
- `postscript("mygraph.ps")` ⇒ postscript file

```
jpeg("c:/mygraphs/myplot.jpg") # Salvando saída de um gráfico em jpeg.  
plot(x)  
dev.off()
```

Saída de Gráficos

Para salvar a saída gráfica:

- `pdf("mygraph.pdf")` ⇒ pdf file
- `png("mygraph.png")` ⇒ png file
- `jpeg("mygraph.jpg")` ⇒ jpeg file
- `bmp("mygraph.bmp")` ⇒ bmp file
- `postscript("mygraph.ps")` ⇒ postscript file

```
jpeg("c:/mygraphs/myplot.jpg") # Salvando saída de um gráfico em jpeg.  
plot(x)  
dev.off()
```

Saída de Gráficos

Para salvar a saída gráfica:

- `pdf("mygraph.pdf")` ⇒ pdf file
- `png("mygraph.png")` ⇒ png file
- `jpeg("mygraph.jpg")` ⇒ jpeg file
- `bmp("mygraph.bmp")` ⇒ bmp file
- `postscript("mygraph.ps")` ⇒ postscript file

```
jpeg("c:/mygraphs/myplot.jpg") # Salvando saída de um gráfico em jpeg.  
plot(x)  
dev.off()
```

Saída de Gráficos

Para salvar a saída gráfica:

- `pdf("mygraph.pdf")` ⇒ pdf file
- `png("mygraph.png")` ⇒ png file
- `jpeg("mygraph.jpg")` ⇒ jpeg file
- `bmp("mygraph.bmp")` ⇒ bmp file
- `postscript("mygraph.ps")` ⇒ postscript file

```
jpeg("c:/mygraphs/myplot.jpg") # Salvando saída de um gráfico em jpeg.  
plot(x)  
dev.off()
```

Saída de Gráficos

Para salvar a saída gráfica:

- `pdf("mygraph.pdf")` ⇒ pdf file
- `png("mygraph.png")` ⇒ png file
- `jpeg("mygraph.jpg")` ⇒ jpeg file
- `bmp("mygraph.bmp")` ⇒ bmp file
- `postscript("mygraph.ps")` ⇒ postscript file

```
jpeg("c:/mygraphs/myplot.jpg") # Salvando saída de um gráfico em jpeg.  
plot(x)  
dev.off()
```

Instalando Pacotes

```
.libPaths() # get library location  
library()   # see all packages installed  
search()    # see packages currently loaded  
detach("package:nomepacote") #remove pacote
```

Instalando pacotes de dados do livro "Introductory Statistics with R"

```
chooseCRANmirror() # Escolha o repositório  
install.packages("ISwR", dependencies = TRUE)  
library (ISwR) # Carrega pacote  
install.packages("caminho.do.arquivo/pacote.tar.gz", repos=NULL)  
# para pacotes que já foram baixados e se encontram localmente
```

Instalando Pacotes

```
.libPaths() # get library location  
library()  # see all packages installed  
search()   # see packages currently loaded  
detach("package:nomepacote") #remove pacote
```

Instalando pacotes de dados do livro "Introductory Statistics with R"

```
chooseCRANmirror() # Escolha o repositório  
install.packages("ISwR", dependencies = TRUE)  
library (ISwR) # Carrega pacote  
install.packages("caminho.do.arquivo/pacote.tar.gz", repos=NULL)  
# para pacotes que já foram baixados e se encontram localmente
```


Instalando Pacotes

```
.libPaths() # get library location  
library()   # see all packages installed  
search()    # see packages currently loaded  
detach("package:nomepacote") #remove pacote
```

Instalando pacotes de dados do livro "Introductory Statistics with R"

```
chooseCRANmirror() # Escolha o repositório  
install.packages("ISwR", dependencies = TRUE)  
library (ISwR) # Carrega pacote  
install.packages("caminho.do.arquivo/pacote.tar.gz", repos=NULL)  
# para pacotes que já foram baixados e se encontram localmente
```

Instalando Pacotes

Para instalar todos os pacotes:

```
# options (CRAN="http://cran.r-project.org") Repositório principal

options(CRAN = "http://www.vps.fmvz.usp.br/CRAN/")
# Escolhe-se o Repositório CRAN da USP

install.packages(available.packages() [,1])
# Instala todos os pacotes disponíveis lá
```

Instalando Pacotes

Para instalar todos os pacotes:

```
# options (CRAN="http://cran.r-project.org") Repositório principal

options(CRAN = "http://www.vps.fmvz.usp.br/CRAN/")
# Escolhe-se o Repositório CRAN da USP

install.packages(available.packages()[,1])
# Instala todos os pacotes disponíveis lá
```

Comandos simples

Alguns comandos para mostrar a saída de uma variável

```
> a = 2 + 2
> a <- 2 + 2
> 4 -> b
> a
[1] 4
> print(b)
[1] 4
```

Importante mostrar como a atribuição pode ser feita nos dois sentidos.

```
> valores = c(6, 7, 4, 3, 2, 0, 0, 6)
> valores[4]
[1] 3
> length(valores)
[1] 8
> min(valores)
[1] 0
> max(valores)
[1] 7
```

Comandos simples

Alguns comandos para mostrar a saída de uma variável

```
> a = 2 + 2
> a <- 2 + 2
> 4 -> b
> a
[1] 4
> print(b)
[1] 4
```

Importante mostrar como a atribuição pode ser feita nos dois sentidos.

```
> valores = c(6, 7, 4, 3, 2, 0, 0, 6)
> valores[4]
[1] 3
> length(valores)
[1] 8
> min(valores)
[1] 0
> max(valores)
[1] 7
```

Comandos simples

Alguns comandos para mostrar a saída de uma variável

```
> a = 2 + 2
> a <- 2 + 2
> 4 -> b
> a
[1] 4
> print(b)
[1] 4
```

Importante mostrar como a atribuição pode ser feita nos dois sentidos.

```
> valores = c(6, 7, 4, 3, 2, 0, 0, 6)
> valores[4]
[1] 3
> length(valores)
[1] 8
> min(valores)
[1] 0
> max(valores)
[1] 7
```

Comandos simples

Alguns comandos para mostrar a saída de uma variável

```
> a = 2 + 2
> a <- 2 + 2
> 4 -> b
> a
[1] 4
> print(b)
[1] 4
```

Importante mostrar como a atribuição pode ser feita nos dois sentidos.

```
> valores = c(6, 7, 4, 3, 2, 0, 0, 6)
> valores[4]
[1] 3
> length(valores)
[1] 8
> min(valores)
[1] 0
> max(valores)
[1] 7
```

Tipos

Vetores Matrizes e Listas

```
M <- cbind(obs,Height,Weight,BMI) # Create a matrix
> typeof(M) # Give the type of the matrix
[1] "double"
> class(M) # Give the class of an object
[1] "matrix"
> is.matrix(M) # Check if M is a matrix
[1] TRUE
> is.vector(M) # M is not a vector
[1] FALSE
> dim(M)
[1] 2 2
```


Tipos

Vetores Matrizes e Listas

```
M <- cbind(obs,Height,Weight,BMI) # Create a matrix
> typeof(M) # Give the type of the matrix
[1] "double"
> class(M) # Give the class of an object
[1] "matrix"
> is.matrix(M) # Check if M is a matrix
[1] TRUE
> is.vector(M) # M is not a vector
[1] FALSE
> dim(M)
[1] 2 2
```

Estatística Descritiva

Estatística Descritiva

```
> sum(valores)
[1] 28
> prod(valores)
[1] 0
> sort(valores)
[1] 0 0 2 3 4 6 6 7
> mean(valores)
[1] 3.5
> median(valores)
[1] 3.5
> sd(valores)
[1] 2.725541
> var(valores)
[1] 7.428571
```

Estatística Descritiva

Estatística Descritiva

```
> sum(valores)
[1] 28
> prod(valores)
[1] 0
> sort(valores)
[1] 0 0 2 3 4 6 6 7
> mean(valores)
[1] 3.5
> median(valores)
[1] 3.5
> sd(valores)
[1] 2.725541
> var(valores)
[1] 7.428571
```

Estatística Descritiva

Estatística Descritiva

```
> he <- c(10, 2, 23, 11, 14, 35, 46, 32, 13, 51, 27, 49)
> ha <- he
> var(he)
[1] 277.3561
> cov(ha, he)
[1] 277.3561
> cor(ha, he)
[1] 1
```

Estatística Descritiva

Estatística Descritiva

```
> he <- c(10, 2, 23, 11, 14, 35, 46, 32, 13, 51, 27, 49)
> ha <- he
> var(he)
[1] 277.3561
> cov(ha, he)
[1] 277.3561
> cor(ha, he)
[1] 1
```

Criando Vetores

Criando Vetores

```
> vetor <- c(1:9)
> vetor = (1:9)
> vetor = (9:1)
> seq(1, 9, by = 2)
[1] 1 3 5 7 9
> seq(1, 9, by = pi)
[1] 1.000000 4.141593 7.283185
> seq(1, 9, by = 0.5)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9
> rep(1, 10)
[1] 1 1 1 1 1 1 1 1 1 1
> rep(vetor, 5)
[1] 9 8 7 6 5 4 3 2 1 9 8 7 6 5 4 3 2 1 9 8 7 6 5 4 3 2 1 9 8 7 6 5 4
> rep(vetor, each = 5)
[1] 9 9 9 9 9 8 8 8 8 8 7 7 7 7 7 6 6 6 6 6 5 5 5 5 5 4 4 4 4 4 3 3 3
```

O índice da primeira posição do vetor é 1

Criando Vetores

Criando Vetores

```

> vetor <- c(1:9)
> vetor = (1:9)
> vetor = (9:1)
> seq(1, 9, by = 2)
[1] 1 3 5 7 9
> seq(1, 9, by = pi)
[1] 1.000000 4.141593 7.283185
> seq(1, 9, by = 0.5)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0 8.5 9
> rep(1, 10)
[1] 1 1 1 1 1 1 1 1 1 1
> rep(vetor, 5)
[1] 9 8 7 6 5 4 3 2 1 9 8 7 6 5 4 3 2 1 9 8 7 6 5 4 3 2 1 9 8 7 6 5 4
> rep(vetor, each = 5)
[1] 9 9 9 9 9 8 8 8 8 8 7 7 7 7 7 6 6 6 6 6 5 5 5 5 5 4 4 4 4 4 3 3 3

```

O índice da primeira posição do vetor é 1

Criando Matrizes

Criando 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
```

	col1	col2	col3
linha1	1	2	3
linha2	11	12	13

Criando Matrizes

Criando 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
```

	col1	col2	col3
linha1	1	2	3
linha2	11	12	13

Criando Matrizes

Criando Matrizes

```
> x <- matrix(1:9, nrow = 3)
> x
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

> coluna1 <- c(10, 20, 30)
> coluna2 <- c(5, 10, 15)
> matriz <- cbind(coluna1, coluna2)
> matriz <- rbind(coluna1, coluna2)
```

Criando Matrizes

Criando Matrizes

```
> x <- matrix(1:9, nrow = 3)
> x
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

> coluna1 <- c(10, 20, 30)
> coluna2 <- c(5, 10, 15)
> matriz <- cbind(coluna1, coluna2)
> matriz <- rbind(coluna1, coluna2)
```

Condicional

IF-ELSE

```
if (condição) {  
  bloco de comandos  
} else {  
  bloco de comandos  
}
```

SWITCH

```
switch(condição, caso1, caso2, caso3)
```

IFELSE

```
ifelse(condição, yes, no)
```

Repetição

FOR

```
for (var in seq) {  
  bloco de comandos  
}
```

WHILE

```
while (condição) {  
  bloco de comandos  
}
```

Funções

```
eleva3<-function(a) { return(a*a*a); }  
eleva3(3)
```

```
trace<-function(A) { return(ifelse(is.square(A), sum(diag(A)), NA)); }  
is.square<-function(A) {return(is.matrix(A)&&(nrow(A) == ncol(A))); }
```

Funções

```
eleva3<-function(a) { return(a*a*a); }  
eleva3(3)
```

```
trace<-function(A) { return(ifelse(is.square(A), sum(diag(A)), NA)); }  
is.square<-function(A) {return(is.matrix(A)&&(nrow(A) == ncol(A))); }
```

Funções

```
eleva3<-function(a) { return(a*a*a); }  
eleva3(3)
```

```
trace<-function(A) { return(ifelse(is.square(A), sum(diag(A)), NA)); }  
is.square<-function(A) {return(is.matrix(A)&&(nrow(A)== ncol(A))); }
```


Lendo Dados

```
dataset <- read.table ("datafile.dat", header=TRUE)
```

Para obter os vetores

```
dataset$ tamanho_lote
```

```
dataset$ homens_hora
```

```
dataset <- scan ("datafile2.dat", what=list (x=0,y=0))
```

```
dataset (cifrao)x
```

```
dataset (cifrao)y
```

```
dataset <- scan ("datafile3.dat", what=list (x="",y=""))
```

```
dados <- read.csv ("dados.csv", sep=";", dec=".", header=FALSE)
```

```
dados <- c (45, 43, 46, 48, 51, 46, 50, 47, 46, 45)
```

```
dados <- edit (data.frame ())
```

```
data.entry (dados)
```

```
x = de (dados)
```

```
x = edit (dados)
```

Lendo Dados

```
dataset <- read.table ("datafile.dat", header=TRUE)
```

Para obter os vetores

```
dataset\[tamanho_lote
```

```
dataset\[homens_hora
```

```
dataset <- scan ("datafile2.dat", what=list (x=0,y=0))
```

```
dataset(cifrao)x
```

```
dataset(cifrao)y
```

```
dataset <- scan ("datafile3.dat", what=list (x="",y=""))
```

```
dados <-read.csv("dados.csv", sep=";", dec=".",header=FALSE)
```

```
dados<-c(45,43,46,48,51,46,50,47,46,45)
```

```
dados<-edit(data.frame())
```

```
data.entry(dados)
```

```
x = de(dados)
```

```
x = edit(dados)
```

Operações entre números

Operações entre números

```
> a <- 20
> b <- 3
> a + b
[1] 23
> a - b
[1] 17
> a * b
[1] 60
> a/b
[1] 6.666667
> a^b
[1] 8000
> a%/%b
[1] 6
> a%%b
[1] 2
```

Operações entre números

Operações entre números

```
> a <- 20
> b <- 3
> a + b
[1] 23
> a - b
[1] 17
> a * b
[1] 60
> a/b
[1] 6.666667
> a^b
[1] 8000
> a%/%b
[1] 6
> a%%b
[1] 2
```

Operações entre números

Operações entre números

```
> abs(b - a)
[1] 17
> sqrt(a)
[1] 4.472136
> log(b)
[1] 1.098612
> log10(b)
[1] 0.4771213
```

Operações entre números

Operações entre números

```
> abs(b - a)
[1] 17
> sqrt(a)
[1] 4.472136
> log(b)
[1] 1.098612
> log10(b)
[1] 0.4771213
```

Operações entre números e matrizes

Operações entre números e matrizes

```
> a <- matrix(c(1, 2, 12, 13), nrow = 2, ncol = 2)
> b <- 3
> a + b
      [,1] [,2]
[1,]    4   15
[2,]    5   16

> a - b
      [,1] [,2]
[1,]   -2    9
[2,]   -1   10

> a * b
      [,1] [,2]
[1,]    3   36
[2,]    6   39
\normalsize
```

Operações entre números e matrizes

Operações entre números e matrizes

```
> a <- matrix(c(1, 2, 12, 13), nrow = 2, ncol = 2)
> b <- 3
> a + b
      [,1] [,2]
[1,]    4   15
[2,]    5   16

> a - b
      [,1] [,2]
[1,]   -2    9
[2,]   -1   10

> a * b
      [,1] [,2]
[1,]    3   36
[2,]    6   39
\normalsize
```


Operações entre números e matrizes

Operações entre números e matrizes

```
> a/b
```

```
      [,1]      [,2]  
[1,] 0.3333333 4.000000  
[2,] 0.6666667 4.333333
```

```
> a^b
```

```
      [,1] [,2]  
[1,]     1 1728  
[2,]     8 2197
```

```
> a%/%b
```

```
      [,1] [,2]  
[1,]     0   4  
[2,]     0   4
```

```
> a%%b
```

```
      [,1] [,2]  
[1,]     1   0  
[2,]     2   1
```

Operações entre números e matrizes

Operações entre números e matrizes

```
> a/b
```

```
      [,1]      [,2]
[1,] 0.3333333 4.000000
[2,] 0.6666667 4.333333
```

```
> a^b
```

```
      [,1] [,2]
[1,]    1 1728
[2,]    8 2197
```

```
> a%/%b
```

```
      [,1] [,2]
[1,]    0    4
[2,]    0    4
```

```
> a%%b
```

```
      [,1] [,2]
[1,]    1    0
[2,]    2    1
```

Operações entre Matrizes

Operações entre Matrizes

```
> a <- matrix(1:4, nrow = 2)
> b <- matrix(c(1, 2, 12, 13), nrow = 2, ncol = 2)
> a + b
```

```
      [,1] [,2]
[1,]    2  15
[2,]    4  17
```

```
> a - b
```

```
      [,1] [,2]
[1,]    0  -9
[2,]    0  -9
```

```
> a %*% b
```

```
      [,1] [,2]
[1,]    7  51
[2,]   10  76
```

Operações entre Matrizes

Operações entre Matrizes

```
> a <- matrix(1:4, nrow = 2)
> b <- matrix(c(1, 2, 12, 13), nrow = 2, ncol = 2)
> a + b
```

```
      [,1] [,2]
[1,]    2  15
[2,]    4  17
```

```
> a - b
```

```
      [,1] [,2]
[1,]    0  -9
[2,]    0  -9
```

```
> a %*% b
```

```
      [,1] [,2]
[1,]    7  51
[2,]   10  76
```

Operadores Lógicos

Operadores Lógicos

```
> x <- 10
> y <- 10
> z <- 11
> x == y
[1] TRUE
> x != z
[1] TRUE
> x > z
[1] FALSE
> z < y
[1] FALSE
> x >= z
[1] FALSE
> z <= y
[1] FALSE
```

Operadores Lógicos

Operadores Lógicos

```
> x <- 10
> y <- 10
> z <- 11
> x == y
[1] TRUE
> x != z
[1] TRUE
> x > z
[1] FALSE
> z < y
[1] FALSE
> x >= z
[1] FALSE
> z <= y
[1] FALSE
```

Operadores Lógicos

Operadores Lógicos

```
> x == y & x == z  
[1] FALSE  
> x >= y | z < y  
[1] TRUE  
> x >= z | !z <= y  
[1] TRUE
```

Operadores Lógicos

Operadores Lógicos

```
> x == y & x == z
```

```
[1] FALSE
```

```
> x >= y | z < y
```

```
[1] TRUE
```

```
> x >= z | !z <= y
```

```
[1] TRUE
```


Utilização dos operadores Lógicos

Utilização dos operadores Lógicos

```
> peso = c(60, 72, 57, 90, 95, 72)
> altura = c(1.75, 1.8, 1.65, 1.9, 1.74, 1.91)
> nome = c("jó", "josé", "joão", "zé", "noé", "mané")
> peso > 70
[1] FALSE TRUE FALSE TRUE TRUE TRUE
> peso[peso > 70]
[1] 72 90 95 72
> nome[peso > 70]
[1] "josé" "zé" "noé" "mané"
> nome[(peso > 70) & (peso < 80)]
[1] "josé" "mané"
> IMC = peso/altura^2
> nome[IMC > 25]
[1] "noé"
```

Utilização dos operadores Lógicos

Utilização dos operadores Lógicos

```
> peso = c(60, 72, 57, 90, 95, 72)
> altura = c(1.75, 1.8, 1.65, 1.9, 1.74, 1.91)
> nome = c("jó", "josé", "joão", "zé", "noé", "mané")
> peso > 70
[1] FALSE TRUE FALSE TRUE TRUE TRUE
> peso[peso > 70]
[1] 72 90 95 72
> nome[peso > 70]
[1] "josé" "zé" "noé" "mané"
> nome[(peso > 70) & (peso < 80)]
[1] "josé" "mané"
> IMC = peso/altura^2
> nome[IMC > 25]
[1] "noé"
```

Outros Valores (Missing Values)

Outros Valores (Missing Values)

```
> 1/0
[1] Inf
> 0/0
[1] NaN
> -1/0
[1] -Inf
> is.infinite(1/0)
[1] TRUE
> is.nan(0/0)
[1] TRUE
> is.finite(-1/0)
[1] FALSE
```

Outros Valores (Missing Values)

Outros Valores (Missing Values)

```
> 1/0
[1] Inf
> 0/0
[1] NaN
> -1/0
[1] -Inf
> is.infinite(1/0)
[1] TRUE
> is.nan(0/0)
[1] TRUE
> is.finite(-1/0)
[1] FALSE
```

Geração de Números

```
> set.seed(19)
> runif(10, 0, 1)
[1] 0.11713058 0.48402973 0.65120703 0.06839668 0.36529780 0.22390370
[9] 0.83662033 0.72605208
> rnorm(10, 5, 1)
[1] 4.763354 5.809740 4.255220 4.740213 4.816916 5.518630 5.883465 5.
> rt(10, 2)
[1] -0.6191805498 0.8282425374 -1.5533998143 0.0009336122 -1.024823
[7] -1.8131882456 2.9385662011 0.1922851229 1.1864419888
> rchisq(10, 2)
[1] 3.2891064 11.4887735 1.2823856 5.5375688 0.3132790 2.7323381
[9] 2.5129717 3.7138720
```

Geração de Números

```
> set.seed(19)
> runif(10, 0, 1)
[1] 0.11713058 0.48402973 0.65120703 0.06839668 0.36529780 0.22390370
[9] 0.83662033 0.72605208
> rnorm(10, 5, 1)
[1] 4.763354 5.809740 4.255220 4.740213 4.816916 5.518630 5.883465 5.
> rt(10, 2)
[1] -0.6191805498 0.8282425374 -1.5533998143 0.0009336122 -1.024825
[7] -1.8131882456 2.9385662011 0.1922851229 1.1864419888
> rchisq(10, 2)
[1] 3.2891064 11.4887735 1.2823856 5.5375688 0.3132790 2.7323381
[9] 2.5129717 3.7138720
```

Amostragem

```
> set.seed(19)
> sample(1:40, 5)
[1] 5 19 25 3 14
> sample(c("H", "T"), 10, replace = T)
[1] "H" "H" "T" "T" "T" "H" "H" "T" "T" "H"
> 1/prod(40:36)
[1] 1.266449e-08
> choose(4, 5)
[1] 0
```

Amostragem

```
> set.seed(19)
> sample(1:40, 5)
[1] 5 19 25 3 14
> sample(c("H", "T"), 10, replace = T)
[1] "H" "H" "T" "T" "T" "H" "H" "T" "T" "H"
> 1/prod(40:36)
[1] 1.266449e-08
> choose(4, 5)
[1] 0
```

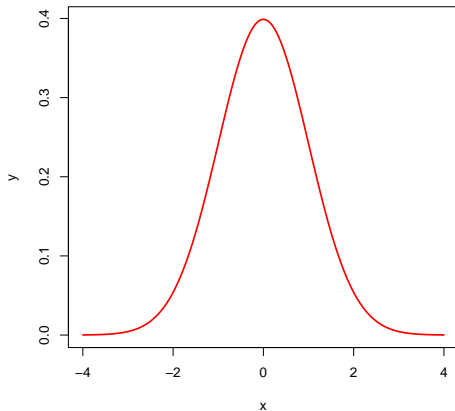

Calculando Densidades de Probabilidade

```
> x = seq(-4, 4, length = 200)
> y = dnorm(x, mean = 0, sd = 1)
> plot(x, y, type = "l", lwd = 2, col = "red")
```

Calculando Densidades de Probabilidade

```
> x = seq(-4, 4, length = 200)
> y = dnorm(x, mean = 0, sd = 1)
> plot(x, y, type = "l", lwd = 2, col = "red")
```

Densidades de Probabilidade



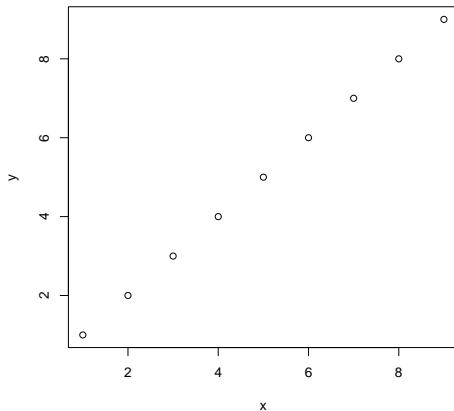
Plot Normal

```
> 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")
```

Plot Normal

```
> 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")
```

Plot Normal



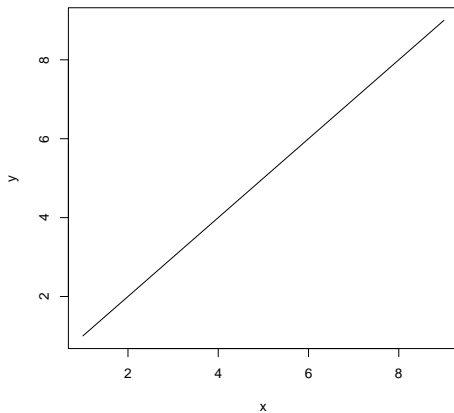
Plot Normal 2

```
> x = c(1:9)
> y = c(1:9)
> plot(x, y, type = "l")
```

Plot Normal 2

```
> x = c(1:9)
> y = c(1:9)
> plot(x, y, type = "l")
```


Plot Normal 2



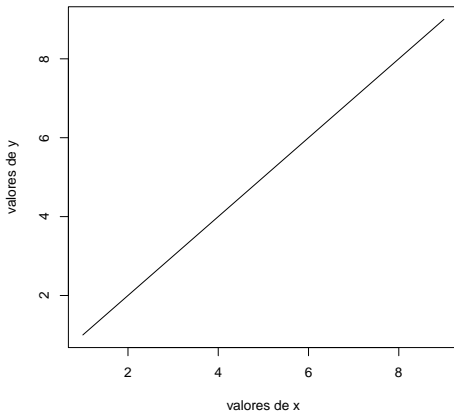
Plot Normal 3

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

Plot Normal 3

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

Plot Normal 3



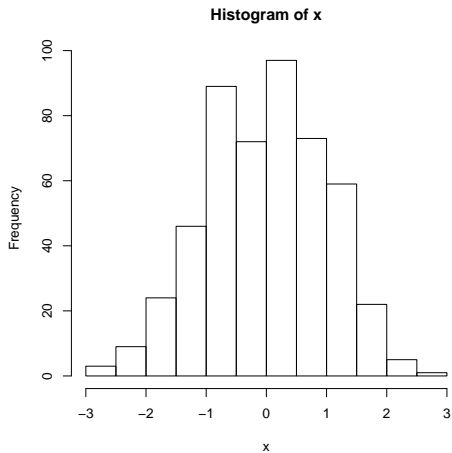
Plot Histograma

```
> x = rnorm(500, 0, 1)
> mean(x)
[1] 0.05350359
> median(x)
[1] 0.04056359
> sd(x)
[1] 0.9759333
> var(x)
[1] 0.9524458
> hist(x)
```

Plot Histograma

```
> x = rnorm(500, 0, 1)
> mean(x)
[1] 0.05350359
> median(x)
[1] 0.04056359
> sd(x)
[1] 0.9759333
> var(x)
[1] 0.9524458
> hist(x)
```

Plot Histograma



Frequência

Uma pesquisa pergunta se as pessoas fumam ou não. As respostas foram:

```
> x = c("Yes", "No", "No", "Yes", "Yes", "No", "No", "Yes")
> table(x)
x
  No Yes
   4   4
> factor(x)
[1] Yes No  No  Yes Yes No  No  Yes
Levels: No Yes
```


Frequência

Uma pesquisa pergunta se as pessoas fumam ou não. As respostas foram:

```
> x = c("Yes", "No", "No", "Yes", "Yes", "No", "No", "Yes")
> table(x)
x
  No Yes
   4   4
> factor(x)
[1] Yes No  No  Yes Yes No  No  Yes
Levels: No Yes
```

Frequência

Uma pesquisa pergunta se as pessoas fumam ou não. As respostas foram:

```
> x = c("Yes", "No", "No", "Yes", "Yes", "No", "No", "Yes")
> table(x)
x
  No Yes
   4   4
> factor(x)
[1] Yes No  No  Yes Yes No  No  Yes
Levels: No Yes
```

Plot Barras

Uma pesquisa pergunta qual a cerveja preferida. 1 - Brahma; 2 - Kaiser; 3 Bohemia; 4 - schin As respostas foram: 3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1

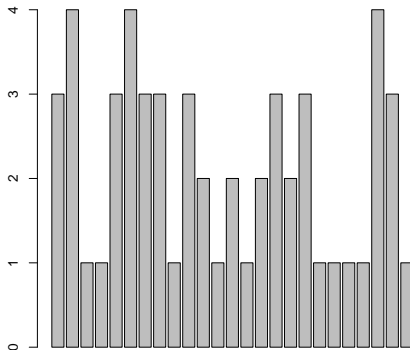
```
> cerveja = c(3, 4, 1, 1, 3, 4, 3, 3, 1, 3, 2, 1, 2, 1, 2, 3, 2, 3, 1,
+            1, 4, 3, 1)
> barplot(cerveja)
```

Plot Barras

Uma pesquisa pergunta qual a cerveja preferida. 1 - Brahma; 2 - Kaiser; 3 Bohemia; 4 - schin As respostas foram: 3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1

```
> cerveja = c(3, 4, 1, 1, 3, 4, 3, 3, 1, 3, 2, 1, 2, 1, 2, 3, 2, 3, 1,
+            1, 4, 3, 1)
> barplot(cerveja)
```

Plot Barras



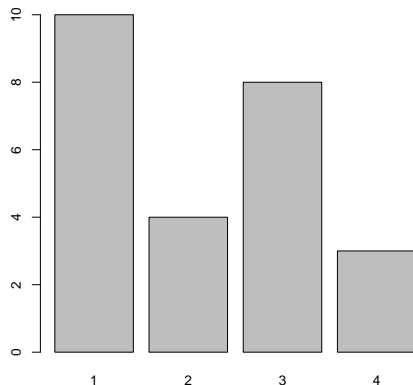
Plot Barras

```
> barplot(table(cerveja))
```

Plot Barras

```
> barplot(table(cerveja))
```

Plot Barras



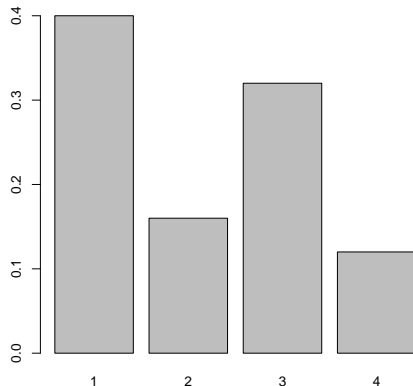
Plot Barras

```
> barplot(table(cerveja)/length(cerveja))
```

Plot Barras

```
> barplot(table(cerveja)/length(cerveja))
```

Plot Barras



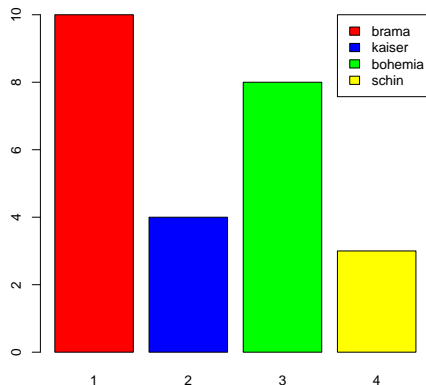
Plot Barras

```
> barplot(table(cerveja), col = c("red", "blue", "green", "yellow"))  
> legend("topright", c("brama", "kaiser", "bohemia", "schin"), fill =  
+       "blue", "green", "yellow"))
```

Plot Barras

```
> barplot(table(cerveja), col = c("red", "blue", "green", "yellow"))  
> legend("topright", c("brama", "kaiser", "bohemia", "schin"), fill =  
+       "blue", "green", "yellow"))
```

Plot Barras



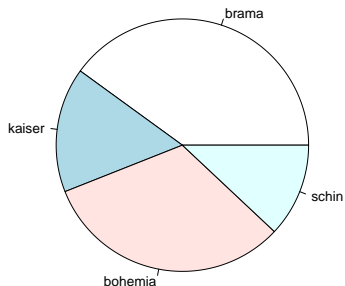
Plot Pizza

```
> nomes = table(cerveja)
> names(nomes) = c("brama", "kaiser", "bohemia", "schin")
> pie(nomes)
```

Plot Pizza

```
> nomes = table(cerveja)
> names(nomes) = c("brama", "kaiser", "bohemia", "schin")
> pie(nomes)
```


Plot Pizza



Dúvidas



Dúvidas?
Arthur Alvim
afma@cin.ufpe.br

Obrigado pela Atenção!



As "Aulas Básicas de R" de Arthur Alvim foram licenciada com uma Licença:
Creative Commons - Atribuição - Uso Não-Comercial - Partilha nos Mesmos Termos 3.0 Não Adaptada.