

Análise de Dados Ambientais com R

Jônatan Tatsch

2018-04-17

Contents

Apresentação	5
1 Introdução	7
1.1 Análise de dados meteorológicos	7
1.2 Ciência de dados	7
1.3 Etapas para abordagem de um problema	7
1.4 Programação computacional	8
1.5 R	8
1.6 Por que o R?	8
1.7 Pacotes da comunidade do R	8
1.8 Por que um meteorologista usaria o R?	9
1.9 R não é perfeito!	9
1.10 Para saber mais sobre o R	9
2 Instalação do R e RStudio	11
2.1 Instalando o R	11
2.2 Diretório para instalação de pacotes	15
2.3 Rstudio no Ubuntu	15
3 Interface do Usuário	21
3.1 R no modo interativo	21
3.2 Expressões em sequência	22
3.3 Navegação entre as expressões já avaliadas	23
3.4 Comentários	23
3.5 Auto preenchimento de funções	24
3.6 R no modo não interativo (<i>Batch</i>)	24
3.7 Rodando script de dentro do R	25
4 Operações básicas	27
4.1 Formação do solo	27
4.2 Propriedades do solo	27
4.3 Textura e Estrutura	27
4.4 Movimento da água no solo	27
4.5 Infiltração	27
5 Tipos de dados	29
5.1 Evapotranspiração de referência	29
5.2 Evapotranspiração da cultura	29
5.3 Evapotranspiração real	29

Apresentação

Este material é uma composição das notas de aula da disciplina **Análise de Dados Ambientais com R** do curso de Graduação em METEOROLOGIA oferecido no Departamento de Física da Universidade Federal de Santa Maria (UFSM).

O livro é designado para quem não tem experiência em programação, ou qualquer um com interesse em aprender o R para manipular dados ambientais. O objetivo é prover uma material para ensinar os conceitos básicos de programação necessários para o processamento, a visualização e a análise de dados ambientais com o sistema computacional R. Estes procedimentos são potencializados com o uso do software RStudio, uma interface de desenvolvimento integrado (IDE) para o R.

Neste livro o leitor aprenderá a sintaxe básica da linguagem R (R Core Team, 2018), a importação e exportação de dados, a criação de gráficos, funções, a padronização e organização de conjunto de dados; e finalmente, a confecção de relatórios dinâmicos e reproduzíveis.

O material do livro inclui o uso de dados ambientais de diferentes áreas (meteorologia, climatologia, hidrologia, sensoriamento remoto) em exemplos práticos e em exercícios, para estimular a prática da programação.

O texto é intercalado com trechos de códigos que podem ser reproduzidos e os resultados visualizados no computador do leitor.

Após a introdução ao R apresenta-se as capacidades específicas do R para manipulação de dados. Baseado na experiência do autor são empregados os pacotes mais adequados para cada finalidade, como `dplyr` e `tidyr` para o processamento de dados e o `ggplot2` para visualização de dados.

A intenção do livro é que após a leitura, o leitor tenha o conhecimento suficiente para desenvolver códigos que automatizem tarefas repetitivas, assim reduzindo o tempo na etapa de preparação de dados. Esta programação mais efetiva permitirá focar mais na análise de dados e na comunicação dos resultados, seja ela na forma de documentos acadêmicos, ou relatórios técnicos em empresas públicas e privadas.

O texto está em formato html para tirar o melhor proveito de recursos de multimídia, da capacidade de busca de texto e links para websites.

O texto é organizado em 5 capítulos:

- 1 Introdução
- 2 Instalação do R e Rstudio
- 3 Interface do Usuário
- 4 Operações Básicas
- 5 Tipos de dados

Chapter 1

Introdução

Breve intro.

1.1 Análise de dados meteorológicos

Processo pelo qual adquire-se conhecimento, compreensão e percepção dos fenômenos meteorológicos a partir de observações (dados) qualitativas e quantitativas.

1.2 Ciência de dados

1.3 Etapas para abordagem de um problema

1. **Questão científica/problema**
2. **Obtenção de dados:** coleta/medida do(as) estado/condições da atmosfera
 - Instrumentos e sensores
3. **Processamento de dados:** *download* —> limpeza —> formatação —> transformação —> controle de qualidade
 - ferramenta/software
 - conhecimento em programação
4. **Análise de dados**
 - ferramenta/software
 - conhecimento em programação
5. **Solução para o problema**
 - Proposta de um modelo
 - estatístico, empírico, ou fisicamente baseado
 - conhecimento em programação
6. **Apresentação/divulgação/publicação**

1.4 Programação computacional

1.5 R

- R é o termo usado para se referir a linguagem de programação e ao software que interpreta os scripts escritos usando esta linguagem.
- Comunidade fantástica
- Contribuidores (R-core Team)
- milhares de pessoas usam o R diariamente e ajudam outras pessoas
- **Software Livre** (GPL), Código aberto e multiplataforma
- Ambiente para Análise de dados interativa

1.6 Por que o R?

- R não é uma GUI (Interface gráfica do usuário) e isso é bom
 - há uma natural resistência e dificuldade ao uso de códigos e scripts
 - scripts favorecem a **automatização** e **reprodutibilidade**
 - força você a ter um conhecimento mais aprofundado do que está fazendo
- Reprodutibilidade
 - qualquer pessoa (inclusive você mesmo no futuro) pode obter os mesmos resultados do mesmo conjunto de dados
 - R é integrado com outras ferramentas de que permitem atualizar seus resultados, figuras e análises automaticamente
- Relatório dinâmicos e interativos
- Acesso ao estado da arte da ciência de dados (*Big Data*, *Data Mining*, *Machine Learning*)
- é um software livre, de código fonte aberto e funciona em diversos sistemas operacionais (Linux, Windows e MacOS).
- Interface com Fortran, C, C++, Python
- Visualização de dados
- R produz gráficos de alta qualidade
- R trabalha com dados de todas formas e tamanhos
- Extensões para Manipulação de dados

1.7 Pacotes da comunidade do R

Evolução do nº de pacotes disponíveis no CRAN

1.8 Por que um meteorologista usaria o R?

A meteorologia é 4D:

```
meteorologia <- function(x, y, z, t){  
  ...muita coisa para caber em um slide...  
}
```

Logo, requer ferramentas específicas para:

- manipulação de dados espaciais
- análise de séries temporais
- importação e ferramentas de SIG
- leitura de dados em formatos específicos (netcdf, binários, grib2, ...)

1.9 R não é perfeito!

- Muitos códigos em R são escritos para resolver um problema;
 - foco nos resultados e não no processo
 - usuários não são programadores
 - códigos deselegantes, lentos e difíceis de entender
- Como o nosso idioma, há muitas exceções para serem lembradas
- R não é muito rápido e códigos mal escritos serão lentos
- São apenas ~20 anos de evolução
- Há muito o que melhorar

1.10 Para saber mais sobre o R

Documentação oficial - Manuais do R traduzidos

Lista de Livros relacionados ao R

- Livros gratuitos (em inglês)

Fóruns:

- lista Brasileira de discussão do programa R: **R-br**
- stackoverflow

Chapter 2

Instalação do R e RStudio

A interação do usuário com o R é por meio da linha de comando. Essa interação pode ser facilitada com o uso do RStudio.

A seguir descreve-se como instalar o *R* no Windows e no Linux Ubuntu. A forma de instalação do R no Linux tenta ser mais didática do que prática. Alguns comandos linux básicos serão utilizados, mas mesmo quem não é usuário linux será capaz de entendê-los.

2.1 Instalando o R

O *R* pode ser instalado a partir dos binários pré-compilados ou do código fonte. Aqui, descreve-se a instalação do *R* a partir dos binários

2.1.1 Windows

A forma de instalar o R no Windows é baixar o binário executável da **Rede Abrangente de Arquivos do R** (CRAN). Depois clicar em *Download R for Windows* e *install R for the first time*. Quando este tutorial foi escrito a última versão foi a R 3.4.4.

A instalação do R para Windows a partir do executável acima incluirá na instalação uma GUI chamada `RGui.exe`, mostrada abaixo.

2.1.2 Linux

2.1.2.1 Ubuntu

Há várias formas de instalar o R no Ubuntu, mas geralmente a versão compilada no repositório *default* do Ubuntu não é a última. Se isso for problema para você então basta executar:

```
sudo apt-get install r-base
```

2.1.2.2 R sempre atualizado

Se você prefere trabalhar com a última versão estável do R, precisamos configurar o Linux Ubuntu para atualizar automaticamente o *R*. Também uma boa prática definir um diretório para armazenamento dos pacotes utilizados.

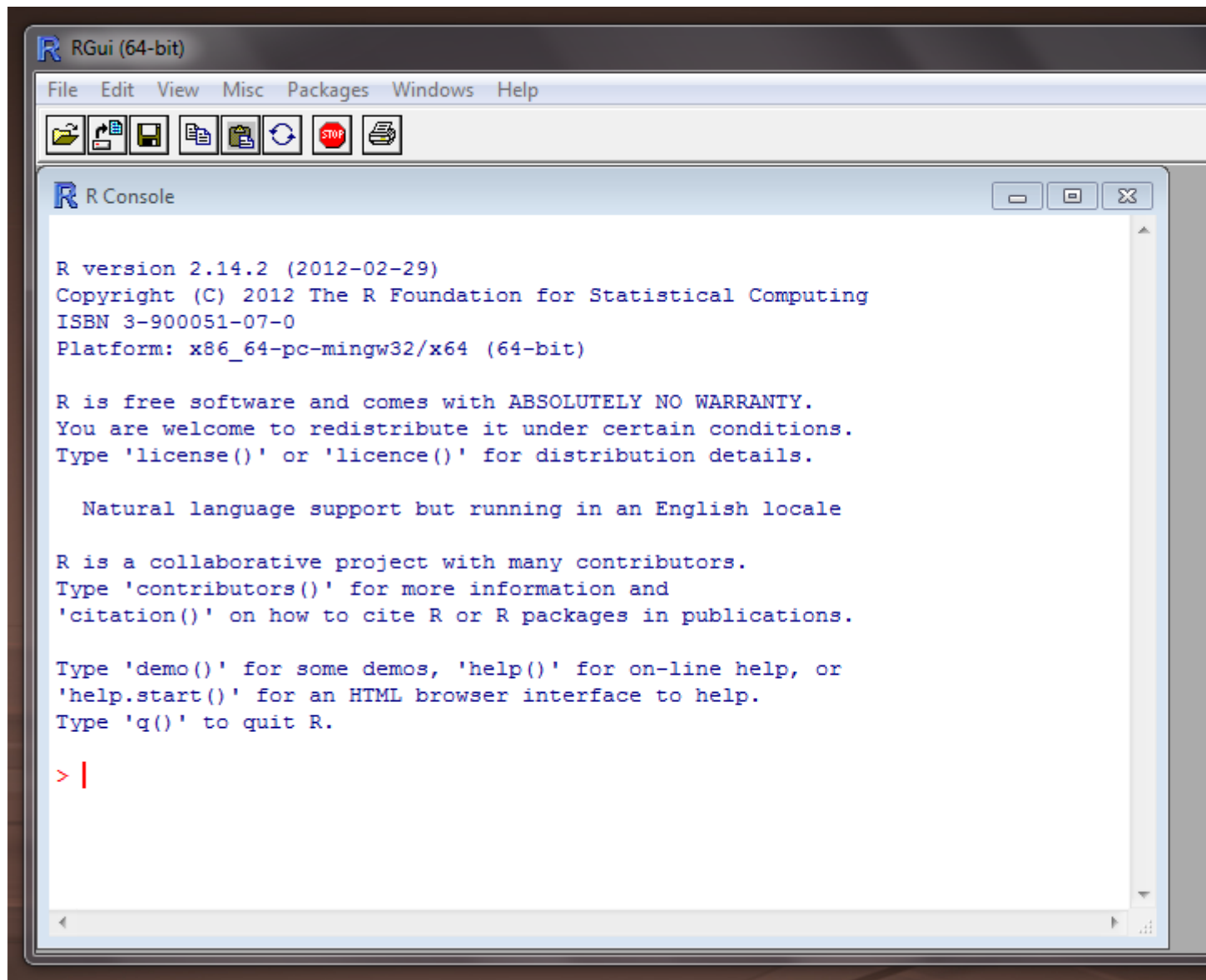


Figure 2.1: Interface gráfica do usuário no R para Windows.

O procedimento de instalação requer senha de superusuário do sistema ou de privilégios sudo. Caso não tenha, consulte o administrador do sistema.

Ao utilizar distribuições Linux Ubuntu é importante optar por versões estáveis¹. As versões de Suporte de longo prazo (LTS) mais recentes são:

- 14.04 (abril de 2014, *codename* `trusty`)
- 16.04 (abril de 2016, *codename* `xenial`)

O R é distribuído na CRAN. Geralmente há duas atualizações ao ano. A versão mais atual é a R version 3.4.4 (2018-03-15). Para que ele seja atualizado automaticamente no Ubuntu precisamos adicionar o repositório do R mais próximo da nossa região à lista de repositórios do Linux. No nosso caso, o repositório mais próximo é o da UFPR (<http://cran-r.c3sl.ufpr.br/>).

2.1.2.2.1 Incluindo repositório do R na Lista de repositórios do Ubuntu

A lista de repositórios do sistema é armazenada no arquivo `/etc/apt/sources.list`. Vamos visualizar o conteúdo desse arquivo. Em um terminal linux (use o atalho `Ctrl+Alt+t`), digite o seguinte comando:

```
$ cat /etc/apt/sources.list | head -15

## # deb cdrom:[Ubuntu 14.04.2 LTS _Trusty Tahr_ - Release amd64 (20150218.1)]/ trusty main restricted
##
## # See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
## # newer versions of the distribution.
## # deb http://br.archive.ubuntu.com/ubuntu/ trusty main restricted
## # deb-src http://br.archive.ubuntu.com/ubuntu/ trusty main restricted
## deb http://ubuntu.c3sl.ufpr.br/ubuntu/ trusty main restricted
## deb-src http://ubuntu.c3sl.ufpr.br/ubuntu/ trusty main restricted
## ## Major bug fix updates produced after the final release of the
## ## distribution.
## deb http://br.archive.ubuntu.com/ubuntu/ trusty-updates main restricted
## deb-src http://br.archive.ubuntu.com/ubuntu/ trusty-updates main restricted
##
## ## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## ## team. Also, please note that software in universe WILL NOT receive any
```

Para descobrir o nome da versão do sistema operacional, digite na terminal linux o seguinte comando²:

```
$ lsb_release --codename | cut -f2

trusty
```

Precisamos incluir no arquivo `sources.list` o repositório da UFPR. Assim o gerenciador de pacotes `apt`³ fará a atualização do R quando uma nova versão estiver disponível. Ou seja, você estará utilizando sempre versão mais atual do R.

O endereço do repositório da UFPR será inserido na última linha do arquivo `sources.list` usando alguns comandos linux. Essa tarefa requer privilégios de superusuário. Vamos trocar do seu usuário para o superusuário.

```
$ sudo su
```

Vamos definir no terminal uma variável com o endereço do repositório e o nome de versão do Ubuntu.

```
# repos="deb http://cran-r.c3sl.ufpr.br/bin/linux/ubuntu `lsb_release --codename | cut -f2`/"
```

Note que a variável `repos` é uma sequência de caracteres com as seguintes informações:

¹Clique aqui para saber mais sobre as versões do Ubuntu.

²Se o comando `lsb_release` não funcionar você precisa instalar o pacote `lsb-release` no sistema. Para isso digite no terminal Linux `$ sudo apt-get install lsb-release`.

³o gerenciador de pacotes `apt` é usado para instalação, atualização e remoção de pacotes em distribuições Debian GNU/Linux.

```
deb `linkRepositorioSelecioneado`/bin/linux/ubuntu `versaoUbuntu`/
```

O valor da variável `repos` é mostrado pelo comando: `echo $repos`. Certifique-se de que a última palavra corresponde ao nome da sua versão Ubuntu.

Para acrescentar essa informação no final do arquivo `sources.list` digite no terminal linux:

```
# echo $repos >> /etc/apt/sources.list
```

Feito isso, você pode retornar a sessão de usuário comum, usando o comando abaixo:

```
# exit
```

2.1.2.2.2 APT protegido

Os arquivos binários do *R* para Ubuntu na CRAN são assinados com uma chave pública⁴ Para adicionar essa chave ao seu sistema digite os seguintes comandos:

```
$ gpg --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys E084DAB9
```

e então use essa informação como entrada no `apt-key` com

```
$ gpg -a --export E084DAB9 | sudo apt-key add -
```

Se aparecer a mensagem de que a chave pública foi importada, então não há necessidade de executar os comandos abaixo. Mas caso seja impresso alguma mensagem de erro, outra alternativa pode ser usada para obter a chave, via os comandos:

```
$ gpg --keyserver keyserver.ubuntu.com --recv-key E084DAB9
```

```
$ gpg -a --export E084DAB9 | sudo apt-key add -
```

2.1.2.2.3 Atualização da lista de repositórios do Ubuntu e instalação do *R*

Após fazer as configurações da lista de repositórios e adicionar a chave é necessário fazer a atualização dessa lista (requer poderes de super usuário):

```
$ sudo apt-get update
```

Agora, pode instalar o binário do *R*:

```
$ sudo apt-get install r-base
```

2.1.2.2.4 Testando o *R*

Para iniciar o *R* no Ubuntu, digite *R* no cursor do terminal:

```
$ R
```

A partir desse momento já começamos uma sessão no *R*. Vamos gerar uma sequência numérica de 1 a 10 e plotá-la.

```
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10
> plot(1:10)
```

Vamos sair do *R* sem salvar os dados da sessão.

```
> q(save = "no")
```

⁴Chave pública de autenticação é um meio alternativo de se logar em um servidor ao invés de digitar uma senha. É uma forma mais segura e flexível, mas mais difícil de ser configurada. Esse meio alternativo de fazer login é importante se o computador está visível na internet. Para saber mais veja aqui.

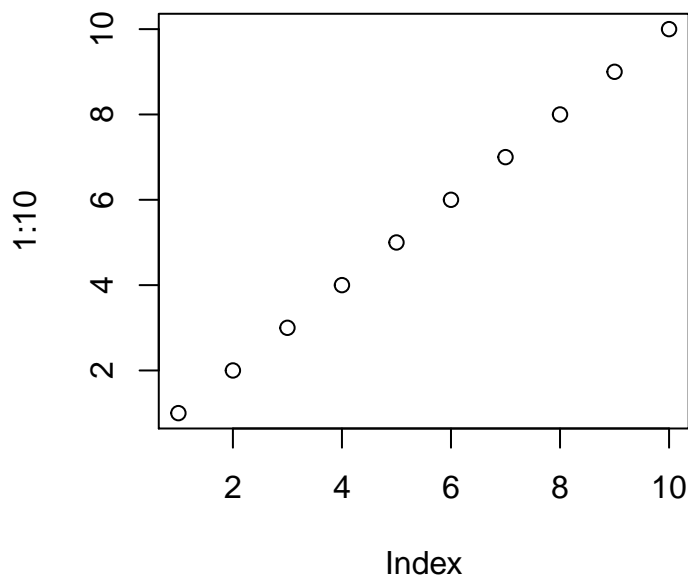


Figure 2.2: Gráfico da sequência de 10 números.

2.2 Diretório para instalação de pacotes

Para termos melhor controle sobre os pacotes do *R* instalados no sistema é recomendado criar um diretório (`/home/usuario/.R/libs`) no seu *home*. O diretório do *home* pode ser obtido com o comando `echo $HOME`.

```
$ mkdir -p `echo $HOME`/.R/libs/
```

Para informar ao *R* onde procurar os pacotes que instalamos criamos um arquivo chamado `.Renviron` no diretório `$HOME` contendo a linha `R_LIBS=/home/usuario/.R/libs/`, o que pode ser feito com o comando:

```
$ R_LIBS=`echo $HOME/.R/libs/`  
$ echo $R_LIBS >> `echo $HOME/.Renviron`
```

Esse caminho fica então visível ao *R*, o que pode ser verificado executando a função `.libPaths()` no console do *R*.

```
$ R
```

No console do *R*:

```
> .libPaths()  
[1] "/home/hidrometeorologista/.R/libs" "/usr/local/lib/R/site-library"  
[3] "/usr/lib/R/site-library"          "/usr/lib/R/library"
```

2.3 Rstudio no Ubuntu

O RStudio é um ambiente integrado de desenvolvimento (IDE) construído especificamente para o *R*. Assim como o *R*, o RStudio pode ser baixado gratuitamente e é multiplataforma.

Para instalação da versão do RStudio para *Desktop*, você precisa saber se seu sistema operacional é 64 ou 32-bit e a versão do Linux Ubuntu. Essas informações podem ser obtidas, respectivamente, pelos comandos:

```
$ arch  
x86_64  
$ lsb_release --release | cut -f2
```

14.04

Se retornar **x86_64** sua máquina é 64-bit.


Com essa informação e versão do sistema operacional, siga os seguintes passos:

1. acesse RStudio
2. clique em *Download RStudio*
3. Procure a opção *RStudio Desktop (FREE)* e clique *download*

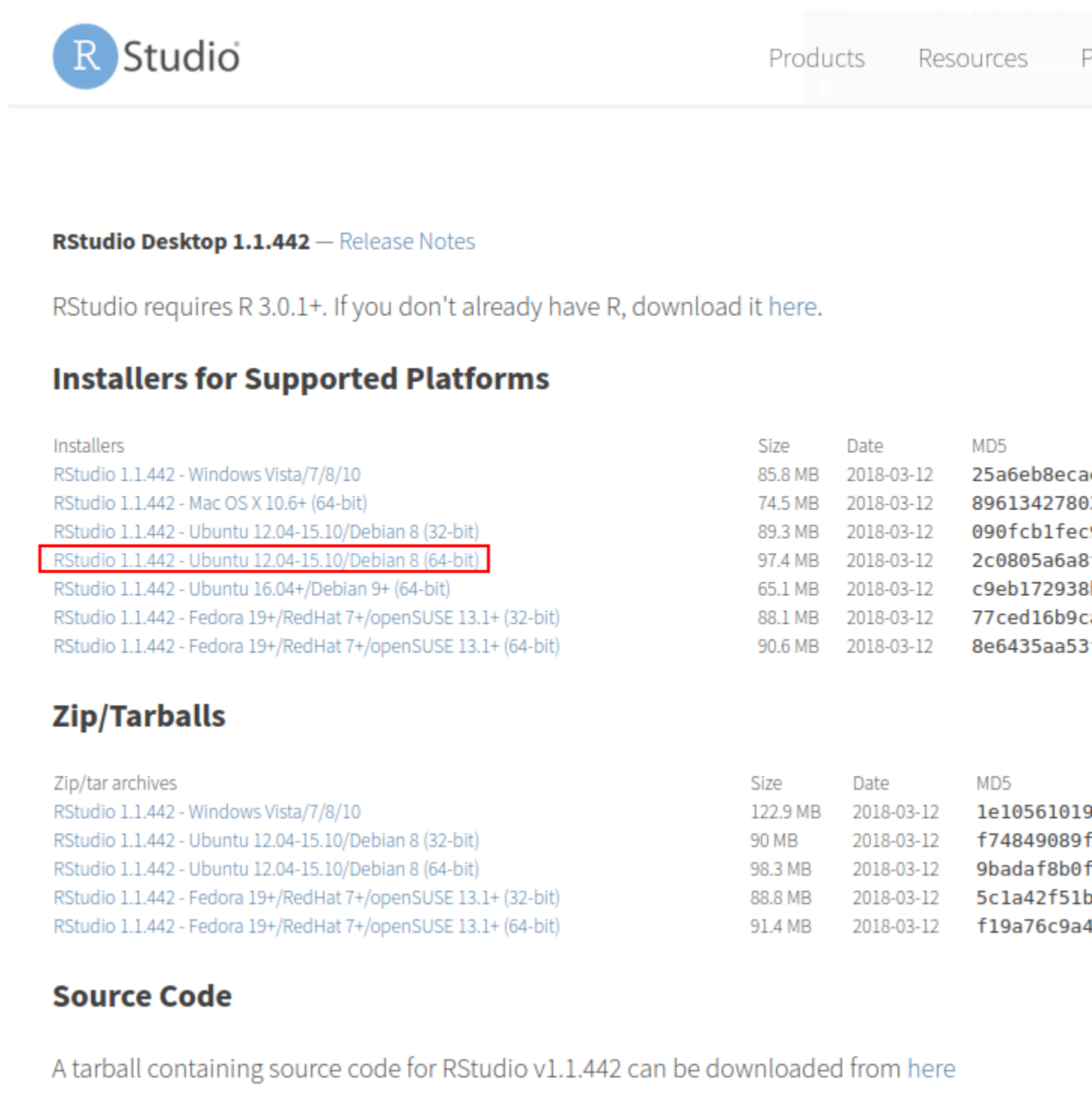
[Products](#)[Resources](#)[Pricing](#)

Choose Your Version of RStudio

RStudio is a set of integrated tools designed to help you be more productive with R. It includes a code editor with syntax-highlighting, a console for direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace. [Learn More about RStudio features.](#)

	RStudio Desktop Open Source License	RStudio Desktop Commercial License	RStudio Server Open Source License	RStudio Server Commercial License
	<u>FREE</u>	\$995 per year	FREE	\$9,995 per year
	 DOWNLOAD	BUY	DOWNLOAD	DOWNLOAD
	Learn More	Learn More	Learn More	Learn More
Integrated Tools for R	●	●	●	●
Priority Support		●		●
Access via Web Browser			●	
Enterprise Security				●
Project Sharing				●

5. Selecione sua plataforma



RStudio Desktop 1.1.442 — Release Notes

RStudio requires R 3.0.1+. If you don't already have R, download it [here](#).

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.1.442 - Windows Vista/7/8/10	85.8 MB	2018-03-12	25a6eb8ecac
RStudio 1.1.442 - Mac OS X 10.6+ (64-bit)	74.5 MB	2018-03-12	8961342780
RStudio 1.1.442 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	89.3 MB	2018-03-12	090fcb1fec
RStudio 1.1.442 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	97.4 MB	2018-03-12	2c0805a6a8
RStudio 1.1.442 - Ubuntu 16.04+/Debian 9+ (64-bit)	65.1 MB	2018-03-12	c9eb172938
RStudio 1.1.442 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.1 MB	2018-03-12	77ced16b9c
RStudio 1.1.442 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	90.6 MB	2018-03-12	8e6435aa53

Zip/Tarballs

Zip/tar archives	Size	Date	MD5
RStudio 1.1.442 - Windows Vista/7/8/10	122.9 MB	2018-03-12	1e10561019
RStudio 1.1.442 - Ubuntu 12.04-15.10/Debian 8 (32-bit)	90 MB	2018-03-12	f74849089f
RStudio 1.1.442 - Ubuntu 12.04-15.10/Debian 8 (64-bit)	98.3 MB	2018-03-12	9badaf8b0f
RStudio 1.1.442 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)	88.8 MB	2018-03-12	5c1a42f51b
RStudio 1.1.442 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)	91.4 MB	2018-03-12	f19a76c9a4

Source Code

A tarball containing source code for RStudio v1.1.442 can be downloaded from [here](#)

clique sobre o link da sua plataforma, p.ex.: *RStudio x.xx.xxx - Ubuntu 12.04-15.10/Debian 8 (64-bit)*

6. Dependendo da sua versão Ubuntu, ao clicar sobre o sobre o arquivo baixado com o botão direito, há a opção de abrir com *Ubuntu Software Center* e então clicar em **instalar**. Se na versão de seu Desktop não há esta opção ao clicar com botão direito sobre o arquivo, instale via **terminal**⁵ com os seguintes

⁵digite 'Ctrl+Alt+t' para abrir um terminal no Linux Ubuntu

comandos:

```
$ cd /local/do/arquivo/baixado  
$ sudo dpkg -i arquivoBaixado.deb  
$ sudo apt-get install -f
```

Abra o RStudio digitando no terminal:

```
$ rstudio &
```

Agora você está pronto para começar a programar em *R* aproveitando as facilidades que o RStudio oferece.

Chapter 3

Interface do Usuário

3.1 R no modo interativo

3.1.1 Convenção

No texto abaixo as expressões a serem avaliadas no R não serão mostradas com o prompt do R (>) para tornar mais fácil de copiá-las e colá-las na linha de comando do R. O resultado da avaliação das expressões será mostrado precedido do símbolo (#>). Esses valores são os resultados que esperam-se sejam reproduzidos também na sua sessão do R. Por exemplo:

```
1:5  
#> [1] 1 2 3 4 5
```

3.1.2 Linha de comando do R

No Linux o R pode ser aberto simplesmente digitando em um terminal a letra R.

```
$ R
```

```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"  
Copyright (C) 2018 The R Foundation for Statistical Computing  
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.
```

```
  Natural language support but running in an English locale
```

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.
```

```
>
```

A janela com a linha de comando do R apresenta o *prompt* do R (>). Após este símbolo digitamos os comandos, pressionamos a tecla <enter>, o R interpreta o comando e retorna o resultado.

Os comandos digitados na linha de comando são chamados de expressões. Esse é o modo iterativo do R. Portanto, a linha de comando é a mais importante ferramenta do R, pois todas expressões são avaliadas através dela.

```
62 + 38
#> [1] 100
```

A expressão é avaliada pelo R, o resultado é mostrado, mas o seu valor é perdido.

O número entre colchetes que aparece como resultado da operação (“[1]” no caso acima) indica o conteúdo resultante da operação iniciando na posição 1 desse objeto. O significado dessa informação torna-se mais óbvio quando trabalhamos com objetos maiores, como por exemplo com vetores. Observe os valores nos colchetes para uma sequência de 100 até 1.

```
100:1
#> [1] 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84
#> [18] 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67
#> [35] 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50
#> [52] 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33
#> [69] 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
#> [86] 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

O elemento [18] da sequência de 100 até 1 é o número 83.

Pode ocorrer da expressão digitada na linha ser muito extensa e ir além de uma linha. Se a expressão estiver incompleta o R mostra um sinal de +.

```
1 * 2 * 3 * 4 * 5 *
6 * 7 * 8 * 9 * 10
#> [1] 3628800
```

Execute a expressão abaixo até o sinal de menos e tecle <enter>. Enquanto a expressão não estiver completa o sinal de + se repetirá. Até que você digite o número que deseja subtrair de 4.

```
4 -
3
#> [1] 1
```

3.2 Expressões em sequência

Podemos executar todas expressões anteriores em apenas uma linha, usando o ponto e vírgula ; para separar as expressões:

```
62 + 38; 100:1; 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10; 4-3
#> [1] 100
#> [1] 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84
#> [18] 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67
#> [35] 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50
#> [52] 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33
#> [69] 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
#> [86] 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
#> [1] 3628800
#> [1] 1
```

As expressões também podem ser separadas por uma nova linha (<enter>) entre cada um delas.

```

62 + 38
#> [1] 100
100:1
#> [1] 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84
#> [18] 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67
#> [35] 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50
#> [52] 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33
#> [69] 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
#> [86] 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10
#> [1] 3628800
4-3
#> [1] 1

```

3.3 Navegação entre as expressões já avaliadas

Você pode usar as teclas `e` e `l` para navegar entre as expressões já avaliadas pelo R. O que é útil quando precisamos repetir um comando anterior com alguma mudança ou para corrigir um erro de digitação ou a omissão de um parenteses.

Quando a linha de comando é usada por muito tempo a sua tela pode ficar poluída com a saída das expressões anteriores. Para limpar a tela, tecle `Ctrl+l`. Assim o console aparece na parte superior do terminal.

```

15 + 4
#> [1] 19
100:1
#> [1] 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84
#> [18] 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67
#> [35] 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50
#> [52] 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33
#> [69] 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
#> [86] 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
#tecle <Ctrl + l>

```

Para parar ou cancelar a execução de uma expressão utilize as teclas `Ctrl + C`. As teclas `Ctrl + l` tem o efeito de limpar a tela.

3.4 Comentários

No R, a cerquilha `#` (hashtag) é um caracter especial. Qualquer coisa após esse caracter será ignorada pelo R. Somente as expressões antes da `#` são avaliadas. Por meio desse símbolo de comentário podemos fazer anotações e comentários no código sem atrapalhar a interpretação das expressões pelo R.

```

17 + 3 # adicionando 17 e 3
#> [1] 20
# A expressão abaixo será ignorada
# 10 + 4

```

3.5 Auto preenchimento de funções

O R inclui o preenchimento automático de nomes de funções e arquivos por meio da tecla <tab>. Uma lista de possíveis funções que começam com as letras inicialmente digitadas aparecerão.

```
read#<tab> pressione <tab> para ver as opções de comandos que iniciam com o termo read
```

3.6 R no modo não interativo (*Batch*)

O uso da linha de comando do R é conveniente durante a análise de dados. Mas para algumas tarefas digitar cada comando pode se tornar inconveniente.

Por exemplo, você precisa visualizar graficamente os padrões de variação temporal de três variáveis meteorológicas antes e após a passagem sistemas meteorológicos precipitantes. Para esse propósito você precisará ler os dados da estação meteorológica (EM) que estão armazenados em um arquivo, fazer algumas conversões de unidades, plotar o gráfico com as séries temporais das variáveis e salvá-lo em um arquivo pdf.

Suponha que você precise repetir esse processo para outras EM. Com o R, assim como em outras linguagem de programação, há uma forma de rodar um conjunto de comandos em sequência e salvar os resultados em um arquivo. Isso é possível através do modo *Batch*, ou seja a execução dos comandos sem iniciar o R e sem a intervenção manual. Esse modo facilita a automatização de tarefas. Nesse caso, alterando apenas a variável que define o nome do arquivo da estação meteorológica, podemos reaplicar os mesmos comandos para outra EM.

Por exemplo, para rodar um suposto script `gera_graficos.R`, sem ter que abrir o R e digitar os comandos para fazer o gráfico, utilizaríamos o comando:

```
$ R CMD BATCH gera_graficos.R
```

Podemos criar um arquivo texto chamado `script1.R` e digitar todos comandos dessa aula nele.

```
$ cd /home/lsi/nomealuno
```

```
$ gedit script1.R &
```

Copie os comandos abaixo num arquivo novo criado no gedit ou outro editor de texto do Linux.

```
62 + 38
1:100
1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10
4-3
## salvando um gráfico da sequência de 1 a 100 em um arquivo pdf
pdf("plot_script1.pdf") # cria e abre um arquivo pdf
plot(1:100)             # faz o gráfico
dev.off()               # fecha o arquivo pdf
```

Rode o R em modo **Batch** informando o nome do script.

```
$ R CMD BATCH script1.R
```

O R rodará os comandos dentro do arquivo especificado `script1.R` e criará um arquivo de saída chamado `script1.Rout` com os resultados. Você também pode especificar o nome daquele arquivo de saída. Por exemplo vamos salvar o arquivo de saída como `saida_script1_aDataDeHoje.log`

```
$ R CMD BATCH script1.R saida_script1_`date +%y%m%d` .log
```

Para mais informações sobre rodar o R na linha de comando do Linux digite no terminal `$ R --help`.

3.7 Rodando script de dentro do R

No modo iterativo, ou seja pela linha de comando, também é possível rodar comandos no modo *Batch* através da função `source()`.

```
source("R/script1.R")
```


Chapter 4

Operações básicas

Os solos são caracterizados por uma típica sequência de horizontes que constituem o perfil do solo.

4.1 Formação do solo

4.2 Propriedades do solo

4.2.1 Materiais

4.2.2 Hídricas

4.3 Textura e Estrutura

4.4 Movimento da água no solo

4.5 Infiltração

4.5.1 Modelo de frente de umedecimento de Green-Ampt

asd

Chapter 5

Tipos de dados

A EVAPORAÇÃO ocorre quando uma superfície úmida está exposta a ar relativamente mais seco. Quando parcelas de ar movem-se sobre a superfície elas carregam umidade daquela superfície.

A água evaporada da superfície aumenta a quantidade de vapor d'água no ar. Quando o ar está saturado com vapor d'água a evaporação cessa.

A TRANSPIRAÇÃO é a evaporação da água das folhas das plantas quando esta move-se do solo, através das plantas e através das folhas para o ar.

As plantas consomem grandes quantidades de água durante o crescimento. Um lavouro de milho de 4000 m² pode consumir 10000-15000 litros de água (2,5 - 3.75 mm) em um dia. Uma árvore sem restrição hídrica pode transpirar 100-150 litros por dia. Os processos meteorológicos próximo a superfície controlam a EVAPORAÇÃO e TRANSPIRAÇÃO. A TRANSPIRAÇÃO é também regulada pela fisiologia das plantas. Quando as plantas cobrem uma pequena porção do solo a evaporação do solo é o fluxo dominante. A TRANSPIRAÇÃO torna-se mais importante com o aumento da cobertura de área das plantas aumenta. Entretanto é difícil distinguir EVAPORAÇÃO de TRANSPIRAÇÃO e os dois termos são frequentemente combinados em EVAPOTRANSPIRAÇÃO.

EVAPORAÇÃO TRANSPIRAÇÃO EVAPOTRANSPIRAÇÃO

5.1 Evapotranspiração de referência

5.2 Evapotranspiração da cultura

5.3 Evapotranspiração real

Bibliography

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.