

Linguagem de Programação Estatística para Biologia



Prof. Dr. Diogo B. Provete
Setor de Ecologia
Instituto de Biociências

Programa do curso

- Introdução ao R
- Organização e curadoria dos dados
- Importação e manipulação de dados
- Controle de versão (git e github)*
- Análise (modelos lineares simples)
- Gráficos (ggplot2)
- Rudimentos de programação
- Reprodutibilidade em pesquisa (RMarkdown)

Avaliação

- Atividades diárias feitas por vocês
- Scripts -> parte da avaliação (P1)
- Trabalho final com Markdown/Quarto (P2)

Cronograma

- Segunda manhã
 - Ambientação no programa R e no IDE RStudio
 - Tipos de objetos
 - Como obter ajuda
 - Como escrever e organizar um script
- Segunda tarde
 - Organização dos dados no seu computador
 - Boas práticas
 - Curadoria de dados

Cronograma disciplina PPG

- Terça manhã
 - Controle de versão com git e github
- Terça tarde
 - Manipulação de dados, estratégia split-apply-combine: o tidyverse
 - Subsetting, summarize, group_by, do., pipe operator (%>%)

Cronograma disciplina PPG

- Quarta manhã
 - Análises simples com modelos lineares
 - Gráficos com ggplot2, ggmap
- Quarta tarde
 - Exercícios com gráficos
 - Análises com tidyverse, modelos lineares,
 - Rudimentos de programação

Cronograma disciplina PPG

- Quinta manhã
 - Reprodutibilidade em pesquisa, Markdown, e knitr
 - Como organizar um markdown
- Quinta tarde
 - Exercícios com markdown
- Sexta manhã
 - Tira dúvidas

Aula 1

Materiais da disciplina inteiramente no Moodle

<https://ava.ufms.br/course/view.php?id=8342>

Senha: adorooR

Baixando e instalando o R e o RStudio

- www.r-project.org
- <https://www.rstudio.org>
- Versão base (Windows)
- Manuais e listas de discussão no site
- “Fichas resumo” (reference cards)

RStudio Cheat Sheets

The cheat sheets below make it easy to learn about and use some of our favorite packages. From time to time, we will add new cheat sheets to the gallery. If you'd like us to drop you an email when we do, let us know by clicking the button to the right.

SUBSCRIBE TO CHEAT SHEET UPDATES HERE

Deep Learning with Keras Cheat Sheet

Keras is a high-level neural networks API developed with a focus on enabling fast experimentation. Keras supports both convolution based networks and recurrent networks (as well as combinations of the two), runs seamlessly on both CPU and GPU devices, and is capable of running on top of multiple back-ends including TensorFlow, CNTK, and Theano. Updated 12/17.

DOWNLOAD

Deep Learning with Keras : : CHEAT SHEET

Intro

Keras is a high-level neural networks API developed with a focus on enabling fast experimentation. It supports multiple back-ends, including TensorFlow, CNTK, and Theano.

TensorFlow is a lower level mathematical library for building deep neural network architectures. The Keras R package makes it easy to use Keras and TensorFlow in R.

<https://keras.rstudio.com>
<https://www.manning.com/books/deep-learning-with-r>

INSTALLATION

The keras R package uses the Python keras library. You can install all the prerequisites directly from R.

```
install.packages(c('keras', 'tensorflow'))
```

[See Notes: install for GPU instructions!](#)

This installs the required libraries in an Anaconda environment or virtual environment 'tensorflow'.

Working with keras models

DEFINE A MODEL

```
keras_model() Keras Model
keras_model_sequential() Keras Model composed of a linear stack of layers
multi_gpu_model() Replicates a model on different GPUs
```

COMPILE A MODEL

```
compile(object, optimizer, loss, metrics = NULL) Configure a Keras model for training
```

FIT A MODEL

```
fit(object, x = NULL, y = NULL, batch_size = NULL, epochs = 10, verbose = 1, callbacks = NULL, ...) Train a Keras model for a fixed number of epochs (iterations)
fit_generator() Fits the model on data yielded batch-by-batch by a generator
train_on_batch() test_on_batch() Single gradient update or model evaluation over one batch of samples
```

EVALUATE A MODEL

```
evaluate(object, x = NULL, y = NULL, batch_size = NULL) Evaluate a Keras model
evaluate_generator() Evaluates the model on a data generator
```

PREDICT

```
predict() Generate predictions from a Keras model
predict_prob() and predict_classes() Generates probability or class probability predictions for the input samples
predict_on_batch() Returns predictions for a single batch of samples
predict_generator() Generates predictions for the input samples from a data generator
```

OTHER MODEL OPERATIONS

```
summary() Print a summary of a Keras model
export_saved_model() Export a saved model
get_layer() Retrieves a layer based on either its name (unique) or index
pop_layer() Remove the last layer in a model
save_model_hdf5(), load_model_hdf5() Save/Load models using HDF5 files
serialize_model(), unserialize_model() Serialize a model to an R object
clone_model() Clone a model instance
freeze_weights(), unfreeze_weights() Freeze and unfreeze weights
```

CORE LAYERS

```
layer_input() Input layer
layer_dense() Add a densely-connected NN layer to an output
layer_activation() Apply an activation function to an output
layer_dropout() Applies Dropout to the input
layer_reshape() Reshapes an output to a certain shape
layer_permute() Permute the dimensions of an input according to a given pattern
layer_repeat_vector() Repeats the input n times
layer_lambda(object, f) Wraps arbitrary expression as a layer
layer_activity_regularization() Layer that applies an update to the opt function based input activity
layer_masking() Masks a sequence by using a mask value to skip timesteps
layer_flatten() Flattens an input
```

TRAINING AN IMAGE RECOGNIZER ON MNIST DATA

```
# Input layer: use MNIST images
mnist <- dataset_mnist()
x_train <- mnist$train$x; y_train <- mnist$train$y
x_test <- mnist$test$x; y_test <- mnist$test$y

# reshape and rescale
x_train <- array_reshape(x_train, c(nrow(x_train), 784))
x_test <- array_reshape(x_test, c(nrow(x_test), 784))
x_train <- x_train/255; x_test <- x_test/255
y_train <- to_categorical(y_train, 10)
y_test <- to_categorical(y_test, 10)

# defining the model and layers
model <- keras_model_sequential()
model %>% layer_dense(units = 256, activation = 'relu', input_shape = c(784)) %>% layer_dropout(rate = 0.4) %>% layer_dense(units = 128, activation = 'relu') %>% layer_dense(units = 10, activation = 'softmax')

# compile (define loss and optimizer)
model %>% compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = c('accuracy'))

# train (fit)
model %>% fit(x_train, y_train, epochs = 10, batch_size = 128, validation_split = 0.2)
model %>% evaluate(x_test, y_test)
model %>% predict_classes(x_test)
```

RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • rstudio.com • 844 488 2232 • rstudio.com • Learn more at keras.rstudio.com • keras 3.2.2 • Updated 2017.12

The image shows a screenshot of the JDroid AI Online R Language IDE interface. The interface is dark-themed and consists of several main sections:

- Top Bar:** Includes the JDroid logo, the text "Online R Language IDE", an "Execute" button with a play icon, a search icon, and a "Login" button.
- Code Editor:** On the left, a code editor displays the following R code:

```
1 x <- 10
2 y <- 25
3 z <- sum(x,y)
4
5 cat("x + y = ", z)
```
- JDroid AI Chat Panel:** On the right, a chat panel titled "JDroid AI" features the JDroid logo and the text "Hello, What should we build today?". Below this is a text input field with the placeholder "Generate a prompt or ask question to JDroid here...".
- AI Capabilities:** Under the heading "Explore full coding abilities of JDroid", there are three interactive buttons:
 - Optimize code:** "Modify code for better performance."
 - Debug code:** "Fix error and bug in your code."
 - Explain code:** "Get detailed explanation of how your code works."
- Footer:** At the bottom, there are social media icons (Facebook, X, Email, LinkedIn, Instagram, YouTube), a "LEAVE A REVIEW ON Product Hunt" button with a star icon, and an "I need help" button.

<https://www.jdoodle.com/execute-r-online>

```
Untitled1.R* x  Run Save Share Upload file New file New directory Download directory Open in editor Delete
1 # Dataset overview and summary statistics
2 head(mtcars)
3 summary(mtcars)
4
5 # Visualization: fuel efficiency vs vehicle weight
6 plot(mtcars$wt, mtcars$mpg,
7       xlab = "Weight (1000 lbs)",
8       ylab = "Miles per Gallon",
9       main = "Fuel Efficiency vs Vehicle Weight",
10      pch = 19)
11
12 # Fitted linear regression line
13 trend <- lm(mpg ~ wt, data = mtcars)
14 abline(trend, col = "red", lwd = 2)
15
16 # Calculate correlation
17 cor(mtcars$mpg, mtcars$wt)

R version 4.5.1 (2025-06-13) -- "Great Square Root"
Copyright (C) 2025 The R Foundation for Statistical Computing
Platform: wasm32-unknown-emscripten (32-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.

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.

>
```

/

- dev
- etc
- home
- proc
- tmp
- usr
- var

Save Plot Clear Plots

WebR 0.2

R IN THE BROWSER

GEORGE STAGG



Contributed Documentation

Note: The CRAN area for contributed documentation is frozen and no longer actively maintained.

[English](#) --- [Other Languages](#)

Manuals, tutorials, etc. provided by users of R. The R core team does not take any responsibility for contents, but we appreciate the effort very much and encourage everybody to contribute to this list! To submit, follow the submission instructions on the [CRAN main page](#). All material below is available directly from CRAN, you may also want to look at the list of [other R documentation](#) available on the Internet.

Note: Please use the [directory listing](#) to sort by name, size or date (e.g., to see which documents have been updated lately).

English Documents

Documents with more than 100 pages:

- **“Visual Statistics. Use R!”** by Alexey Shipunov ([PDF](#), 2017-05-15, 388 pages). All book materials are accessible from [Alexey Shipunov's English R page](#).
- **“Using R for Data Analysis and Graphics - Introduction, Examples and Commentary”** by John Maindonald ([PDF](#), data sets and scripts are available at [JM's homepage](#)).
- **“Practical Regression and Anova using R”** by Julian Faraway ([PDF](#), data sets and scripts are available at the [book homepage](#)).
- The [Web Appendix](#) to the book “An R Companion to Applied Regression” (second edition) by John Fox and Sanford Weisberg contains information about R to fit a variety of regression models.
- **“An Introduction to S and the Hmisc and Design Libraries”** by Carlos Alzola and Frank E. Harrell, especially of interest to SAS users, users of the Hmisc or Design packages, or R users interested in data manipulation, recoding, etc. ([PDF](#)).
- **“Statistical Computing and Graphics Course Notes”** by Frank E. Harrell, includes material on S, LaTeX, reproducible research, making good graphs, brief overview of computer languages, etc. ([PDF](#)).
- **“An Introduction to R: Software for Statistical Modelling & Computing”** by Petra Kuhnert and Bill Venables ([ZIP 3.8MB](#)): A 360 page PDF document of lecture notes in combination with the data sets and R scripts used in the manuscript.
- **“Introduction to the R Project for Statistical Computing for Use at the ITC”** by David Rossiter ([PDF](#), 2012-08-20, 141 pages).
- **“Analysis of Epidemiological Data Using R and Epicalc”** by Virasakdi Chongsuvivatwong ([PDF](#)).
- **“Statistics Using R with Biological Examples”** by Kim Seefeld and Ernst Linder ([PDF](#)).
- **“IcebreakeR”** by Andrew Robinson ([PDF](#), 2016-06-21, 161 pages).
- **“Applied Statistics for Bioinformatics Using R”** by Wim Krijnen ([PDF](#), 2009-11-17, 278 pages).
- **“An Introduction to R”** by Longhow Lam ([PDF](#), 2010-10-28, 212 pages).
- **“R and Data Mining: Examples and Case Studies”** by Yanchang Zhao ([PDF](#), 2013-04-26, 160 pages).
- **“A Student's Guide to R”** by Nicholas J. Horton, Randall Pruim, and Daniel T. Kaplan ([PDF](#), 2015-11-16, 119 pages).
- **“Is R Suitable Enough for Biostatisticians?”** by Adrian Olszewski ([PDF](#), 2015-06-28, 365 pages).

Documents with fewer than 100 pages:

- **“R for Beginners”** by Emmanuel Paradis ([PDF](#)).
- **“Kickstarting R (version 1.6)”** compiled by Jim Lemon, a short introduction in English as HTML files: download as [zipped TAR](#) or [ZIP](#); or [browse directly](#).
- **“Notes on the use of R for psychology experiments and questionnaires”** by Jonathan Baron and Yuelin Li ([PDF](#)). A browsable version is available at [JB's homepage](#).
- **“R for Windows Users (version 2.0)”** by Ko-Kang Wang ([PDF](#), [LaTeX source](#)).
- **“Building Microsoft Windows Versions of R and R packages under Intel Linux”** by Jun Yan and A. J. Rossini ([PDF](#), associated [Makefile](#)).
- **“A Guide for the Unwilling S User”** by Patrick Burns ([PDF](#)).
- **“The R language — a short companion”** by Marc Vandemeulebroecke ([PDF](#)).
- **“Fitting Distributions with R”** by Vito Ricci ([PDF](#)).

<https://cran.r-project.org/other-docs.html>

Explorando o site do R

- Manuais/tutoriais
- R-SIG-XXX mailing lists
- R-Br
 - <http://www.leg.ufpr.br/doku.php/software:rbr>
- CRAN e Bioconductor
 - 20 mil + pacotes disponíveis
 - Como navegar entre eles?
- TaskViews
 - <http://nbcgib.uesc.br/mirrors/cran/web/views>

Enquanto instala...

- Programação orientada ao objeto
 - Controle do que se quer fazer, conhecimento de todos os passos da análise
- Necessidade de se conhecer uma linguagem de programação
- Disponibilidade gratuita, amplamente utilizado por biólogos => novas análises

EDITORIAL

Ten simple rules for biologists learning to program

Maureen A. Carey¹, Jason A. Papin^{2*}

Language	Key features	Documentation	Sample tutorials	Community groups
R	<ul style="list-style-type: none"> • Community involvement • Application-focused development • Easy to learn by coupling basic programming and applications • Well-developed visualization • Variable package quality • “Tidy data” community • Interpreted language 	<ul style="list-style-type: none"> • rdocumentation.org • r-project.org • cran.r-project.org 	<ul style="list-style-type: none"> • R for cats: rforcats.net • Books by Hadley Wickham: hadley.nz • R Tutorial’s introduction: r-tutor.com/r-introduction • Cyclismo’s R Tutorial: cyclismo.org/tutorial/R/ 	<ul style="list-style-type: none"> • R-Ladies: rladies.org • R Users Group: many

EMILY DREYFUSS SCIENCE 03.10.17 12:00 PM

WANT TO MAKE IT AS A BIOLOGIST? BETTER LEARN TO CODE

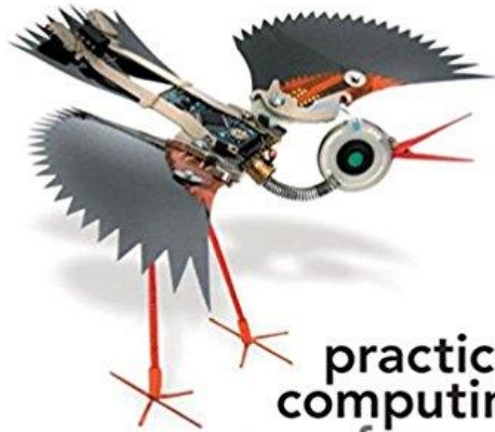
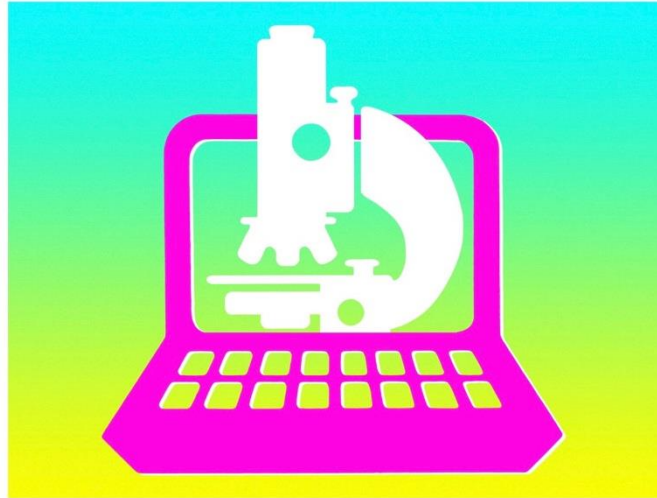
SHARE

SHARE 3779

TWEET

COMMENT

EMAIL



practical
computing
for
biologists

HADDOCK • DUNN

Ad closed by Google

MOST POPULAR



CULTURE
"Black Museum": Why the Controversial 'Black Mirror' Episode Is the...
JASON PARHAM



CULTURE
Does This 'Black Mirror' Fan Theory Mean We're Finally Ready For the...
MIRANDA KATZ



SPONSOR CONTENT
Shyama Rose: Never-Ending Challenge of Cybersecurity
JUNIPER

For students entering my lab, I place a premium on **quantitative skills**, intellectual creativity, hard work, and independence. I am not interested in students who take a casual, leisurely approach to graduate school or have applied here because they would like to live in Vermont.

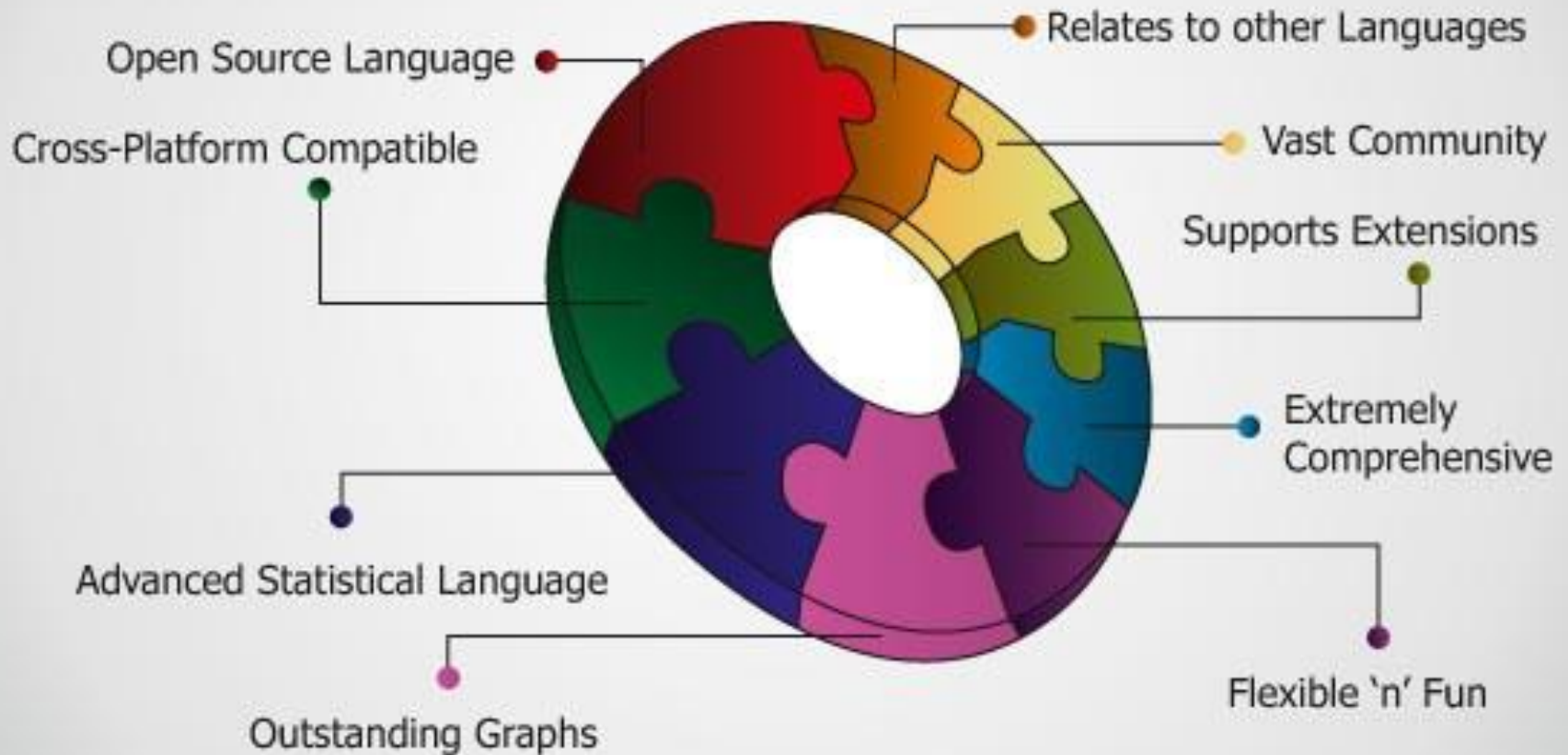


What kinds of skills do you need for success in community ecology?

You should invest the time in learning a programming/statistical language... R is the language that I recommend... there is no better (or more enjoyable) way to learn about statistics and ecological theory than by programming.



Why Learn R?



Plataformas comerciais com o R

Database

Database In-Memory

Multitenant

More Key Features

Application Development

Big Data Appliance

Cloud Database Services

Private Database Cloud

Data Warehousing & Big Data

Database Appliance

Exadata Database Machine

High Availability

Manageability

Migrations

Security

Unstructured Data

Upgrades

Windows

Database Technology Index

Multilingual Engine

Overview

Downloads

Documentation

Community

Learn More

R Technologies from Oracle



Oracle has adopted R as a language and environment to support Statisticians, Data Analysts, and Data Scientists in performing statistical data analysis and advanced analytics, as well as generating sophisticated graphics. In addressing the enterprise and the need to analyze Big Data, Oracle provides R integration through four key technologies:



Oracle R Distribution

Oracle's supported redistribution of open source R, provided as a free download from Oracle, enhanced with dynamic loading of high performance linear algebra libraries.



Oracle R Enterprise

Integration of R with Oracle Database. A component of the Oracle Advanced Analytics Option. Oracle R Enterprise makes the open source R statistical programming language and environment ready for the enterprise with scalability, performance, and ease of production deployment.

Oracle R Advanced
Analytics for Hadoop

High performance native access to the Hadoop Distributed File System (HDFS) and MapReduce programming framework for R users. Oracle R Advanced Analytics for Hadoop is a component of Oracle Big Data Connectors software suite.



ROracle

An open source R package, maintained by Oracle and enhanced to use the Oracle Call Interface (OCI) libraries to handle database connections - providing a high-performance, native C-language interface to Oracle Database.

Why Oracle for Advanced Analytics?

If you're an enterprise company, chances are you have your data in an Oracle database. You chose

Try Database in the
Cloud for Free

See what you can accomplish
with US\$300 in free credits.

Free cloud trial >

Analytics and
Data Summit

ALL ANALYTICS. ALL DATA.
NO NONSENSE.

MARCH 20-22, 2018

Formerly BIWA Summit

EARLY SCHEDULE
NOW LIVE!

Oracle Live SQL

Learn and Share SQL

Try Now >

Bring the world closer with Bing Wallpaper

Download the free app and enjoy breathtaking views with a new background each day.

Get Bing Wallpaper



Microsoft R Open 3.2.2 for Revolution R Enterprise 8.0.1

Microsoft R Open is a free, complete open source platform for statistical analysis and data science. It includes the open source distribution of R 3.2.2 as well as some enhanced R features, and supports a variety of big data statistics, predictive modeling, and machine learning capabilities.

Important! Selecting a language below will dynamically change the complete page content to that language.

Select language

English

Download

Expand all | [Collapse all](#)

> Details

Version:

Date Published:

↑ Back To Top

<https://www.microsoft.com/en-in/download/details.aspx?id=51205>

Workspace do R

- Linguagem interpretada
- Prompt: > e +
- `getwd()`
- Arquivo>Mudar dir...
 - Faça uma pasta para este curso e mude o diretório para ela. Sempre que iniciar um conjunto de análises novo faça o mesmo. Aparecerá um símbolo do R na pasta que guardará todos os objetos daquela sessão

```
R Console
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.

[R.app GUI 1.70 (7463) x86_64-apple-darwin15.6.0]

[Workspace restored from /.RData]
[History restored from /.Rapp.history]

>
```

```
Untitled
<functions>
1
```

TESE_mirna
Delatorr...INAL.pdf

journal.pcbi.
1005709.pdf



10.1111@oik.
05168.pdf

RStudio interface showing the source editor, workspace, and console.

Source Editor:

```

1 library(ggplot2)
2 source("plots/formatPlot.R")
3
4 view(diamonds)
5 summary(diamonds)
6
7 summary(diamonds$price)
8 aveSize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
10
11
12 var_global <- 1
13 fn <- function(var_argument) {
14   var_local <- 2
15   var_
16 }
17 }
18
19
20
21

```

Workspace:

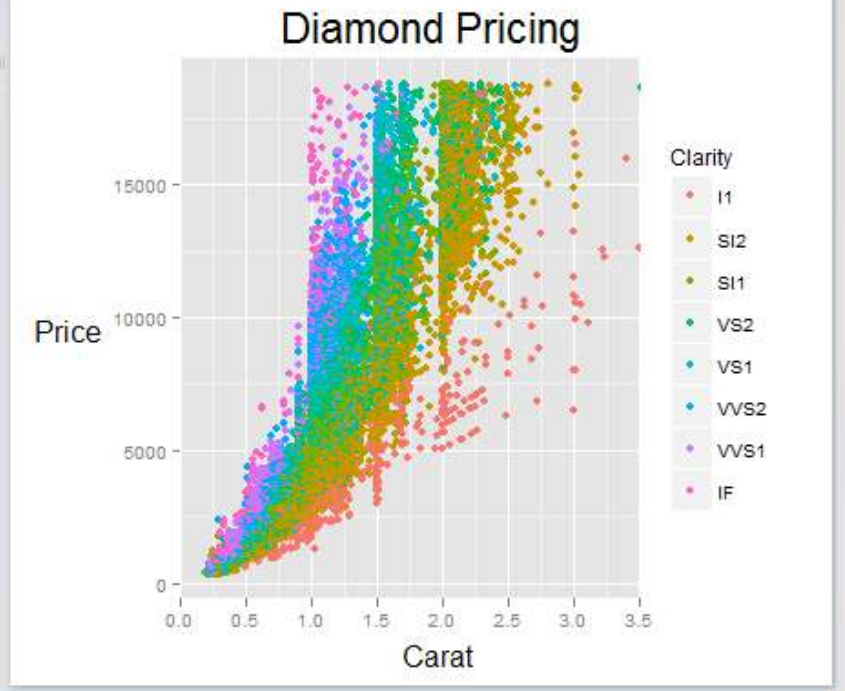
- Data: diamonds (53940 obs. of 10 variables)
- Values:
 - aveSize: 0.7979
 - clarity: character [8]
 - p: ggplot [8]
- Functions: format.plot(plot, size)

Console:

```

15:1 (Top Level) R Script
> summary(diamonds)
      x          y          z
Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
Median : 5.700   Median : 5.710   Median : 3.530
Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  326    950    2401   3933   5324  18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+           data=diamonds, color=clarity,
+           xlab="carat", ylab="Price",
+           main="Diamond Pricing")
>
> format.plot(p, size=24)
>

```



mean.R × dataframe.R × grob.R × NEWS ×

Source on Save Run Source

```

123 as.data.frame <- function(x, row.names = NULL, optional = FALSE)
124 {
125   if(is.null(x)) # can't assign class to NULL
126     return(as.data.frame(list()))
127   UseMethod("as.data.frame")
128 }

```

```

130 as.data.frame.default
131   stop(gettextf("c
132     dep
133     domain = NA
134

```

```

135 ### Here are method
136 ### are in a form s
137

```

```

138 as.data.frame.data.f
139 {

```

```

140   cl <- oldClass(x
141   i <- match("data
142   if(i > 1L)

```

```

143   class(x) <- cl[ -
144   if(!is.null(row.n
145     nr <- .row_n

```

```

146   if(length(row.name
147     attr(x, "row.n
148   else stop(gettextf

```

```

149
150 }

```

```

151 x
152 }

```

```

154 ## prior to 1.8.0 this coerced names - PR#3280

```

```

155 as.data.frame.list <-

```

```

156   function(x, row.names = NULL, optional = FALSE, ...,
157     stringsAsFactors = FALSE, default.stringsAsFactors =

```

```

123:1 as.data.frame ↕ R Script ↕

```

Console

Workspace History

Load Save Import Dataset Clear All

Values

names character[5]

rate integer[124]

Functions

as.data.frame(x, row.names = NULL, optional = FALSE, ...)

New Project

Create project from:



New Directory

Start a project in a brand new working directory



Existing Directory

Associate a project with an existing working directory



Version Control

Checkout a project from a version control repository

Cancel

Modified

Mar 17, 2010, 10:43 AM

Mar 29, 2011, 6:02 PM

Dec 14, 2011, 6:06 PM

Nov 9, 2011, 6:05 PM

Mar 17, 2010, 10:43 AM

etc			
INSTALL	1.7 KB	Mar 17, 2010, 10:43 AM	
m4			
Makeconf.in	4.3 KB	Mar 17, 2011, 7:05 PM	
Makefile.fw	7 KB	Oct 2, 2011, 6:02 PM	
Makefile.in	7.1 KB	Oct 2, 2011, 6:02 PM	
NEWS	195.7 KB	Dec 22, 2011, 3:05 AM	

Facilidades para escrever código

- Tab completion
- Snippets

<https://support.rstudio.com/hc/en-us/articles/204463668-Code-Snippets>

Seguindo no livro

<https://analises-ecologicas.com/cap4.html#funcionamento-da-linguagem-r>

Operações aritméticas básicas

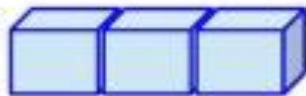
Tipos de objetos

- Função
- Vetor
 - Caracter
 - Lógico
 - Numérico
- Matriz
- Data frame
- Lista

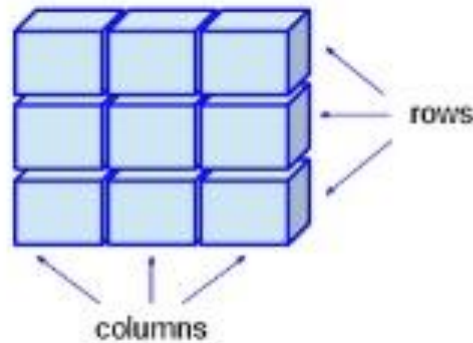
Data structure types



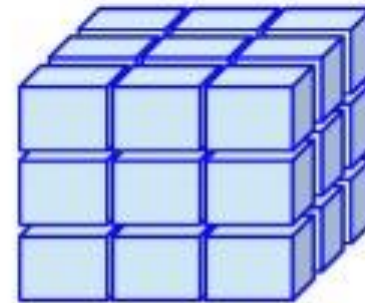
Vector



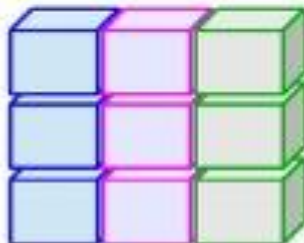
Matrix



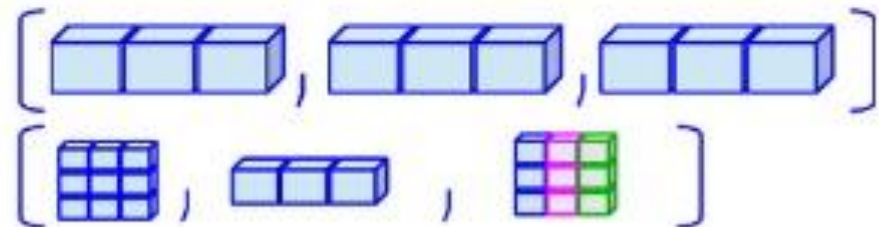
Array



Data Frame
(Table)



Lists



Vectors

Atomic vectors

Logical

Numeric

Integer

Double

Character

List

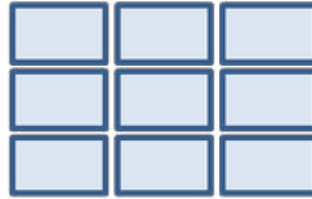
NULL

Vector



- 1 column or row of data
- 1 type (numeric or text)

Matrix



- multiple columns and/or rows of data
- 1 type (numeric or text)

Data Frame



- multiple columns and/or rows of data
- multiple types

Entendendo o arquivo de ajuda

- ? / help()
help.search("regression")
- ?aov
- Componentes do arquivo de ajuda
- www.rseek.org
- StackExchange
 - CrossValidated (stats.stackexchange.com)

Instalando e carregando pacotes

- Função `install.packages()`
- `install.packages("vegan")`
- `library(vegan)`
- `data(BCI)`

Importação de dados

- Alternativas para importar dados:
- `>obj <- read.table(file.choose(), header=TRUE)`
- `>obj <- read.table("clipboard", h=T)`
- `>obj <- read.table("nomedoarquivo.txt", h=T)`
- `>obj <- read.csv(file.choose(), h=T)`

Importação de dados

- Alternativas para importar dados:
- `>obj <- read.table(file.choose(), header=TRUE)`
- `>obj <- read.table("clipboard", h=T)`
- `>obj <- read.table("nomedoarquivo.txt", h=T)`
- `>obj <- read.csv(file.choose(), h=T)`

Exportação de dados

- `>write.table(nomeDoObjeto, "NomeDoObjetoParaSerGravado.extensão", sep=" ", quote=F, dec=".")`
- `>sink("japi-so.xls")` #Exporta pra o wd o(s) objetos que forem exibidos depois com o nome que for colocado nesta linha de comando
- `>japi.so1`
- `>sink()#Fecha o dispositivo`