Iago Mosqueira

Introduction to R (and computing)

May 27, 2013

Why programming?

"Can one be a good data analyst without being a half-good programmer? The short an- swer to that is, 'No'. The long answer to that is, 'No!'."

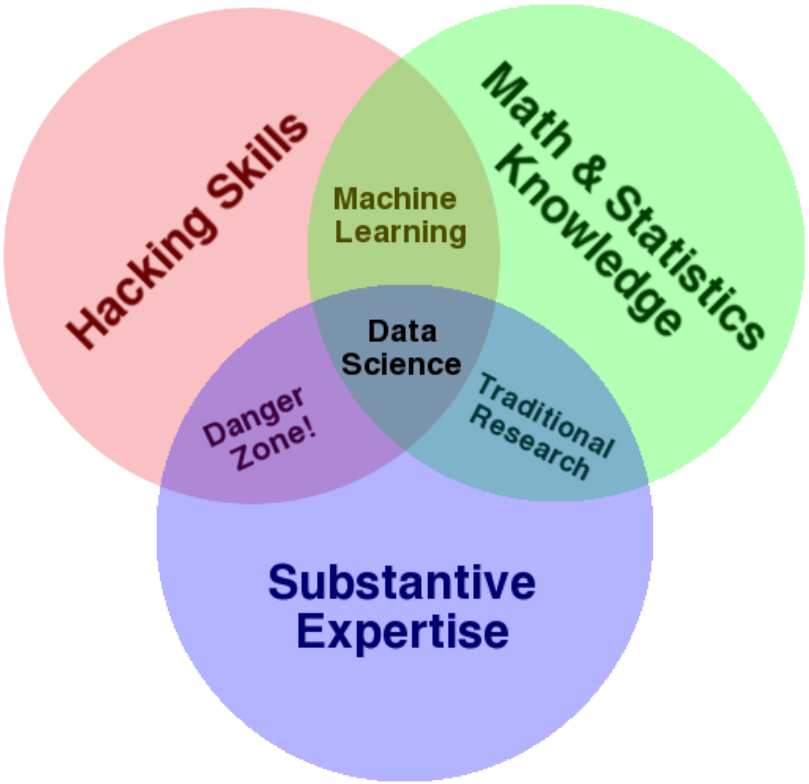
– Frank Harrell, 1999 S-PLUS User Conference, New Orleans (October 1999)

But this should be easy

"Managing fisheries is hard: it's like managing a forest, in which the trees are invisible and keep moving around"

- Professor John Shepherd

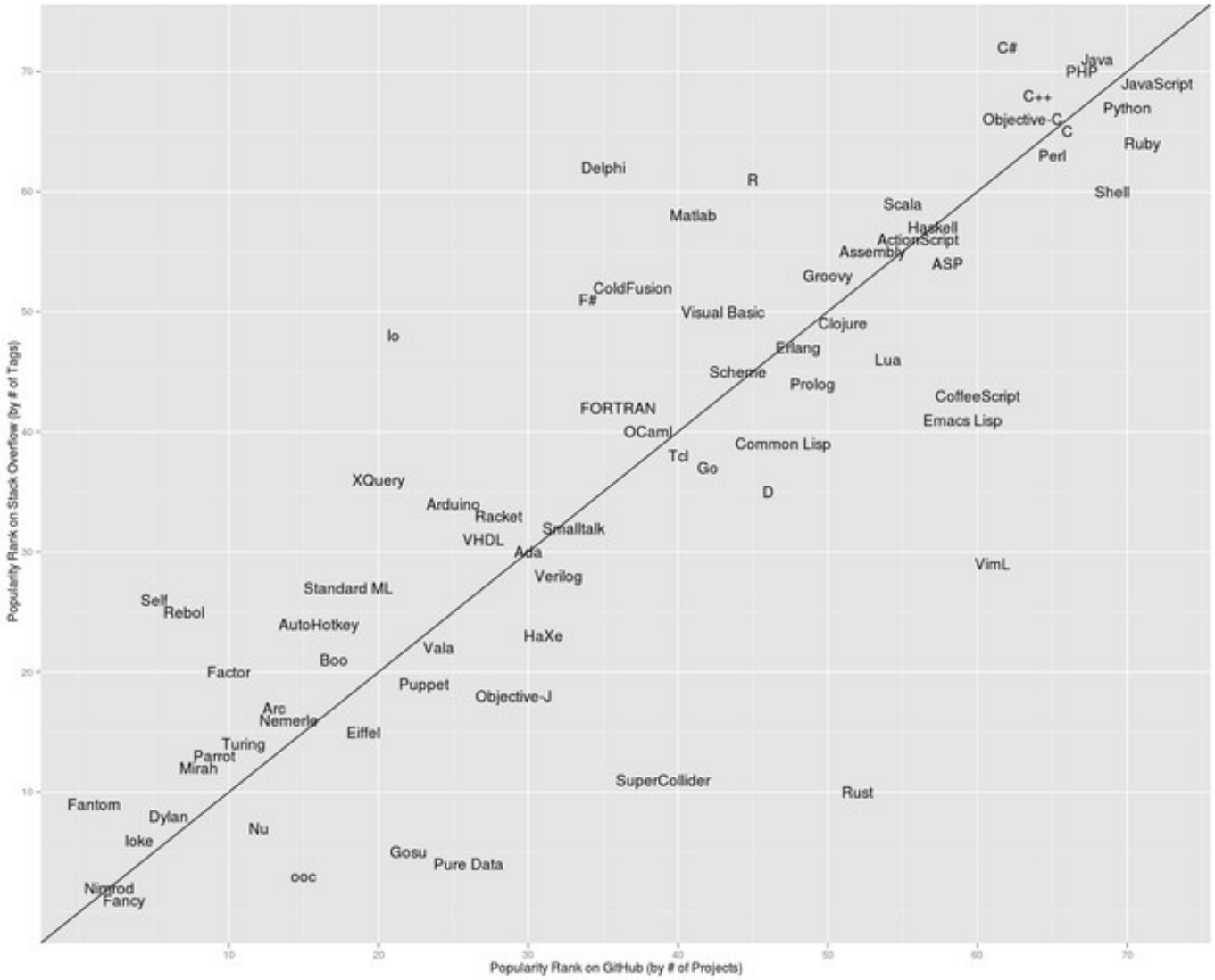
Data analyst



What is R

- Data analysis and statistics environment
- Interpreted computer language
- Open-source software project
- Active community of developers and practicioners
- Current version: 2.15.1, 2012-06-22, Roasted Marshmallows

Why R?


Popularity Rank on Stack Overflow (by # of Tags)
Popularity Rank on GitHub (by # of Projects)
C#
Java
PHP
JavaScript
C++
Python
Objective-C
C
Ruby
Perl
Shell
Delphi
R
Scala
Matlab
Haskell
ActionScript
Assembly
ASP
Groovy
ColdFusion
F#
Visual Basic
Clojure
Io
Erlang
Lua
Scheme
Prolog
CoffeeScript
FORTRAN
Emacs Lisp
OCaml
Common Lisp
Tcl
Go
XQuery
D
Arduino
Racket
Smalltalk
VHDL
Ada
VimL
Verilog
Standard ML
Self
Rebol
AutoHotkey
HaXe
Vala
Boo
Factor
Puppet
Objective-J
Arc
Nemerle
Eiffel
Turing
Parrot
Mirah
SuperCollider
Rust
Fantom
Dylan
Nu
Ioke
Gosu
Pure Data
ooc
Nimrod
Fancy

OSS = Peer review

Previous Next 1 (1 of 2) Fit Page Width

POLICYFORUM

COMPUTATIONAL SCIENCE

Troubling Trends in Scientific Software Use

"Blind trust" is dangerous when choosing software to support research.

Lucas N. Joppa,[1*] Greg McInerny,[12] Richard Harper,[1] Lara Salido,[3] Kenji Takeda,[1] Kenton O'Hara,[1] David Gavaghan,[2] Stephen Emmott[1]

Software pervades every domain of science (*1*–*3*), perhaps nowhere more decisively than in modeling. In key scientific areas of great societal importance, models and the software that implement them define both how science is done and what science is done (*4*, *5*). Across all science, this dependence has led to concerns around the need for open access to software (*6*, *7*), centered on the reproducibility of research (*1*, *8*–*10*). From fields such as high-performance computing, we learn key insights and best practices for how to develop, standardize, and implement software (*11*). Open and systematic approaches to the development of software are essential for all sciences. But for many scientists this is not sufficient. We across all disciplines that are dependent upon a computational approach.

Surveying Species Distribution Modelers

We surveyed scientists across a single domain, species distribution modeling (SDM) (*15*) [see supplementary materials for details]. This strategic targeting separates our analysis from previous efforts in important ways, allowing an analysis spanning computational skill sets, while addressing the interplay between models and computation. Our ~400 respondents ranged from those who "find it difficult to use software" to those "very experienced and very technical." Asking people to first identify with a scientific domain and addressing models and software through that

used "click-and-run" software with easy-to-manipulate user interfaces and dropped to 11% for those who used "syntax-driven" platforms. Further, 7, 9, and 18% of scientists cited "the developer is well-respected," "personal recommendation," and "recommendation from a close colleague," respectively, as reasons for using software. Only 8% claimed they had validated software against other methods as a primary reason for choice; 79% expressed a desire to learn additional software and programming skills.

Many of these scientists rely on the fact that the software has appeared in a peer-reviewed article, recommendations, and personal opinion, as their reason for adopting software. This is scientifically misplaced, as the software code used to conduct the science

GPL v3.0

GNU General Public License

From Wikipedia, the free encyclopedia

"GPL" redirects here. For other uses, see GPL (disambiguation).

The **GNU General Public License** (**GNU GPL** or simply **GPL**) is the most widely used[5] free software license. It was originally written by Richard Stallman for the GNU Project.

The GPL is the first copyleft license for general use, which means that derived works can only be distributed under the same license terms. Under this philosophy, the GPL grants the recipients of a computer program the rights of the free software definition and uses copyleft to ensure the freedoms are preserved, even when the work is changed or added to. This is in distinction to permissive free software licenses, of which the BSD licenses are the standard examples.

Contents [hide]

GNU General Public License

GNU GPLv3 Logo

Author	Free Software Foundation
Version	3
Publisher	Free Software Foundation, Inc.
Published	29 June 2007
DFSG compatible	Yes[1]
FSF approved	Yes[2]
OSI approved	Yes[3]
Copyleft	Yes[2][4]
Linking from code with a different license	No (except for linking GNU AGPLv3 with GNU GPLv3 – see section)
Website	www.gnu.org/licenses

CRAN

hadley/roxyg... | factbased: Un... | Metadata sta... | Automatic Di... | Home - TCSAI... | My Little Pon... | Today | CRAN - Packa... | The Compr...

cran.r-project.org

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

Contributed Packages

Available Packages

Currently, the CRAN package repository features 4555 available packages.

Table of available packages, sorted by date of publication

Table of available packages, sorted by name

Installation of Packages

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this repository. The manual R Installation and Administration [PDF] (also contained in the R base sources) explains the process in detail.

CRAN Task Views allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 31 views are available.

Package Check Results

All packages are tested regularly on machines running Debian GNU/Linux, Fedora and Solaris. Packages are also checked under Mac OS X and Windows, but typically only on the day the package appears on CRAN.

The results are summarized in the check summary (some timings are also available). Additional details for Windows checking and building can be found in the Windows check summary.

Writing Your Own Packages

The manual Writing R Extensions [PDF] (also contained in the R base sources) explains how to write new packages and how to contribute them to CRAN.

Repository Policies

The manual CRAN Repository Policy [PDF] describes the policies in place for the CRAN package repository.

Related Directories

cran.r-project.org/bin/windows/contrib/checkSummaryWin.html

0:9

Task views

Blog with Knitr and Je... | Welcome | Travelling... | iagomosqueira/ICESSAI | Home - TCSAI2012 | Oráculo manual y art... | The Comprehensive R...

cran.r-project.org

ggplot2

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

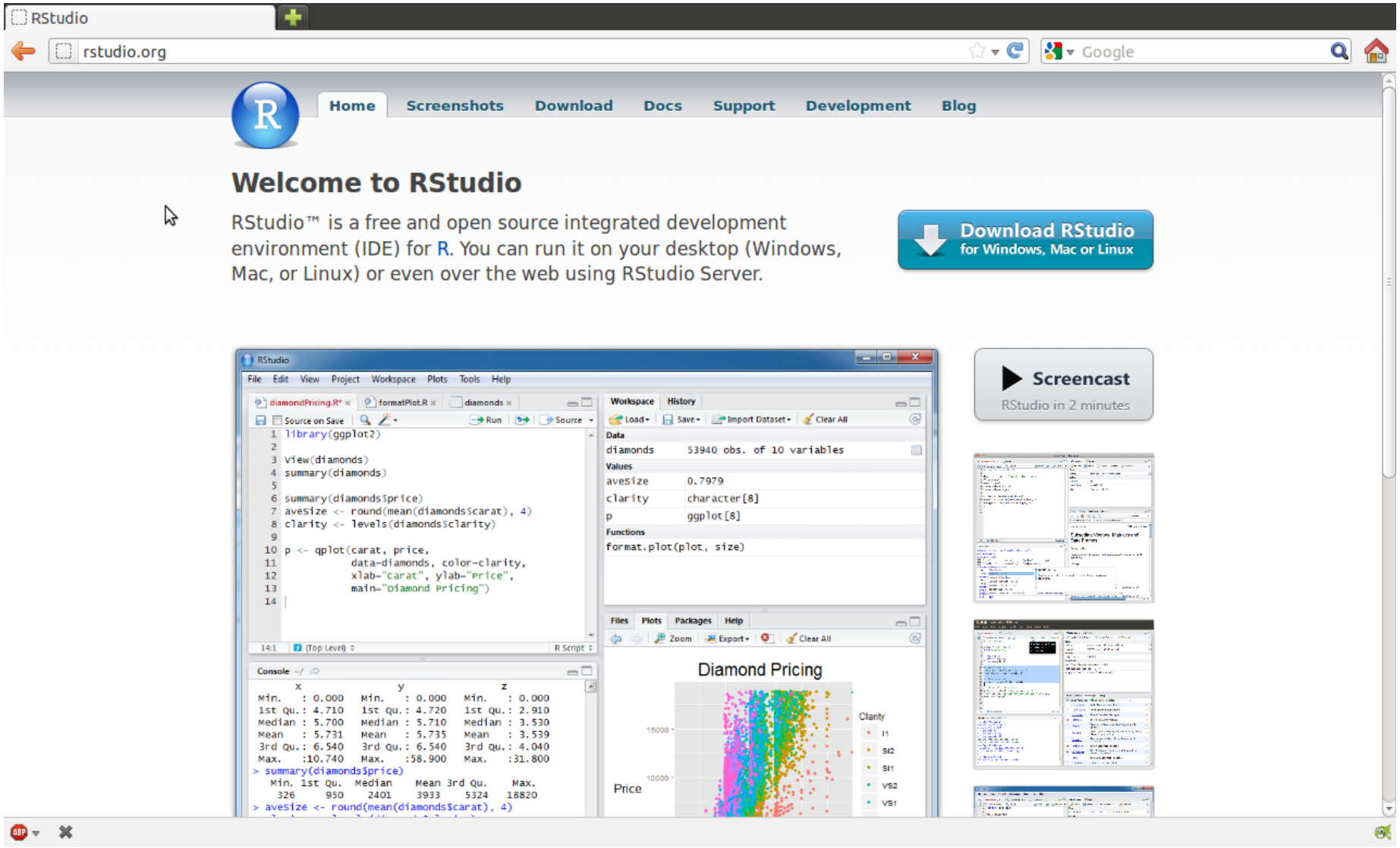
Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

CRAN Task Views

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Computational Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
Finance	Empirical Finance
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
OfficialStatistics	Official Statistics & Survey Methodology
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods
Psychometrics	Psychometric Models and Methods
ReproducibleResearch	Reproducible Research
Robust	Robust Statistical Methods
SocialSciences	Statistics for the Social Sciences
Spatial	Analysis of Spatial Data
Survival	Survival Analysis
TimeSeries	Time Series Analysis

RStudio


RStudio
rstudio.org
Google
Home
Screenshots
Download
Docs
Support
Development
Blog
Welcome to RStudio
RStudio™ is a free and open source integrated development environment (IDE) for R. You can run it on your desktop (Windows, Mac, or Linux) or even over the web using RStudio Server.
Download RStudio
for Windows, Mac or Linux
Screencast
RStudio in 2 minutes
File Edit View Project Workspace Plots Tools Help
library(ggplot2)
view(diamonds)
summary(diamonds)
summary(diamonds$price)
avesize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)
p <- qplot(carat, price,
data=diamonds, color=clarity,
xlab="Carat", ylab="Price",
main="Diamond Pricing")
Workspace History
diamonds 53940 obs. of 10 variables
avesize 0.7979
clarity character[8]
p ggplot[8]
format.plot(plot, size)
Diamond Pricing
Price
Clarity
Console

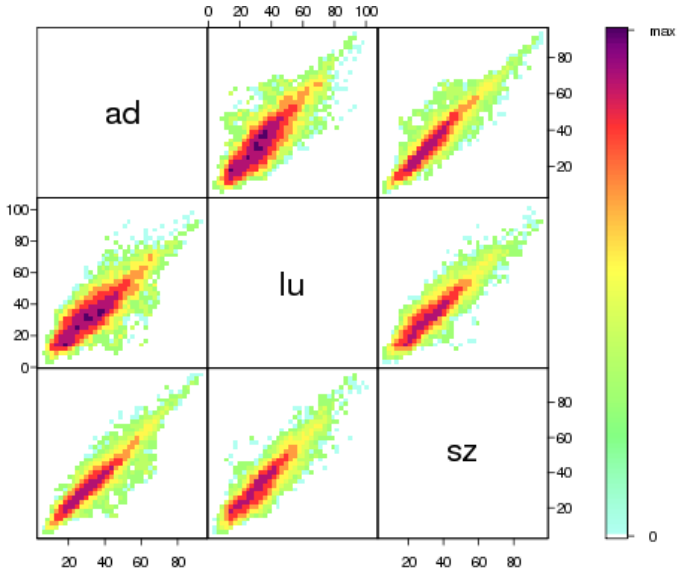
Basic features

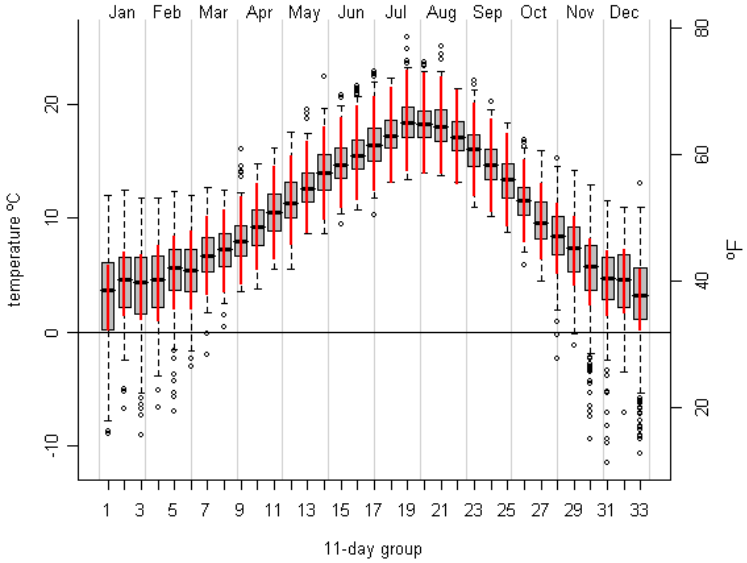
- Numerous procedures (algebra, matrix, stats)
- Named storage (everything is an object)
- Functions
- Classes and methods (S3, S4)
- Special values (NA, NaN, Inf, NULL)
- Logical objects and boolean algebra
- basic_features .R

What else can it do?

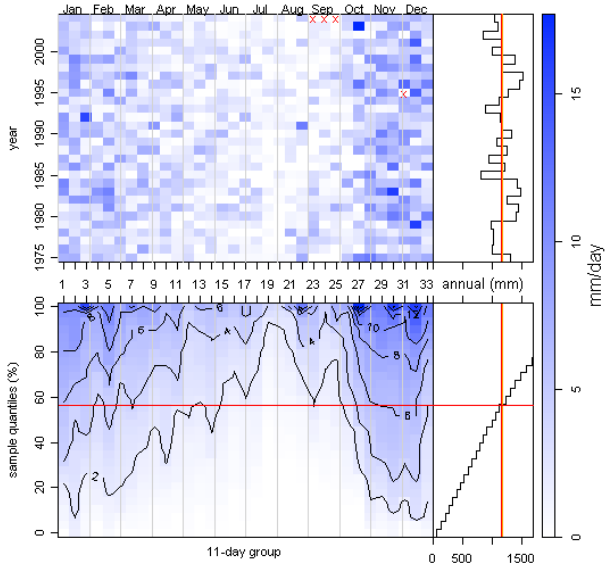
- Data handling and storage
- Matrix algebra
- Regular expressions
- Statistics!
- OOP
- Programming
- Graphics

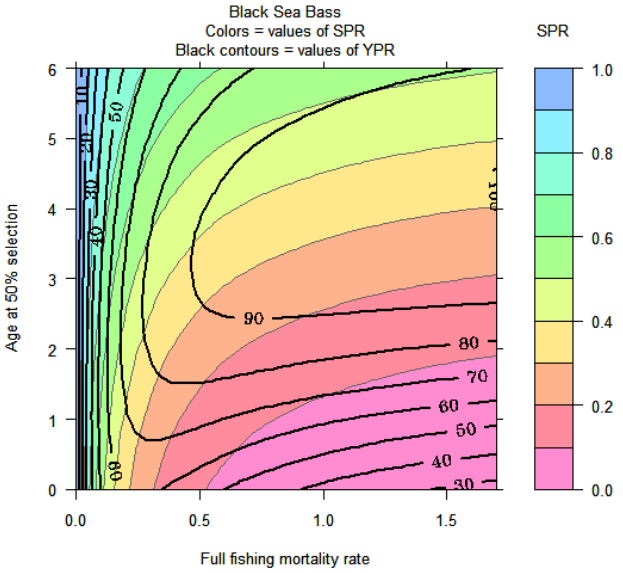
Eye candy


ad
lu
sz
max
0

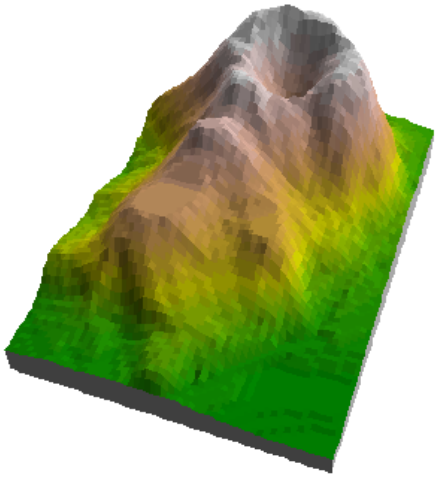

Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
temperature °C
°F
11-day group

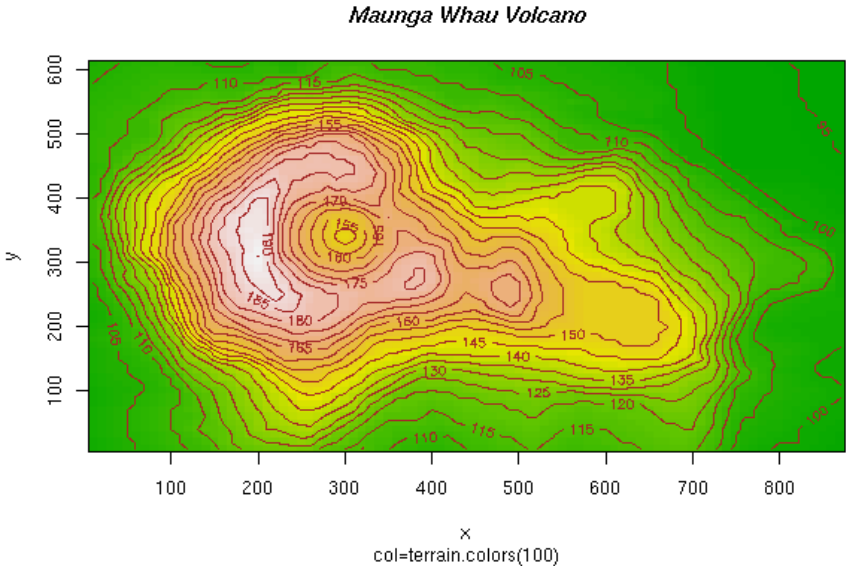
Eye candy


Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
year
1975 1980 1985 1990 1995 2000
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33
annual (mm)
sample quantiles (%)
0 20 40 60 80 100
11-day group
0 500 1500
mm/day
0 5 10 15


Black Sea Bass
Colors = values of SPR
Black contours = values of YPR
SPR
1.0 0.8 0.6 0.4 0.2 0.0
Age at 50% selection
0 1 2 3 4 5 6
Full fishing mortality rate
0.0 0.5 1.0 1.5
90 80 70 60 50 40 30

Eye candy




Maunga Whau Volcano
y
x
col=terrain.colors(100)
100
200
300
400
500
600
700
800

What doesn't it do

- No DB, but connections (SQL, NoSQL, Spreadsheets)
- No GUI, but IDE & GUI toolsets - CLI
- Slow, but C/C++, HPC
- No commerciasl support, but community
- Think for you

Help!

- Help for each function and data type
- ?mean
- ??mean
- ?help
- http://rseek.org
- stackoverflow, http://stackoverflow.com/questions/tagged/r
- Mailing lists

FLR

- Stock assessment and provision of management advice
  - Well tested, robust methods
  - Open to detailed inspection
- Data and model validation through simulation
- Risk analysis
- Capacity development & education
- Promote collaboration and openness in quantitative fisheries science
  - Open source
  - Community involvement
  - R as lingua franca
- Support the development of new models and methods
  - Extensible toolset
  - Links to other tools (ADMB, BUGS, . . .)

flr-project.org

start - FLR Project

flr-project.org/doku.php?id=start

FLR PROJECT

TABLE OF CONTENTS

Training Course

Intro to R and FLR for Fisheries

3-5 JULY 2012 – Varese, Italy

Registration now open at FishReg EC JRC

The FLR library is a collection of tools in the R statistical language that facilitates the construction of bio-economic simulation models of fisheries and ecological sytems. It is a generic toolbox, but is specifically suited for the construction of simulation models for evaluations of fisheries management strategies. The FLR library is under development by researchers across a number of laboratories and universities.

Love it !! Take me to the Table of Contents !!

You can install the **stable releases** for R >= 2.13.0 with

```
install.packages(repos="http://flr-project.org/R")
```

or you can install the **development packages** for R >= 2.14.0 with

```
install.packages(repos="http://flr-project.org/Rdevel")
```

You can get help by subscribing our mailing list and posting your doubt, or reading the **tutorials** on the "Teach yourself FLR" wiki.

Tools of the trade

- Version Control Systems
  - CVS
  - SVN
  - git
- Editors & IDEs
- Literate Programming
  - Sweave
  - knitr
- Validation, Verification and Testing (VV&T)

Sexy data analysis

Setting up R & RStudio

- http://cran.r-project.org
- http://rstudio.org