

# INTRODUCTION



# Know your Instructor



<http://linkedin.com/in/ajavohri>

- Author "[R for Business Analytics](#)"
- Author "[R for Cloud Computing](#)"
- Founder "[Decisionstats.com](#)"
- University of Tennessee, Knoxville  
MS (courses in statistics and  
computer science)
- MBA (IIM Lucknow, India-2003)
- B.Engineering (DCE 2001)

# Classroom Rules

- From Instructor
  
- From Audience
  - mobile phones should be kindly switched off
    - Yes, this includes Whatsapp
  - Ask Questions at end of session
  - Take Notes
  - Please Take Notes



# What is data science ?

Hacking ( Programming) + Maths/Statistics + Domain Knowledge = Data Science

*The Data Science Venn Diagram*



# Oh really, is this a Data Scientist ?

a data scientist is simply a person who can

write code = in R, Python, Java, SQL, Hadoop (Pig, HQL, MR) etc

= for data storage, querying, summarization, visualization

= how efficiently, and in time (fast results?)

= where on databases, on cloud, servers

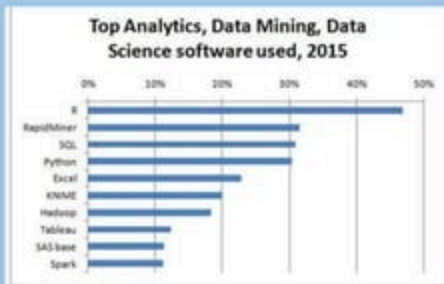
and understand enough statistics

to derive insights from data

so business can make decisions

# Data Science with R

## A popular language in Data Science



The top 10 tools by share of users were

Subscribe to KDNuggets News | Follow [@kdnuggets](#)

### R moves up to 5th place in IEEE language rankings

IEEE Spectrum has just published its third annual ranking with its [2016 Top Programming Languages](#), and the R Language is once again near the top of the list, moving up one place to 5th position.

| Language Rank | Types   | Spectrum Ranking |
|---------------|---------|------------------|
| 1. C          | 📱 🖨️ 📄  | 100.0            |
| 2. Java       | ☺️ 📱 🖨️ | 98.1             |
| 3. Python     | ☺️ 🖨️   | 98.0             |
| 4. C++        | 📱 🖨️ 📄  | 95.8             |
| 5. R          | 🖨️      | 87.8             |
| 6. C#         | ☺️ 📱 🖨️ | 86.7             |
| 7. PHP        | ☺️      | 82.8             |
| 8. JavaScript | ☺️ 📱    | 82.2             |
| 9. Ruby       | ☺️ 🖨️   | 74.5             |
| 10. Go        | ☺️ 🖨️   | 71.8             |

As I said [last year](#) (when R moved up to take sixth place), this is an extraordinary result for a domain-specific language. The other four languages in the top 5 (C, Java, Python and C++) are all general-purpose languages, suitable for just about any programming task. R by contrast is a language specifically for data science, and its high ranking here reflects both the critical importance of data science as a discipline today, and of R as the language of choice for [data scientists](#).

# What Is R

<https://www.r-project.org/about.html>

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

# Install R

<https://cran.r-project.org/bin/windows/base/>

R-3.3.1 for Windows (32/64 bit)

[Download R 3.3.1 for Windows](#) (70 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded exactly matches the package distributed by R, you can compare the [md5sum](#) of the .exe to the [true fingerprint](#). You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

[Frequently asked questions](#)

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

[Other builds](#)

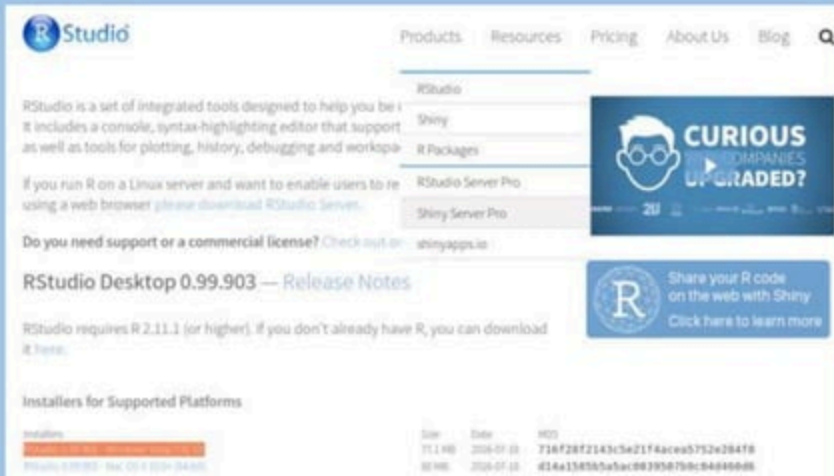
- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is <https://CRAN.MIRROR>/bin/windows/base/release.htm>.



# Install RStudio

<https://www.rstudio.com/products/rstudio/download/>



The screenshot shows the RStudio website's product page. At the top, there is a navigation bar with links for Products, Resources, Pricing, About Us, and Blog, along with a search icon. The main content area features the RStudio logo and a description of the software as a set of integrated tools for R. Below this, there is a list of products: RStudio, Shiny, R Packages, RStudio Server Pro, and Shiny Server Pro. A sidebar on the right contains two promotional banners: one for 'CURIOUS COMPANIES UPGRADED?' and another for 'Share your R code on the web with Shiny'. The main content area also includes a section for 'RStudio Desktop 0.99.903 -- Release Notes' and a table of installers for supported platforms.

**RStudio**

Products Resources Pricing About Us Blog

RStudio is a set of integrated tools designed to help you be productive. It includes a console, syntax-highlighting editor that supports many languages and targets many systems, as well as tools for plotting, history, debugging and workspace saving. Cross-referenced documentation and code snippets make the workflow more efficient.

If you run R on a Linux server and want to enable users to use R using a web browser, please download RStudio Server.

Do you need support or a commercial license? Check out our [Shinyapps.io](#) or [RStudio Pro](#) options.

**RStudio Desktop 0.99.903 -- Release Notes**

RStudio requires R 2.11.1 (or higher). If you don't already have R, you can download it [here](#).

**Installers for Supported Platforms**

| Platform         | Size    | Date       | MD5                              |
|------------------|---------|------------|----------------------------------|
| Windows (64-bit) | 71.1 MB | 2016-07-18 | 716f28f2143c3e21f4cc55752e284f8  |
| Windows (32-bit) | 58.9 MB | 2016-07-18 | 414a158365a5ac083958750c94846606 |

# Statistical Software Landscape

SAS

Python (Pandas)

IBM SPSS

R

Julia

Clojure

Octave

Matlab

JMP

E views



# Using R with other software

<https://rforanalytics.wordpress.com/useful-links-for-r/using-r-from-other-software/>

Tableau <http://www.tableausoftware.com/new-features/r-integration>

Qlik <http://qliksolutions.ru/qlikview/add-ons/r-connector-eng/>

Oracle R <http://www.oracle.com/technetwork/database/database-technologies/r/r-enterprise/overview/index.html>

Rapid Miner <https://rapid-i.com/content/view/202/206/lang,en/#r>

JMP <http://blogs.sas.com/jmp/index.php?/archives/298-JMP-Into-R!.html>



# Using R with other software

<https://rforanalytics.wordpress.com/useful-links-for-r/using-r-from-other-software/>

SAS/IML <http://www.sas.com/technologies/analytics/statistics/iml/index.html>

Teradata <http://developer.teradata.com/applications/articles/in-database-analytics-with-ter>

Pentaho <http://bigdatatechworld.blogspot.in/2013/10/integration-of-rweka-with-pentaho-data.html>

IBM SPSS [https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=ibm-analytics&S\\_PKG=ov18855&S\\_TACT=M161003W&dynform=127&lang=en\\_US](https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=ibm-analytics&S_PKG=ov18855&S_TACT=M161003W&dynform=127&lang=en_US)

TIBCO TERR <http://spotfire.tibco.com/discover-spotfire/what-does-spotfire-do/predictive-analytics/tibco-enterprise-runtime-for-r-terr>



# Some Advantages of R

open source

free

large number of algorithms and packages esp for statistics

flexible

very good for data visualization

superb community

rapidly growing

can be used with other software



# Some Disadvantages of R

in memory (RAM) usage

steep learning curve

some IT departments frown on open source

verbose documentation

tech support

evolving ecosystem for corporates



# Solutions for Disadvantages of R

- in memory (RAM) usage → specialized packages, in database computing
- steep learning curve → TRAINING !!!
- some IT departments frown on open source → TRAINING and education!
- verbose documentation → CRAN View , R Documentation
- tech support → expanding pool of resources
- evolving ecosystem for corporates → getting better with MS et al

# R used by Government

- In the early days of the [Deepwater Horizon disaster](#), NIST used uncertainty analysis in R to harmonize spill estimates from various sources, and to provide ranges of estimates to other agencies and the media.
- Before new drugs are allowed on the market, the FDA works with pharmaceutical companies to verify safety and efficacy through clinical trials. Despite a [false perception](#) that only commercial software may be used, many pharmaceutical companies are now using open-source R to [analyze data from clinical trials](#).
- The National Weather Service uses R for research and development of [models to predict river flooding](#).
- The newly-formed [Consumer Financial Protection Bureau](#) -- freed from the restrictions of a legacy IT infrastructure -- is championing the use of open-source technologies in government.
- Local governments are also building data-based applications. The SF Estuary Institute [uses R and Google Maps](#) to provide a [tool to track pollution](#) in the San Francisco Bay area.

[http://gsnmagazine.com/node/26483?c=cyber\\_security](http://gsnmagazine.com/node/26483?c=cyber_security)



# R used by Telecom

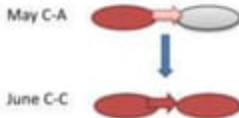
- Churn using

## Social Network Analysis

<http://www.slideshare.net/dataspora/social-network-a>

**Results: A Customer With a Canceller in Their Network  
Churns at Twice the Rate**

Types of Connections (Edges)



| reality | expected<br>by chance | delta |
|---------|-----------------------|-------|
| X       | Y                     | 2.0   |

In essence, we are asking whether being connected to another canceller has any effect on one's rate of cancellation. It turns out that it does.

And if we only look at voluntary port-outs, we see that customers churn at 3x the rate.

# R used by Insurance

a few more insurance related packages:

- [ChainLadder](#) – Reserving methods in R. The package provides Mack-, Munich-, Bootstrap, and Multivariate-chain-ladder methods, as well as the LDF Curve Fitting methods of Dave Clark and GLM-based reserving models.
- [cplm](#) – Monte Carlo EM algorithms and Bayesian methods for fitting Tweedie compound Poisson linear models
- [lossDev](#) – A Bayesian time series loss development model. Features include skewed-t distribution with time-varying scale parameter, Reversible Jump MCMC for determining the functional form of the consumption path, and a structural break in this path; by Christopher W. Laws and Frank A. Schmid
- [actuar](#): Loss distributions modelling, risk theory (including ruin theory), simulation of compound hierarchical models and credibility theory check out the [actuar](#) package by C. Dutang, V. Goulet and M. Pigeon.
- [favir](#): Formatted Actuarial Vignettes in R. FAVIR lowers the learning curve of the R environment. It is a series of peer-reviewed Sweave papers that use a consistent style.
- [mondate](#): R packackge to keep track of dates in terms of months
- [lifecontingencies](#) – Package to perform actuarial evaluation of life contingencies

and

[Introduction to R for Actuaries](#) by Nigel de Silva

and <http://www.rininsurance.com/>

# R in Finance

<http://www.rinfinance.com/>

R/Finance

[home](#)

[agenda](#)

[register](#)

[travel](#)

[committee](#)

## Friday, May 29th, 2015

08:00 - 09:00 Optional Pre-Conference Tutorials

Ross Bennett: PortfolioAnalytics: Advanced Moment Estimation & Optimization ([pdf](#))

Kris Boudt: High-frequency Price Data Analysis in R ([pdf](#))

Dirk Eddelbuettel: Hands-on Introduction to Rcpp ([pdf](#))

Guy Yoffie: Getting Started with Quantstrat

María Bellanina: An Introduction to OneTick

09:00 - 09:30 Registration (2nd floor Inner Circle) & Continental Breakfast (3rd floor by Sponsor Tables)

Transition between seminars

09:30 - 09:35 Kickoff

09:35 - 09:40 Sponsor Introduction

09:40 - 10:30 **Emanuel Derman**: Understanding the World

10:30 - 10:54 **John Burkett**: Portfolio Optimization: Price Predictability, Utility Functions, Computational Methods, and Applications ([pdf](#))

**Kyle Balkinsson**: A Framework for Integrating Portfolio-level Backtesting with Price and Quantity Information ([html](#))

**Anthony Tsou**: Implementation of Quality Minus Junk

**Ilya Kipnis**: Flexible Asset Allocation With Stepwise Correlation Rank ([pptx](#))

10:54 - 11:20 Break

11:20 - 11:40 **Sanjiv Das**: Efficient Rebalancing of Taxable Portfolios ([pdf](#))

11:40 - 12:00 **Marjan Wauters**: Characteristic-based equity portfolios: economic value and dynamic style allocation ([pdf](#))

12:00 - 12:20 **Bernhard Pfaff**: The sequel of cccp: Solving cone constrained convex programs

12:20 - 13:40 Lunch

13:40 - 14:00 **Markus Gesmann**: Communicating risk - a perspective from an insurer ([pdf](#))

14:00 - 14:20 **Doug Martin**: Nonparametric vs Parametric Shortfall: What are the Differences?

# R in Finance

<http://cran.r-project.org/web/views/Finance.html>

This CRAN Task View contains a list of packages useful for empirical work in Finance, grouped by topic.

- The Rmetrics suite of packages comprises [fArma](#), [fAsianOptions](#), [fAssets](#), [fBasics](#), [fBonds](#), [timeDate](#) (formerly: fCalendar), [fCopulae](#), [fExoticOptions](#), [fExtremes](#), [fGarch](#), [fImport](#), [fNonlinear](#), [fOptions](#), [fPortfolio](#), [fRegression](#), [timeSeries](#) (formerly: fSeries), [fTrading](#), [fUnitRoots](#) and contains a very large number of relevant functions for different aspect of empirical and computational finance.
- The [RQuantLib](#) package provides several option-pricing functions as well as some fixed-income functionality from the QuantLib project to R.
- The [quantmod](#) package offers a number of functions for quantitative modelling in finance as well as data acquisition, plotting and other utilities.
- The [portfolio](#) package contains classes for equity portfolio management; the [portfolioSim](#) builds a related simulation framework. The [backtest](#) offers tools to explore portfolio-based hypotheses about financial instruments. The [stockPortfolio](#) package provides functions for single index, constant correlation and multigroup models. The [pa](#) package offers performance attribution functionality for equity portfolios.
- The [PerformanceAnalytics](#) package contains a large number of functions for portfolio performance calculations and risk management.

# R in Pharma

<http://blog.revolutionanalytics.com/2013/08/r-drug-development-and-the-fda.html>

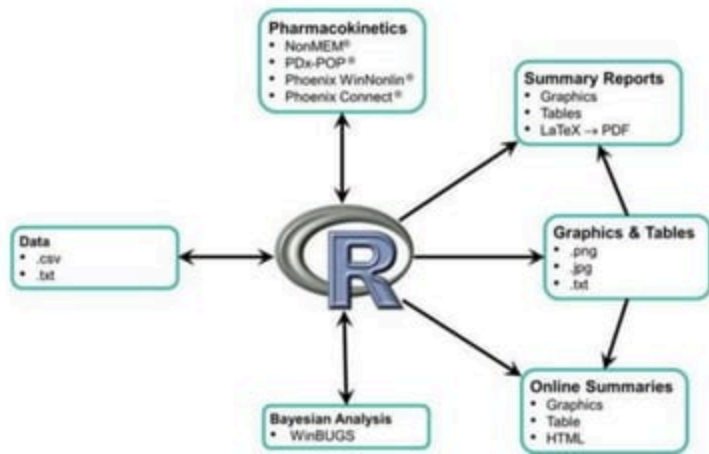
*[Opening the Doors to Open Source Programming in Drug Development.](#)*

*[R: Regulatory Compliance and Validation Issues A Guidance Document for the Use of R in Regulated Clinical Trial Environments](#)* in which he concluded that useR 2012 FDA statistician Jea Brodsky presented a [poster](#) described how FDA scientists "use R on a daily basis" and have themselves written R packages for use at various stages in the drug submission process.

*[Open Source Software in the Biopharma Industry: Challenges and Opportunities.](#)*

# R in Pharma

<http://web.quanticate.com/bid/102741/Using-the-Statistical-Programming-Language-R-in-the-Pharma-Industry>



# R in Pharma

<http://cran.r-project.org/web/views/ClinicalTrials.html>

This task view gathers information on specific R packages for design, monitoring and analysis of data from clinical trials. It focuses on including packages for clinical trial design and monitoring in general plus data analysis packages for a specific type of design.

## Design and Monitoring

- adonis2** This package has over 30 functions from the book Design for Cost-Effective Clinical Research (Dunn & Wang & Shaw, 2017, 2nd ed., Chapman & Hall/CRC).
- ad** This package was specifically designed to provide a simple design tool for researchers in the pharmaceutical industry.
- ada** This package implements a wide variety of one and two parameter Bayesian CDE designs. The program can be used to generate designs for each subject for drug treatment, or for comparing to control (placebo).
- adonis2** creates randomization for block randomization trials. It can also produce a PDF file of randomization plans.
- adonis2** This package contains a series of sample tools for constructing and comparing randomized and factorial treatment designs.
- AD2Sim** This package contains tools for the generation of sample size calculations for various group sequential trials. The package contains traditional group-based methods, sequential monitoring (Bretz and Hothorn, 2003), and adaptive sample size calculations (Bretz and Hothorn, 2003).
- adonis** This package provides functions to use the CDE and TDE (or other) trials and calibration tools for trial planning purposes.
- adonis2** contains tools for clinical experiments, e.g., a randomization tool, and a generator for a sequential randomization system for clinical trials.
- AD2** This package creates random and non-random Treatment-Factorial designs. Randomized adaptive trials for Treatment-Factorial designs with 2-level factors are allowed (cross-over, within and between-patient) for all factors sequentially, with plan for looking at the main-effects of both factors, full or half factorial plans, also designs as a special case (factorial design) for both or for factors alone. The package is currently under active development. While work of the standard factorization is already available, some changes and improvements are still to be expected.
- AD2Sim** provides comprehensive support for group sequential designs via the alpha spending approach, i.e., address within-group and/or across-group, and then enables user to be specified as within.
- ad2Design** defines group sequential design and describes their properties.
- adonis2** This package implements and plans group sequential designs from the Lan, DeLamater method with a variety of spending functions using the alpha program from the Department of Biostatistics, University of Wisconsin (written by David Finkelstein, IS, DeLamater, 2005, Lan, 2003, Lan, 2003, Lan).
- AD2Sim** uses Lan-DeLamater Method for group sequential trials. In fact, the package contains methods and publications of a group sequential trial.
- AD2Sim** This package provides functions for both the sequential design and sample size for (one or more) of (sequential) designs (one or more) as described in Lan and Lan (2007) and Hothorn et al (2007). Other functions are provided to assess the impact of (sequential) designs (one or more) as described in Lan and Lan (2007) parameters. The package contains functions which transfer point estimate effect size (parameter) to a random sample size in design, and sequential design parameters in the function of design in Lan and Lan (2007) can be applied in practice sample size calculations for two equally important designs, including binary response.
- AD2** provides graphical user interface, simulation and plan, confidence intervals of an effect measure, group effect of data and a hypothesis about the distribution of these data.
- AD2Sim** provides functions to calculate group and sample size for various study designs used for sequential designs. See function AD2Sim() for the study design control. Moreover, the package contains functions for group and sample size based on sequential design. See function AD2Sim() for the study design control. Moreover, the package contains functions for group and sample size based on sequential design. See function AD2Sim() for the study design control.
- ada** has group randomization functions for the use of TDEs (2005).
- AD2Sim** is a set of tools to sample points in a group sequential design.
- AD2Sim** provides tools for the design of TDEs experiments.
- AD2Sim** implements the probability of meeting sequential efficacy and futility boundaries in a clinical trial. Implemented for Adaptive 2/1, Pocock and Simon algorithms using the method described in Finkelstein (2007).

## Design and Analysis

- AD2Sim** This package provides tools and functions for parameter estimation in adaptive group sequential trials.
- AD2Sim** provides functions for both design and analysis of clinical trials. The design 2 trials, it has functions to estimate sample size, effect size, and power based on Fisher's exact test. The spending distributions of a two-stage boundary, optimal and Hothorn, Lan, 2003 design points for Fisher's exact test, the two-stage design and one-sample stopping rule and its spending distributions for Monte Carlo simulation based on sequential significance testing. For design 2 trials, it has calibration sample size for group sequential design.

# Companies using R

from <http://www.revolutionanalytics.com/companies-using-r>

ANZ, the fourth largest bank in Australia, using R for credit risk analysis

Bank of America uses R for reporting.

The Consumer Financial Protection Bureau uses R for data analysis.

Facebook

Facebook and R:

- Analysis of Facebook Status Updates
- Facebook's Social Network Graph
- How Google and Facebook are using R
- Predicting Colleague Interactions with R



# Refresher in Statistics

## Mean

Arithmetic Mean- the sum of the values divided by the number of values.

The [geometric mean](#) is an average that is useful for sets of positive numbers that are interpreted according to their product and not their sum (as is the case with the arithmetic mean) e.g. rates of growth.

## Median

the median is the number separating the higher half of a data sample, a population, or a probability distribution, from the lower half

## Mode-

The "mode" is the value that occurs most often.

# Refresher in Statistics

## Range

the **range** of a set of data is the difference between the largest and smallest values.

## Variance

mean of squares of differences of values from mean

## Standard Deviation

square root of its variance

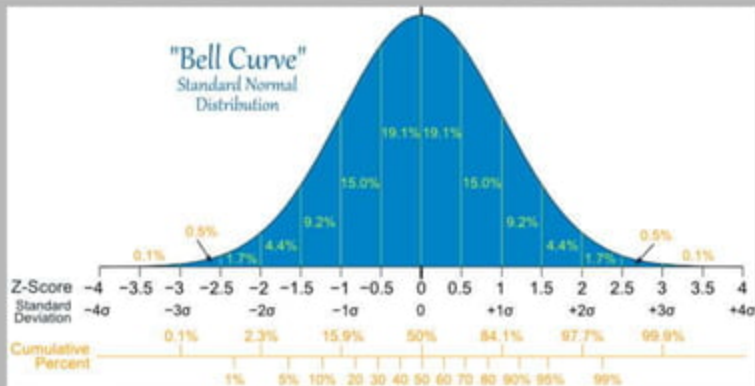
## Frequency

a **frequency distribution** is a table that displays the **frequency** of various outcomes in a sample.

# Distributions

## Normal

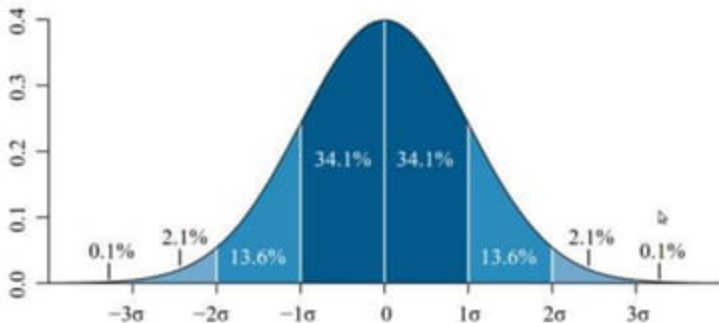
The simplest case of a normal distribution is known as the *standard normal distribution*. This is a special case where  $\mu=0$  and  $\sigma=1$ .



# Refresher in Statistics

## Probability Distribution

The [probability density function](#) (pdf) of the [normal distribution](#), also called Gaussian or "bell curve", the most important continuous random distribution. As notated on the figure, the probabilities of intervals of values correspond to the area under the curve.



# Pre Requisites

- Installation of R

<http://cran.rstudio.com/bin/windows/base/>



Home  
About  
Downloads  
Links  
Help  
Contact  
Privacy  
Terms  
Sponsors  
Partners  
Feedback

- R Studio

- R Packages

R 4.1.1 for Windows (2024-04-11)

Download R 4.1.1 for Windows (64-bit) installer

Download R 4.1.1 for Windows (32-bit) installer

Download R 4.1.1 for Windows (ARM64) installer

It is recommended to install R on a 64-bit system. The 32-bit installer is only available for systems with 32-bit processors. The 64-bit installer is available for systems with 64-bit processors. The ARM64 installer is available for systems with ARM64 processors.

Pre-requisites for Windows

- Microsoft Windows 10 or later
- Microsoft Windows 11 or later
- Microsoft Windows 12 or later

When using R 4.1.1 for Windows (64-bit) or Windows (ARM64) installers, you also need:

Other tools

- Python 3.8 or later (see [Python for Windows](#))
- A C++ compiler (see [C++ for Windows](#))
- A C++ standard library (see [C++ for Windows](#))

For more information, see [R for Windows](#) or [R for Windows](#).

For more information, see [R for Windows](#).



# Pre Requisites

- Installation of R
  - RTools

- R Studio

<http://www.rstudio.com/products/rstudio/download/>

- R Packages



The screenshot shows the RStudio website's download page. At the top, there is a navigation bar with links for Products, Resources, Pricing, About Us, and Blog. Below this, the main heading is "Download RStudio".

There are two main sections for downloading:

- Download RStudio Desktop v0.99.1074**: This section includes a description of RStudio as an IDE for R, a list of features (code editor, debugger, console, etc.), and a "Download" button. A sidebar on the right asks "Do you need support or a commercial license?" with a "Contact Us" button.
- Download RStudio Server**: This section includes a description of RStudio Server and a "Download" button.

At the bottom, there is a table titled "Installers for Mac Platforms" with columns for "Platform", "Date", and "Download".

| Platform          | Date       | Download  |
|-------------------|------------|---|
| Mac OS X (64-bit) | 2014-09-10 | <a href="#">Download RStudio Desktop (64-bit)</a> |
| Mac OS X (32-bit) | 2014-09-10 | <a href="#">Download RStudio Desktop (32-bit)</a> |
| Mac OS X (64-bit) | 2014-09-10 | <a href="#">Download RStudio Server (64-bit)</a>  |
| Mac OS X (32-bit) | 2014-09-10 | <a href="#">Download RStudio Server (32-bit)</a>  |

# Pre Requisites

- Installation of R
  - RTools

- R Studio

<http://www.rstudio.com/products/rstudio/download/>



- R Packages about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.



# CRAN

107 sites in 49 regions



CRAN  
Mirror  
What's new?  
Task Views  
Search

About R  
R Homepage  
The R Journal

Software  
R Sources  
R Datasets  
Packages  
Other

Documentation  
Manuals  
FAQs  
Contributors

## CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: [main page](#), [windows release](#), [windows old release](#)

### ICloud

<http://cran.icscloud.com/>

### Algeria

<http://cran.univ-bz.dz/>

### Argentina

<http://mirror.fing.edu.uy/pub/CRAN/>

### Australia

<http://cran.usits.ac/>

<http://cran.us.unimelb.edu.au/>

### Austria

<http://cran.at.z-project.org/>

### Belgium

<http://www.fisimatistica.org/cran/>

### Brazil

<http://fbcg.igb.usp.br/mirror/cran/>

<http://cran.z.cefal.ufpe.br/>

<http://cran.fisimat.br/>

<http://www.fgq.br/cran.org.br/CRAN/>

<http://bragor.cofa.org.br/CRAN/>

### Canada

<http://cran.stat.ubc.ca/>

<http://mirror.its.dal.ca/cran/>

<http://cran.utoronto.ca/>

<http://cran.ackabreyon.com/>

<http://cran.parcingamerica.com/>

### Chile

<http://dsv.chil.usp.puc.cl/>

### China

<http://ftp.cnc.org/mirror/CRAN/>

<http://mirror.fjhu.edu.cn/cran/>

<http://mirror.opm.com.cn/cran/>

Rstudio, automatic redirection to servers worldwide

University of Science and Technology Housti Dostanovici

Universidad Nacional de La Plata

CSIRO

University of Melbourne

Wirtschaftsuniversität Wien

K.U. Leuven Association

Center for Comp. Biol. at Universidade Estadual de Santa Cruz

Universidade Federal do Paraná

Oswaldo Cruz Foundation, Rio de Janeiro

University of Sao Paulo, Sao Paulo

University of Sao Paulo, Piracicaba

Simon Fraser University, Burnaby

Dalhousie University, Halifax

University of Toronto

(Web, Montreal)

(Web, Montreal)

Pontificia Universidad Católica de Chile, Santiago

CTEX.ORG

Beijing Jiaotong University, Beijing

Chinese Academy of Sciences, Beijing

# Non CRAN Repositories

<http://www.rdocumentation.org/>



The screenshot shows the RDocumentation website interface. On the left is a navigation menu with categories like Biopack, CloudPkg, and others. The main content area features a search bar with the text "Search the R documentation of 7382 R packages and 85000 R functions." Below this is a descriptive paragraph and a search form with fields for "All Fields", "Package Name", "Function Name", "Title", and "Description". A green "Start search" button is at the bottom of the form. On the right side, there is a DarkCam advertisement for a \$25/month service and a footer section listing "Aggregating packages from" CRAN, Bioconductor, and GitHub.

# github

<https://github.com/trending?l=R>

The screenshot shows the GitHub website's trending repositories page. At the top, there is a navigation bar with the GitHub logo, a search bar, and links for Explore, Star, Blog, and Help. Below the navigation bar, the page title is "Explore GitHub" with tabs for All, Overview, Trending, and Stars. The main heading is "Trending repositories" with a sub-heading "Find what repositories the GitHub community is most excited about today." Below this, there are filters for "Repositories" (all, Developers) and "Trending" (today, weekly, monthly). A sidebar on the right shows "All languages" with a list of programming languages, where "R" is selected and highlighted in blue. Below the sidebar, there is a list of trending repositories for R. Each repository entry includes the repository name, a brief description, and a "Star" button.

Repositories: [all](#) [Developers](#) Trending: [today](#) [weekly](#) [monthly](#)

**rtong/ProgrammingAssignment2** [Star](#)  
Repository for Programming Assignment 2 for R Programming in Courses  
It is built by

**gnaf/awesome-R** [Star](#)  
A curated list of awesome R frameworks, packages and software.  
It is built by

**bendroschi/mlr** [Star](#)  
mlr: Machine Learning in R  
It is built by

**rstudio/shinyapps** [Star](#)

All languages  
Common languages  
C  
C++  
C#  
Go  
Java  
JavaScript  
Python  
[View all languages](#)

Perfect! Looking for more GitHub R repositories? Try this search.

# bioconductor

<http://www.bioconductor.org/>

The screenshot shows the Bioconductor website homepage. At the top left is the Bioconductor logo with the tagline "OPEN SOURCE SOFTWARE FOR BIOINFORMATICS". To the right of the logo is a search bar and a navigation menu with links for Home, Contact, Help, Developers, and About. The main content area is divided into several sections:

- BioC2015**: A section announcing the Bioconductor 2015 conference, held in Seattle, WA, from October 21-23. It invites users to join for morning talks, workshops, and networking.
- About Bioconductor**: A section describing Bioconductor as an R ecosystem for high-throughput genomic data analysis and development. It mentions that it has two releases per year and is supported by a community.
- News**: A section with a list of recent updates, including the availability of Bioconductor 2.2, the release of the high-throughput genomic analysis package `HTSeq`, and the release of the `limma` package.
- Install**: A section titled "Get started with Bioconductor" that provides a list of links for installing Bioconductor on various operating systems (Linux, Windows, Mac OS) and for installing the `BiocManager` package.
- Learn**: A section titled "Master Bioconductor tools" that provides a list of links for learning about Bioconductor, including courses, tutorials, and documentation.
- Use**: A section titled "Create bioinformatic solutions with Bioconductor" that provides a list of links for using Bioconductor, including software, tutorials, and documentation.
- Develop**: A section titled "Contribute to Bioconductor" that provides a list of links for contributing to Bioconductor, including the `Bioc-devel` mailing list, the `Bioc-devel` package, and the `Bioc-devel` website.

# Install R

<https://cran.r-project.org/bin/windows/base/>

R-3.3.1 for Windows (32/64 bit)

[Download R 3.3.1 for Windows](#) (70 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

If you want to double-check that the package you have downloaded exactly matches the package distributed by R, you can compare the [md5sum](#) of the .exe to the [true fingerprint](#). You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

[Frequently asked questions](#)

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

[Other builds](#)

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is <https://CRAN.MIRROR>/bin/windows/base/release.htm>.

# Install RStudio

<https://www.rstudio.com/products/rstudio/download/>

The screenshot shows the RStudio website's product page. At the top, there is a navigation bar with links for Products, Resources, Pricing, About Us, and Blog, along with a search icon. The main content area features the RStudio logo and a description of the software as a set of integrated tools for R. A dropdown menu is open, listing various products: RStudio, Shiny, R Packages, RStudio Server Pro, Shiny Server Pro, and shinyapps.io. Below this, there is a section for 'RStudio Desktop 0.99.903 -- Release Notes' and a table of installers for supported platforms. On the right side, there are two promotional banners: one for 'CURIOUS COMPANIES UPGRADED?' and another for 'Share your R code on the web with Shiny'.

**RStudio**

Products Resources Pricing About Us Blog

RStudio is a set of integrated tools designed to help you be productive. It includes a console, syntax-highlighting editor that supports all major languages, as well as tools for plotting, history, debugging and workspace saving. Cross platform synchronization is available for the desktop client. If you run R on a Linux server and want to enable users to use your R environment from a web browser, please download RStudio Server.

Do you need support or a commercial license? Check out our [support](#) or [commercial licenses](#).

**RStudio Desktop 0.99.903 -- Release Notes**

RStudio requires R 2.11.1 (or higher). If you don't already have R, you can download it [here](#).

**Installers for Supported Platforms**

| Platform       | Size    | Date       | MD5                             |
|----------------|---------|------------|---------------------------------|
| Windows 32-bit | 71.1 MB | 2016-07-18 | 716f28f2143c3e21f4cc55752e284f8 |
| Windows 64-bit | 82 MB   | 2016-07-18 | 414a158365a5ac083958750c8484666 |

**CURIOUS COMPANIES UPGRADED?**

Share your R code on the web with Shiny. [Click here to learn more](#)

# Pre Requisites

- R Packages

`install.packages()` INSTALLS

`update.packages()` UPDATES

`library()` LOADS

- Packages are **installed** once, updated periodically, but **loaded** every time

# Interfaces to R

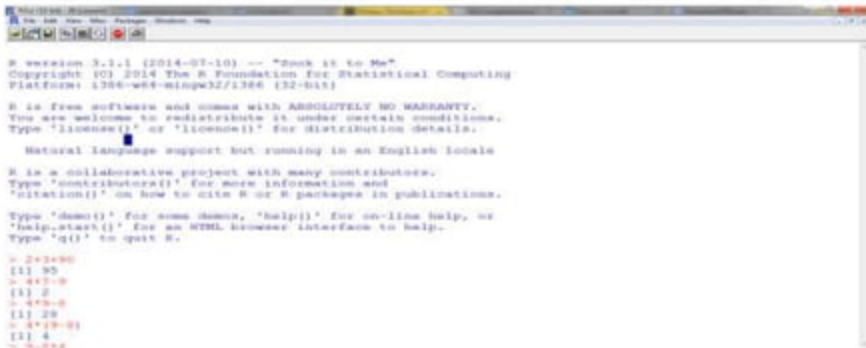
- Console

*Default*

*Customization*

- IDE

- GUI



```
R version 3.1.1 (2014-07-10) -- "Rock to the Max"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/x32 (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.

  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.

> 2+3*40
[1] 95
> 4+7-9
[1] 2
> 4*5-8
[1] 12
> 4*(2-3)
[1] 4
~ ~ ~
```

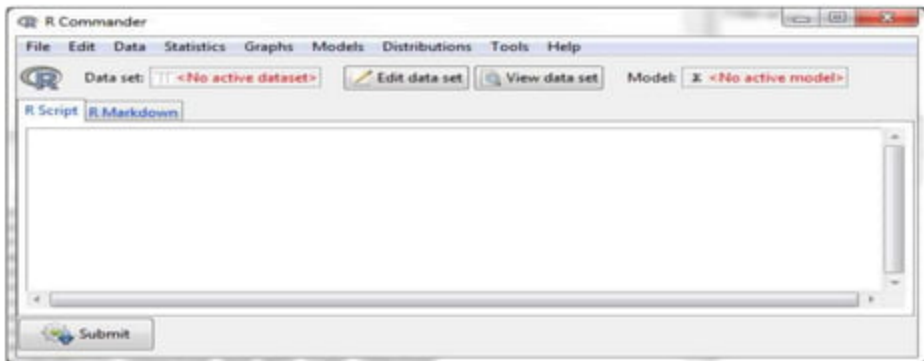


# Graphical Interfaces to R

- R Commander
- Rattle
- Deducer

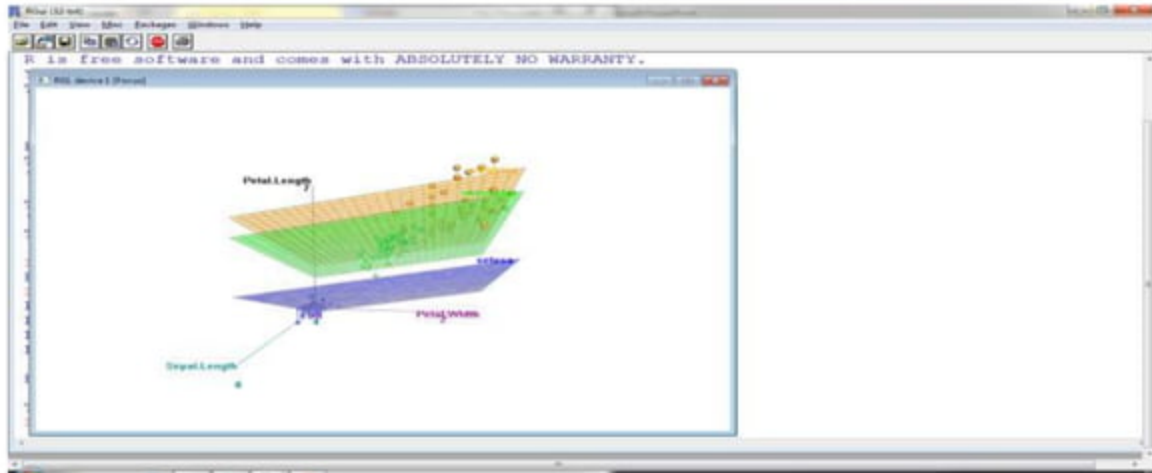


# Overview of R Commander

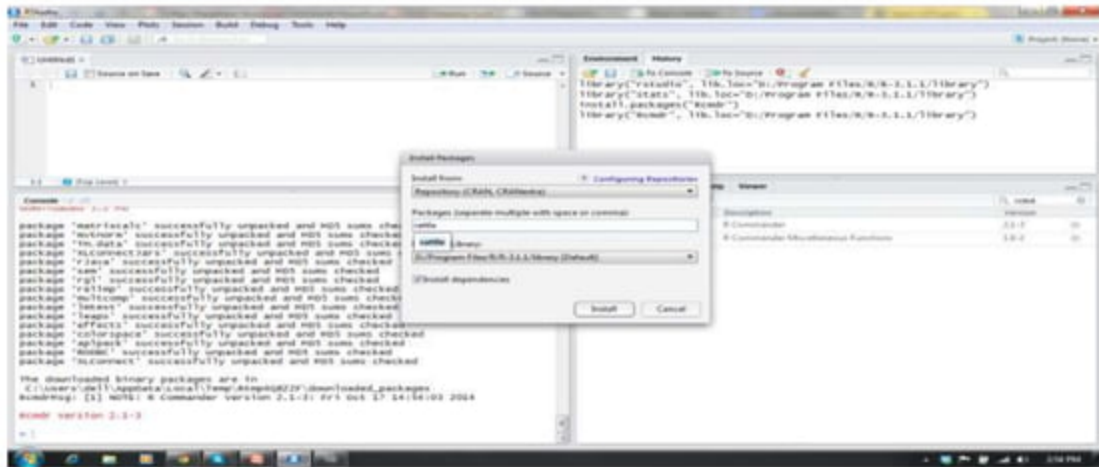


# Demo

## R Commander – 3D Graphs

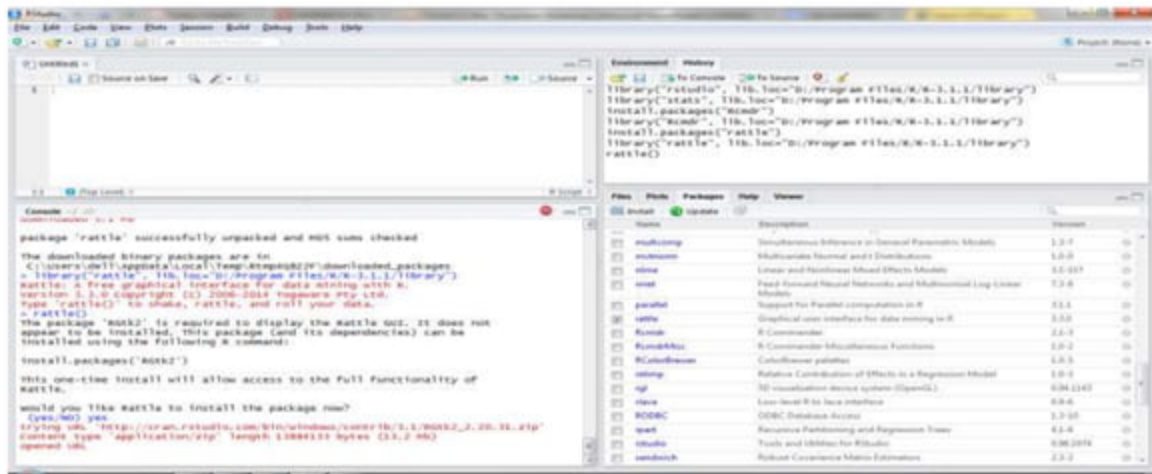


# Installation of Rattle





# Installation of Rattle



The screenshot shows the RStudio interface with the following components:

- Environment: History** pane (top right): Shows the installation of several packages:

```
library("rstatfx", lib.loc="C:/Program Files/R/R-3.1.1/library")
library("stats", lib.loc="C:/Program Files/R/R-3.1.1/library")
install.packages("R60k2")
library("R60k2", lib.loc="C:/Program Files/R/R-3.1.1/library")
install.packages("rattle")
library("rattle", lib.loc="C:/Program Files/R/R-3.1.1/library")
rattle()
```
- Files, Plots, Packages, Help, View** pane (bottom right): Shows a list of installed and available packages with their descriptions and versions.
- Console** (bottom): Shows the execution of R commands and their output:

```
package 'rattle' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\users\de11\appdata\local\temp\Rtmp0022\downloaded_packages
> library("rattle", lib.loc="C:/Program Files/R/R-3.1.1/library")
rattle: a free graphical interface for data mining with R,
version 3.3.0 Copyright (C) 2006-2014 Rogers Pty Ltd,
type 'rattle()' to show, rattle, and roll your data.
> rattle()
The package 'R60k2' is required to display the rattle GUI. It does not
appear to be installed. This package (and its dependencies) can be
installed using the following R command:

install.packages("R60k2")

This one-time install will allow access to the full functionality of
rattle.

would you like rattle to install the package now?
[yes/no] yes
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/R60k2_3.20.11.zip'
content type 'application/zip' length 1388433 bytes (13.2 MB)
opened url
```

# Installation of Rattle

Environment: History

```
library("rstatista", lib.loc="C:/Program Files/R/R-3.1.1/library")
library("rstatista", lib.loc="C:/Program Files/R/R-3.1.1/library")
install.packages("Rcmdr")
library("Rcmdr", lib.loc="C:/Program Files/R/R-3.1.1/library")
install.packages("Rattle")
```

90% downloaded

URL: ... //cran.rstudio.com/bin/windows/contrib/3.1/RGtk2\_2.20.31.zip

Console

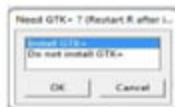
```
install.packages("rattle")
package 'rattle' successfully
The downloaded binary packages are:
C:\users\de11\appdata\local\temp\81mpq82\downloaded_packages
> library("rattle", lib.loc="C:/Program Files/R/R-3.1.1/library")
rattle: a free graphical interface for data mining with R,
version 3.3.0 Copyright (C) 2006-2014 Rogers Kuylenstierna,
type 'rattle()' to shake, 'rattle', and roll your data.
> rattle()
The package 'Rcmdr' is required to display the rattle GUI. It does not
appear to be installed. This package (and its dependencies) can be
installed using the following R command:
install.packages("Rcmdr")
This one-time install will allow access to the full functionality of
rattle.
would you like rattle to install the package now?
[yes/no] yes
trying URL 'http://cran.rstudio.com/bin/windows/contrib/3.1/RGtk2_2.20.31.zip'
content type 'application/zip' length 1388433 bytes (13.2 MB)
opened URL
```

| Package        | Version   |
|----------------|-----------|
| nlme           | 3.3-0     |
| lme4           | 1.1-8     |
| nlmixr2        | 3.0-107   |
| neuralnet      | 3.0-8     |
| parallel       | 3.5.1     |
| rattle         | 3.3.0     |
| Rcmdr          | 2.1-1     |
| RcmdrMisc      | 1.0-2     |
| RcmdrUtilities | 1.0-3     |
| nlmixr         | 1.0-1     |
| rgl            | 0.94.1147 |
| RJava          | 0.10-4    |
| RODBC          | 1.3-10    |
| rsn            | 0.1-0     |
| rstatista      | 0.90.2019 |
| rstanarm       | 2.19-2    |



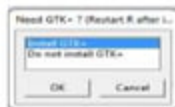
# Installation of Rattle

- GTK+ Installation Necessary

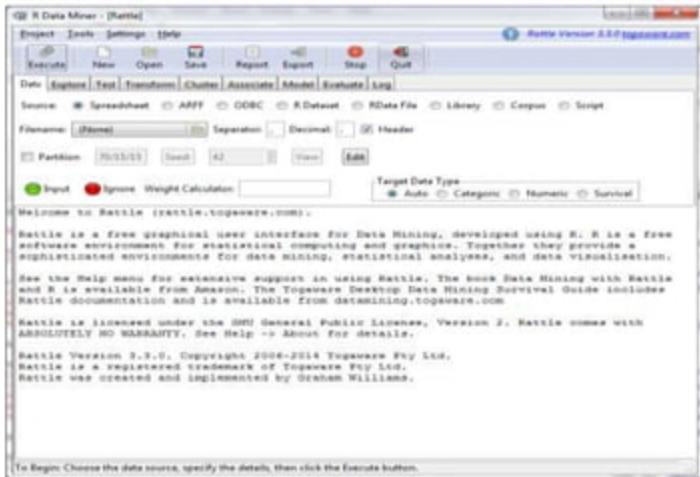


# Installation of Rattle

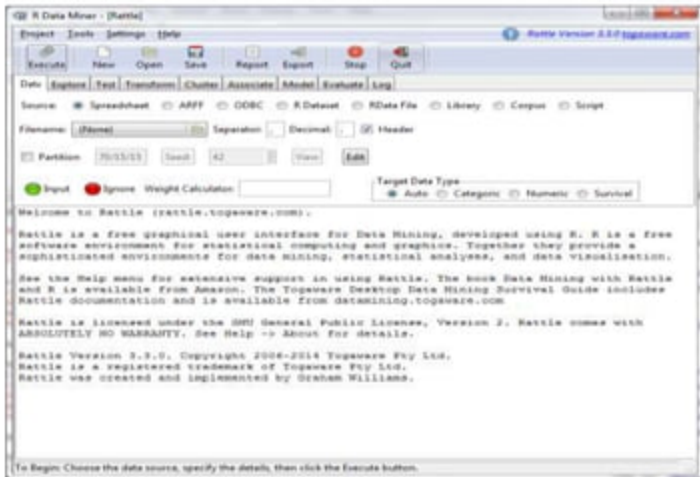
- GTK+ Installation Necessary



# Overview of Rattle



# Demo Rattle



# RStudio

RStudio Desktop enables you with following advantages of native R console

- Syntax highlighting, code completion, and smart indentation
- Execute R code directly from the source editor
- Quickly jump to function definitions
- Easily manage multiple working directories using projects
- Integrated R help and documentation
- Interactive debugger to diagnose and fix errors quickly
- Extensive package development tools

<http://www.rstudio.com/products/>

# RStudio

RStudio Server enables you to provide a browser based interface (the RStudio IDE) to a version of R running on a remote Linux server. Deploying R and RStudio on a server has a number of benefits, including:

- The ability to access your R workspace from any computer in any location;
- Easy sharing of code, data, and other files with colleagues;
- Allowing multiple users to share access to the more powerful compute resources (memory, processors, etc.) available on a well equipped server; and
- Centralized installation and configuration of R, R packages, TeX, and other supporting libraries.

```

new2.R | packages.R | chapter1.Rmd | Untitled1*
Source on Save
1 library(ggplot2)
2 data(diamonds)
3 barplot(diamonds$price)
4 plot(diamonds$price)
5 plot(diamonds$price,diamonds$carat)
6 pie(table(diamonds$cut))
7 boxplot(diamonds$price)
8 boxplot(diamonds$price~diamonds$cut)
9 boxplot(diamonds$price~diamonds$color)
10 plot(diamonds$cut,diamonds$color)
11 hist(diamonds$price)
12 |
**
12.1 (Top Level) | R Script |

```

RStudio -  
Interface

```

Console
> kmeans
function(x, centers, iter.max = 10, nstart = 1, algorithm = c("Hartigan-Wong",
  "Lloyd", "Forgy", "MacQueen"), trace = FALSE)
{
  do_one <- function(nmeth) {
    switch(nmeth, {
      lsteps.Qtran <- 50 * n
      ltran <- c(as.integer(lsteps.Qtran), integer(max(0,
        k - 1)))
      Z <- .Fortran(C_knns, x, n, p, centers = centers,
        as.integer(k), ci = integer(n), c2 = integer(n),
        nc = integer(k), double(k), double(k), ncp = integer(k),
        D = double(n), ltran = ltran, llve = integer(k),
        iter = iter.max, wss = double(k), lfault = as.integer(trace))
      switch(Z$lfault, stop["empty cluster: try a better set of initial centers",
        call. = FALSE], Z$iter <- max(Z$iter, iter.max +
        1L), stop["number of cluster centres must lie between 1 and nrow(x)",
        call. = FALSE], warning(paste("Quick-TRANSFER stage steps exceeded maximum (=
        5d)",
          lsteps.Qtran), call. = FALSE))
    }, {
      Z <- .C(C_kmeans_lloyd, x, n, p, centers = centers,
        k, ci = integer(n), iter = iter.max, nc = integer(k),
        wss = double(k))

```

Environment History

Import Dataset Clear

Global Environment

Data

|          |                            |
|----------|----------------------------|
| diamonds | 53940 obs. of 10 variables |
| iris3    | 50 obs. of 12 variables    |


Values

|   |              |
|---|--------------|
| a | NULL (empty) |
| i | 90L          |

Files Plots Packages Help Viewer

R Search Results Find in Files

## Search Results



The search string was "kmeans"

Vignettes:

[broom::kmeans](#) kmeans with dplyr+broom [HTML source R code](#)

Help pages:

[arnap::Kmeans](#) K-Means Clustering

[broom::augment\\_kmeans](#) Tidying methods for kmeans objects

[e1071::cmeans](#) Fuzzy C-Means Clustering

# R Landscape





# R Documentation

<http://cran.r-project.org/manuals.html>

## Manuals

*edited by the R Development Core Team.*

The following manuals for R were created on Debian Linux and may differ from the manuals for Mac or Windows on platform version of the manuals for each platform are part of the respective R installations. The manuals change with R, hence we provide a version for the patched release version (R-patched) and finally a version for the forthcoming R version that is still in development.

Here they can be downloaded as PDF files, EPUB files, or directly browsed as HTML:

| Manual   | R-release   |
|--|---|
| <b>An Introduction to R</b> is based on the former "Notes on R", gives an introduction to the language and how to use R for doing statistical analysis and graphics.   | <a href="#">HTML</a>   <a href="#">PDF</a>   <a href="#">EPUB</a> |
| <b>R Data Import/Export</b> describes the import and export facilities available either in R itself or via packages which are available from CRAN.   | <a href="#">HTML</a>   <a href="#">PDF</a>   <a href="#">EPUB</a> |
| <b>R Installation and Administration</b>   | <a href="#">HTML</a>   <a href="#">PDF</a>   <a href="#">EPUB</a> |
| <b>Writing R Extensions</b> covers how to create your own packages, write R help files, and the foreign language (C, C++, Fortran, ...) interfaces.  | <a href="#">HTML</a>   <a href="#">PDF</a>   <a href="#">EPUB</a> |
| A draft of <b>The R language definition</b> documents the language <i>per se</i> . That is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming R functions. | <a href="#">HTML</a>   <a href="#">PDF</a>   <a href="#">EPUB</a> |
| <b>R Internals</b> : a guide to the internal structures of R and coding standards for the core team working on R itself.   | <a href="#">HTML</a>   <a href="#">PDF</a>   <a href="#">EPUB</a> |
| <b>The R Reference Index</b> : contains all help files of the R standard and recommended packages in printable form. (9MB, approx. 3500 pages)   | <a href="#">PDF</a>   |

Translations of manuals into other languages than English are available from the [contributed documentation](#) section (only a

The LaTeX or Texinfo sources of the latest version of these documents are contained in every R source distribution (in the form of the respective [archives of the R sources](#). The HTML versions of the manuals are also part of most R installations.

Please check the manuals for R-devel before reporting any issues with the released versions.

# R

# Documentation

## Vignettes

### ggplot2: An Implementation of the Grammar of Graphics

An implementation of the grammar of graphics in R. It combines the advantages of both base and lattice graphics by step from multiple data sources. It also implements a sophisticated multidimensional conditioning system with information, documentation and examples.

Version: 1.0.1  
Depends: R (≥ 2.14), stats, methods  
Imports: plyr (≥ 1.7.1), digest, grid, gtable (≥ 0.1.1), reshape2, scales (≥ 0.2.3), proto, MASS  
Suggests: quantreg, Hmisc, mapproj, maps, hexbin, maptools, multcomp, nlme, testthat, knitr, mgcv  
Enhances: sp  
Published: 2015-03-17  
Author: Hadley Wickham [aut, cre], Winston Chang [aut]  
Maintainer: Hadley Wickham <h.wickham@gmail.com>  
BugReports: <https://github.com/hadley/ggplot2/issues>  
License: GPL-2  
URL: <http://ggplot2.org>, <https://github.com/hadley/ggplot2>  
NeedsCompilation: no  
Citation: [ggplot2.citation.info](#)  
Materials: [README](#) [NEWS](#)  
In views: [Graphics](#), [Phylogenetics](#)  
CRAN checks: [ggplot2.results](#)

#### Downloads:

Reference manual: [ggplot2.pdf](#)  
Vignettes: [Contributing to ggplot2 development](#)  
[ggplot2 release process](#)  
Package source: [ggplot2\\_1.0.1.tar.gz](#)  
Windows binaries: r-devel: [ggplot2\\_1.0.1.zip](#), r-release: [ggplot2\\_1.0.1.zip](#), r-oldest: [ggplot2\\_1.0.1.zip](#)  
OS X Snow Leopard binaries: r-release: not available, r-oldest: [ggplot2\\_1.0.1.tgz](#)  
OS X Mavericks binaries: r-release: [ggplot2\\_1.0.1.tgz](#)  
Old sources: [ggplot2.archive](#)

#### Reverse dependencies:

Reverse depends: [alphahull](#), [AmpliconDuo](#), [aoristic](#), [apsimr](#), [bcrm](#), [bde](#), [benchmark](#), [biomod2](#), [bootnet](#), [brms](#)

# CRAN Views

<http://cran.r-project.org/web/views/>

|   |   |
|---|---|
| <a href="#">Bayesian</a>                  | Bayesian Inference  |
| <a href="#">ChemPhys</a>                  | Chemometrics and Computational Physics                                |
| <a href="#">ClinicalTrials</a>            | Clinical Trial Design, Monitoring, and Analysis                       |
| <a href="#">Cluster</a>                   | Cluster Analysis & Finite Mixture Models                              |
| <a href="#">DifferentialEquations</a>     | Differential Equations  |
| <a href="#">Distributions</a>             | Probability Distributions   |
| <a href="#">Econometrics</a>              | Econometrics  |
| <a href="#">Environmetrics</a>            | Analysis of Ecological and Environmental Data                         |
| <a href="#">ExperimentalDesign</a>        | Design of Experiments (DoE) & Analysis of Experimental Data           |
| <a href="#">Finance</a>                   | Empirical Finance   |
| <a href="#">Genetics</a>                  | Statistical Genetics  |
| <a href="#">Graphics</a>                  | Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization |
| <a href="#">HighPerformanceComputing</a>  | High-Performance and Parallel Computing with R                        |
| <a href="#">MachineLearning</a>           | Machine Learning & Statistical Learning                               |
| <a href="#">MedicalImaging</a>            | Medical Image Analysis  |
| <a href="#">MetaAnalysis</a>              | Meta-Analysis   |
| <a href="#">Multivariate</a>              | Multivariate Statistics   |
| <a href="#">NaturalLanguageProcessing</a> | Natural Language Processing   |
| <a href="#">NumericalMathematics</a>      | Numerical Mathematics   |
| <a href="#">OfficialStatistics</a>        | Official Statistics & Survey Methodology                              |
| <a href="#">Optimization</a>              | Optimization and Mathematical Programming                             |
| <a href="#">Pharmacokinetics</a>          | Analysis of Pharmacokinetic Data                                      |
| <a href="#">Phylogenetics</a>             | Phylogenetics, Especially Comparative Methods                         |
| <a href="#">Psychometrics</a>             | Psychometric Models and Methods                                       |
| <a href="#">ReproducibleResearch</a>      | Reproducible Research   |
| <a href="#">Robust</a>                    | Robust Statistical Methods  |
| <a href="#">SocialSciences</a>            | Statistics for the Social Sciences                                    |
| <a href="#">Spatial</a>                   | Analysis of Spatial Data  |
| <a href="#">SpatioTemporal</a>            | Handling and Analyzing Spatio-Temporal Data                           |
| <a href="#">Survival</a>                  | Survival Analysis   |
| <a href="#">TimeSeries</a>                | Time Series Analysis  |
| <a href="#">WebTechnologies</a>           | Web Technologies and Services   |
| <a href="#">gR</a>                        | gRaphical Models in R   |

# R Community

- email groups <http://www.r-project.org/mail.html>

R-announce

R-help

R-package-devel

R-devel

R-packages

Special Interest Groups

- Stack Overflow [r]
- Twitter #rstats
- Blogs at <http://www.r-bloggers.com/> (573 blogs)

# Stack Overflow

<http://stackoverflow.com/questions/tagged/r>

Stack Overflow

Questions Tags Users Badges Unanswered Ask Question

### Tagged Questions

0 votes  
1 answer  
2 views

**R Count number of rows in one column of a data frame?**

I just want to know how to get `n` to list the number of unique rows of a specific column of a data frame. My guess was `length(unique(column))` though that didn't work.

asked 2 mins ago  
answered 5 m

0 votes  
0 answers  
2 views

**Create interactive webmap with markers in R using Shiny, Leaflet and rCharts**

I am trying to create an interactive webmap in R to display storms using Shiny, Leaflet and rCharts. The structure is heavily based on the <http://rmarkdown.github.io/shinyrmap> app. The idea is that...

asked 2 mins ago  
answered 15 m

0 votes  
0 answers  
7 views

**R - grab a specific character of a specific position**

I would like to divide the last character of a variable. I was wondering if it is possible to select the position with `grep` and divide the character of the particular position. In the example, I...

asked 15 mins ago  
answered 1 m

90,861 questions tagged

Featured on Meta

- April 2015 Community Moderator Election Results
- Hot Meta Posts
- What will be a question about "Your answer couldn't be submitted"?
- The First Answerer will be the one who asks the question - are they even necessary?
- Flagging questions with "delete only" in comments

Favorite Tags

Looking for a job?

# Twitter

<https://twitter.com/search?q=rstats&src=sprv>

Results for #rstats  
Top / All

**Mark Benson** @markbenson · 5m  
Power and heat are related. Here's an R plot I did that proves it on the Kindle Fire. #rstats [vanilladraft.com/s/mes/](http://vanilladraft.com/s/mes/)

Power and temperature

(Watt)

(Watt)

View photo

**Stéphane Fréchette** @sthechete · 8m  
How to get your very own RStudio Server and Shiny Server with DigitalOcean [r-bloggers.com/how-to-get-you...](http://r-bloggers.com/how-to-get-you...) #datascience #feedly #rstats #shiny

**Ankit kansal** @systemmarkt · 8m  
Interesting post on configuring parallel computing on R #rstudio #rstats #dataprocessing #data

**Learn R** @R\_Programming  
How to do parallel computing with R? [statistics.net/parallel-compu...](http://statistics.net/parallel-compu...)  
#rstats #datascience

# Help within R

? "keyword"

? ? "keyword"

Example-

```
> ?kmeans
```

```
|
```

```
> ??kmeans
```

# Functions Used in this Lesson

function(x)

for

library

install.packages

update.packages

ls

rm

print



# Citations and References

```
> citation()
```

```
To cite R in publications use:
```

```
R Core Team (2015). R: A language and environment for statistical computing. R Foundation for  
Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.
```

# Introductory R

```
> Sys.Date()
```

```
[1] "2015-05-10"
```

```
> Sys.time()
```

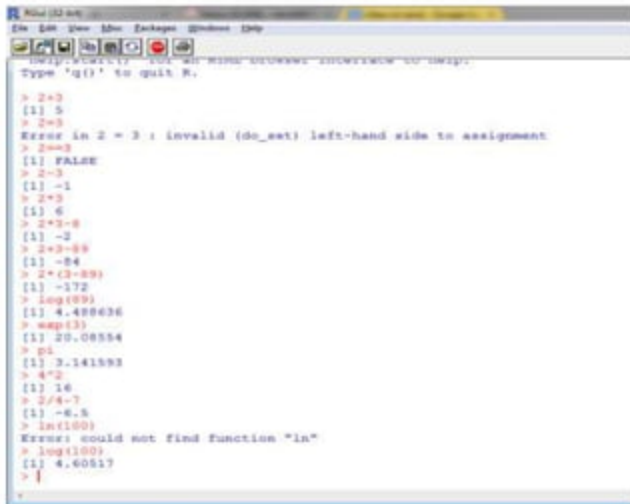
```
[1] "2015-05-10 18:28:32 IST"
```

# R as a Calculator

## Basic Math on R Console

- +
- -
- Log
- Exp
- \*
- /
- ()
- mean
- sum
- sd
- log
- median
- exp

# Demo- Basic Math on R Console

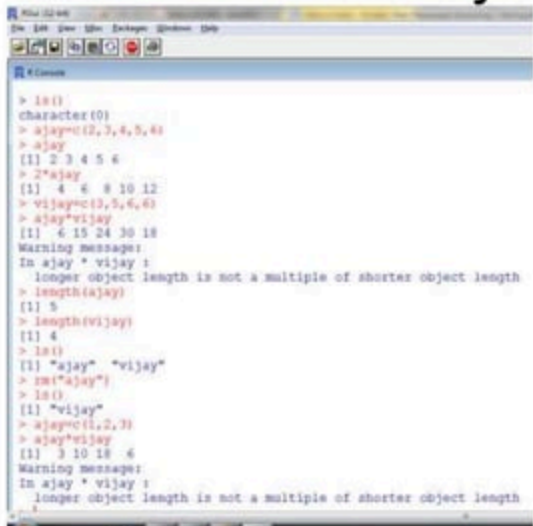


```
RStudio [2]
File Edit View Misc Packages Windows Help
[Icons]
[Help window] [URL: http://www.r-project.org/doc/2.10.0/help-contents-to-help.html]
Type 'q()' to quit R.

> 2+3
[1] 5
> 2=3
Error in 2 = 3 : invalid (do_set) left-hand side to assignment
> 2==3
[1] FALSE
> 2-3
[1] -1
> 2*3
[1] 6
> 2*3-6
[1] -3
> 2*3-89
[1] -84
> 2*(3-89)
[1] -172
> log(100)
[1] 4.488436
> exp(3)
[1] 20.08554
> pi
[1] 3.141593
> 4^2
[1] 16
> 2/4-7
[1] -6.5
> ln(100)
Error: could not find function "ln"
> log(100)
[1] 4.60517
> |
```

Hint- Ctrl +L clears screen

# Demo- Basic Objects on R Console



```
> ls()
character(0)
> ajay=c(2,3,4,5,6)
> ajay
[1] 2 3 4 5 6
> 2*ajay
[1] 4 6 8 10 12
> vijay=c(1,5,6,6)
> ajay*vijay
[1]  6 15 24 30 18
Warning message:
In ajay * vijay :
  longer object length is not a multiple of shorter object length
> length(ajay)
[1] 5
> length(vijay)
[1] 4
> ls()
[1] "ajay" "vijay"
> rm("ajay")
> ls()
[1] "vijay"
> ajay=c(1,2,3)
> ajay*vijay
[1]  3 10 18  6
Warning message:
In ajay * vijay :
  longer object length is not a multiple of shorter object length
```

Functions-

ls() – what objects are here

rm("foo") removes object named foo

Assignment

Using = or -> assigns object names to values

Hint- Up arrow ↑ gives you last typed command

# Functions and Loops

- Loops

```
for (number in 1:5){ print (number) }
```

```
> for (number in 1:5){ print (number) }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
> for (i in 1:5){ print (i) }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
> for (i in 1:5){ rnorm(i,10,10) }
> for (i in 1:5){ print(rnorm(i,10,10)) }
[1] 1.090406
[1] 8.611727 16.670168
[1] 10.84623 13.13938 11.56230
[1] 6.068250 -18.723389 33.174107 -1.320091
[1] 13.939702 -9.037375 13.755986 9.459680 9.625309
> |
```

# Functions and Loops

- Function

`functionajay=function(a)(a^2+2*a+1)`

```
> functionajay=function(a)(a^2+2*a+1)
> functionajay(1)
[1] 4
> functionajay(2)
[1] 9
> for (i in 1:5){ print(rnorm(i) )
Error: unexpected ')' in "for (i in 1:5){ print(rnorm(i) )"
>
> for (i in 1:5){ print(functionajay(i)) }
[1] 4
[1] 9
[1] 16
[1] 25
[1] 36
> |
```

Hint: Always match brackets

Each ( deserves a )

Each { deserves a }

Each [ deserves a ]

# Other sources to learn R

swirlstats

<http://swirlstats.com/>

datacamp

<https://www.datacamp.com/>

codeschool

<http://tryr.codeschool.com/>

coursera

<https://www.coursera.org/course/compdata>

<https://www.coursera.org/course/rprog>

The logo for the 'swirl' R package. It features the word 'swirl' in a dark grey, sans-serif font. The letters 's' and 'l' are enclosed in large, blue curly braces. The letter 'i' has a blue dot above it.



# Good coding practices

- Use # for comment
- Use git for version control
- Use Rstudio for multiple lines of code

# Functions in R

- custom functions
- source code for a function

```
function (x, centers, iter.max = 10, nstart = 1, algorithm = c("Martigan-Wong",
  "Lloyd", "Forgy", "MacQueen"), trace = FALSE)
{
  do_one <- function(nmeth) {
    switch(nmeth, {
      lsteps.Qtran <- 50 * n
      ltran <- c(as.integer(lsteps.Qtran), integer(max(0,
        k - 1)))
      Z <- .Fortran(C_kmeans, x, n, p, centers = centers,
        as.integer(k), c1 = integer(n), c2 = integer(n),
        nc = integer(k), double(k), double(k), nsp = integer(k),
        D = double(n), ltran = ltran, lrow = integer(k),
        lter = iter.max, wss = double(k), ifault = as.integer(trace))
      switch(Z$ifault, stop("empty cluster: try a better set of initial centers",
        call. = FALSE), Z$iter <- max(Z$iter, iter.max +
        1), stop("number of cluster centres must lie between 1 and nrow(x)",
        call. = FALSE), warning(gettextf("Quick-TRANSFER stage steps exceeded maximum (%
        %d)",
        lsteps.Qtran), call. = FALSE))
    }, {
      Z <- .C(C_kmeans_Lloyd, x, n, p, centers = centers,
        k, c1 = integer(n), lter = iter.max, nc = integer(k),
        wss = double(k))
    }, {
      Z <- .C(C_kmeans_MacQueen, x, n, p, centers = as.double(centers),
        k, c1 = integer(n), lter = iter.max, nc = integer(k),
        wss = double(k))
    })
    if (n23 <- any(nmeth == c(2L, 3L))) {
      if (any(Z$nc == 0))
        warning("empty cluster: try a better set of initial centers",
          call. = FALSE)
    }
  }
}
```

# HOMework TIME !



# Learning Objectives

- how to input data in R using various ways
- how to check for correct data input
- how to use special packages for fast data input
- how to input data from statistical file formats
- how to input data from databases
- how to input data from web (web scraping)

# What will you learn from this lesson

- data input from various kinds of format
- efficient data input via various packages
- sql to R
- web scraping
- piping in R
- using json in R

# Environment

ls() -lists objects

rm()-removes an object

gc() -does garbage collection and frees up  
memory

Console - / 20

Copyright (C) 2015 The R Foundation for Statistical Computing  
Platform: i686-pc-linux-gnu (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.

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.

[Workspace loaded from ~/.RData]

```
> ls()
[1] "a"      "i"      "iris3"
> rm(a)
> ls()
[1] "i"      "iris3"
> gc()
      used (Mb) gc trigger (Mb) max used (Mb)
Ncells 334887  9.0   597831 16.0   407500 10.9
Vcells 624499  4.8   1215808  9.3   1215802  9.3
>
```

Environment History

Import Dataset Clear

Global Environment

Data

iris3 50 obs. of 12 variables

Values

i 90L

Files Plots Packages Help Viewer

R: Search Results

Search Results



The search string was "kmeans"

Vignettes:

[broom::kmeans](#) kmeans with dplyr+broom

[HTML source file code](#)

Help pages:

[arnap::Kmeans](#) K-Means Clustering

[broom::augment.kmeans](#) Tidying methods for kmeans objects

[e1071::cmeans](#) Fuzzy C-Means Clustering

Console - / 20

Copyright (C) 2015 The R Foundation for Statistical Computing  
Platform: i686-pc-linux-gnu (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.

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.

[Workspace loaded from ~/.RData]

```
> ls()
[1] "a"      "i"      "iris3"
> rm(a)
> ls()
[1] "i"      "iris3"
> gc()
      used (Mb) gc trigger (Mb) max used (Mb)
Ncells 334887  9.0   597831 16.0   407500 10.9
Vcells 624499  4.8   1215808  9.3   1215802  9.3
>
```

Environment History




Global Environment

Data

iris3 50 obs. of 12 variables

Values

1 90L

Files Plots Packages Help Viewer





R: Search Results

Search Results



The search string was "kmeans"

Vignettes:

[broom::kmeans](#) kmeans with dplyr+broom

[HTML source file code](#)

Help pages:

[arnap::Kmeans](#) K-Means Clustering

[broom::augment.kmeans](#) Tidying methods for kmeans objects

[e1071::cmeans](#) Fuzzy C-Means Clustering



# File System

getwd()- get working directory

setwd()- set or change working directory

dir() - lists files in working directory

Console ~/Desktop/new/ :D

```
> getwd()
[1] "/home/ajay/Desktop"
> setwd("~/home/ajay/Desktop/new")
> dir()
[1] "obama"
> |
```

Environment History




Global Environment

Data

iris3 50 obs. of 12 variables

Values

l 90L

Files Plots Packages Help Viewer






Home

| Name  | Size     | Modified              |
|---|----------|-----------------------|
| .RData  | 3.8 KB   | May 2, 2015, 11:47 AM |
| .Rhistory   | 10.7 KB  | May 10, 2015, 2:20 PM |
| 17811636-Brain-function-as-gears-and-cogs-in-the-shape-of-a-human-head-as-a-medical-symbol-of-mental-health-c-Stock-Photo.jpg | 139.8 KB | Apr 16, 2015, 9:41 AM |
| 21.png  | 352.7 KB | May 4, 2015, 5:18 PM  |
| 2167434.jpg   | 32.8 KB  | May 4, 2015, 5:36 PM  |
| a.out   | 7.7 KB   | May 2, 2015, 2:26 PM  |
| anaconda  |          |                       |
| animation   |          |                       |
| animation2  |          |                       |
| backports-3.18.1-1  |          |                       |
| backports-3.18.1-1.tar.xz   | 8.6 MB   | Dec 22, 2014, 3:14 AM |
| Call R and Python from base SAS.html  | 48.4 KB  | May 5, 2015, 12:53 PM |

Console ~/Desktop/new/ R

```
> getwd()
[1] "/home/ajay/Desktop"
> setwd("/home/ajay/Desktop/new")
> dir()
[1] "obama"
> |
```

Environment History

Import Dataset Clear

Global Environment

Data

iris3 50 obs. of 12 variables

Values

l 90L

Files Plots Packages Help Viewer

New Folder Delete Rename More

Home

| Name  | Size     | Modified              |
|---|----------|-----------------------|
| .RData  | 3.8 KB   | May 2, 2015, 11:47 AM |
| .Rhistory   | 10.7 KB  | May 10, 2015, 2:20 PM |
| 17811636-Brain-function-as-gears-and-cogs-in-the-shape-of-a-human-head-as-a-medical-symbol-of-mental-health-c-Stock-Photo.jpg | 139.8 KB | Apr 16, 2015, 9:41 AM |
| 21.png  | 352.7 KB | May 4, 2015, 5:18 PM  |
| 2167434.jpg   | 32.8 KB  | May 4, 2015, 5:36 PM  |
| a.out   | 7.7 KB   | May 2, 2015, 2:26 PM  |
| anaconda  |          |                       |
| animation   |          |                       |
| animation2  |          |                       |
| backports-3.18.1-1  |          |                       |
| backports-3.18.1-1.tar.xz   | 8.6 MB   | Dec 22, 2014, 3:14 AM |
| Call R and Python from base SAS.html  | 48.4 KB  | May 5, 2015, 12:53 PM |

Console ~/Desktop/new/ :R

```
> getwd()
[1] "/home/ajay/Desktop"
> setwd("~/home/ajay/Desktop/new")
> dir()
[1] "obama"
> |
```

Environment History

Import Dataset Clear

Global Environment

Data

iris3 50 obs. of 12 variables

Values

l 90L

Files Plots Packages Help Viewer

New Folder Delete Rename More

| Name  | Size     | Modified              |
|---|----------|-----------------------|
| .RData  | 3.8 KB   | May 2, 2015, 11:47 AM |
| .Rhistory   | 10.7 KB  | May 10, 2015, 2:20 PM |
| 17811636-Brain-function-as-gears-and-cogs-in-the-shape-of-a-human-head-as-a-medical-symbol-of-mental-health-c-Stock-Photo.jpg | 139.8 KB | Apr 16, 2015, 9:41 AM |
| 21.png  | 352.7 KB | May 4, 2015, 5:18 PM  |
| 2167434.jpg   | 32.8 KB  | May 4, 2015, 5:36 PM  |
| a.out   | 7.7 KB   | May 2, 2015, 2:26 PM  |
| anaconda  |          |                       |
| animation   |          |                       |
| animation2  |          |                       |
| backports-3.18.1-1  |          |                       |
| backports-3.18.1-1.tar.xz   | 8.6 MB   | Dec 22, 2014, 3:14 AM |
| Call R and Python from base SAS.html  | 48.4 KB  | May 5, 2015, 12:53 PM |

## Assigning

```
objectname=read.csv(filepath,parameters)
```

OR

```
objectname<-read.csv(filepath,parameters)
```

# Data Input

`read.table()` or `read.csv()`

`read.spss()`

`read.sas7bdat()`

# read.table()



**Data Input**

## Description

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

## Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"",
  dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
  as.is.names, col.names, as.is = !stringsAsFactors,
  na.strings = "NA", colClasses = NA, rows = -1,
  skip = 0, check.names = TRUE, fill = !blank.lines.skip,
  strip.white = FALSE, blank.lines.skip = TRUE,
  comment.char = "#",
  allowEscapes = FALSE, flush = FALSE,
  stringsAsFactors = default.stringsAsFactors(),
  fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "#", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "#", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "#", ...)

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "#", ...)
```

## Arguments

<https://stat.ethz.ch/R-manual/R-devel/library/utils/html/read.table.html>

# Statistical formats

- read.spss from foreign package
- read.sas7bdat from sas7bdat package



The screenshot shows a web browser window displaying the documentation for the `read.sas7bdat` function. The browser's address bar shows the URL `https://cran.r-project.org/web/packages/sas7bdat/read.sas7bdat.pdf`. The left sidebar contains a navigation menu with items: `read.sas7bdat`, `sas7bdat`, and `index`. The main content area displays the following information:

`read.sas7bdat` SAS Database Reader (experimental)

**Description**  
Read SAS files in the sas7bdat data format.

**Usage**  
`read.sas7bdat (file, debug = FALSE)`

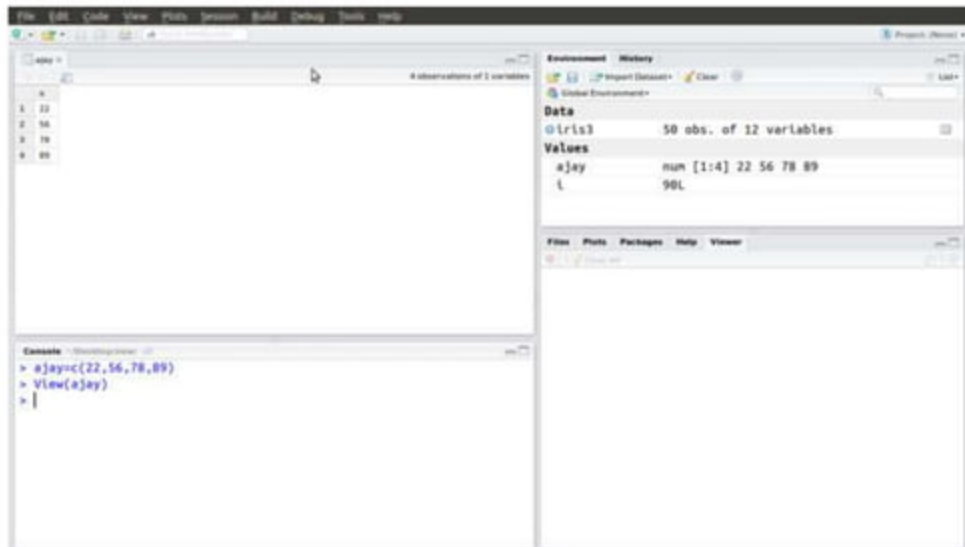
**Arguments**

| Argument           | Type      | Description   |
|--------------------|-----------|---|
| <code>file</code>  | character | Path to a file or an URL.                                   |
| <code>debug</code> | logical   | Save R session environment as attribute of returned object. |

**Value**  
A data frame corresponding to the SAS database. The returned data frame has an column, `info` attribute and other attributes that contain additional information about each field in the data frame.



# Manual Entry



The screenshot shows the RStudio interface with the following components:

- Environment History:** Shows the current environment with a variable named `iris` containing 50 observations of 12 variables.
- Values:** Displays the values for the variable `ajay`, which is a numeric vector of length 4 with values 22, 56, 78, and 89. The variable `l` is shown as a character vector with the value "90L".
- Console:** Shows the R code used to create and view the data:

```
> ajay=c(22,56,78,89)
> View(ajay)
> |
```

# Manual Editing

The screenshot shows the RStudio interface. The console on the left contains the following commands:

```
> ajoy=c(23,56,78,89)
> edit(ajoy)
```

The Environment pane on the right shows the variable `ajoy` as a numeric vector of length 4 with values 23, 56, 78, and 89.

A dialog box titled `edit` is open in the center, displaying the current values of the vector:

```
1 c(23, 56, 78, 89)
2
```

The dialog box has `Save` and `Cancel` buttons at the bottom.

The bottom right pane shows the documentation for the `edit` function, including the following usage information:

```
## Default S3 method:
edit(x) = WSL, file = "", title = WSL,
editor = getOption("editor"), ...

ajoy = WSL, file = ""
ajoy[1:n] = WSL, file = ""
ajoy[1:n] = WSL, file = ""
ajoy[1:n] = WSL, file = ""
ajoy[1:n] = WSL, file = ""
```

# Manual Editing

The screenshot displays the RStudio interface. The **Console** on the left shows the commands `data(iris)` and `edit(iris)`. The **Data Editor** window in the center shows a table of the iris dataset with columns: `Species`, `Petal.Width`, `Petal.Length`, `Sepal.Width`, and `Sepal.Length`. The **Environment** pane on the right shows the `iris` object with 50 observations and 12 variables. The **Plots** pane is empty. The **Source** pane at the bottom shows the R script code.

```
data(iris)
edit(iris)
```

|    | Species | Petal.Width | Petal.Length | Sepal.Width | Sepal.Length |
|----|---------|-------------|--------------|-------------|--------------|
| 1  | setosa  | 0.2         | 1.4          | 1.0         | 5.0          |
| 2  | setosa  | 0.2         | 1.4          | 1.0         | 4.9          |
| 3  | setosa  | 0.2         | 1.3          | 1.2         | 4.7          |
| 4  | setosa  | 0.2         | 1.5          | 1.1         | 4.6          |
| 5  | setosa  | 0.2         | 1.4          | 1.0         | 5.0          |
| 6  | setosa  | 0.2         | 1.4          | 1.0         | 5.4          |
| 7  | setosa  | 0.3         | 1.4          | 1.0         | 4.6          |
| 8  | setosa  | 0.2         | 1.5          | 1.4         | 5.0          |
| 9  | setosa  | 0.2         | 1.4          | 1.0         | 4.4          |
| 10 | setosa  | 0.1         | 1.5          | 1.3         | 4.9          |
| 11 | setosa  | 0.2         | 1.5          | 1.7         | 5.4          |
| 12 | setosa  | 0.1         | 1.4          | 1.4         | 4.8          |
| 13 | setosa  | 0.1         | 1.3          | 1.0         | 4.8          |
| 14 | setosa  | 0.1         | 1.4          | 1.0         | 4.3          |
| 15 | setosa  | 0.2         | 1.2          | 1.0         | 5.0          |
| 16 | setosa  | 0.4         | 1.4          | 1.0         | 5.7          |
| 17 | setosa  | 0.4         | 1.4          | 1.0         | 5.4          |
| 18 | setosa  | 0.3         | 1.4          | 1.0         | 5.1          |
| 19 | setosa  | 0.3         | 1.7          | 1.0         | 5.7          |
| 20 | setosa  | 0.3         | 1.5          | 1.0         | 5.1          |
| 21 | setosa  | 0.2         | 1.7          | 1.4         | 5.4          |
| 22 | setosa  | 0.4         | 1.5          | 1.0         | 5.1          |
| 23 | setosa  | 0.2         | 1.5          | 1.0         | 4.6          |
| 24 | setosa  | 0.5         | 1.5          | 1.7         | 5.1          |
| 25 | setosa  | 0.2         | 1.5          | 1.9         | 4.8          |

Environment History

- iris 50 obs. of 12 variables
- iris
- run [1:4] 23 56 78 89
- 99L
- ~Promises

Plots Packages Help Viewer

Open a Text Editor

```
## A function to open a text editor
##
## @param filename The filename to open
## @return The path to the text editor
##
## Usage
##
##   open(filename)
##
## Example
##
##   open("iris.csv")
##
## Details
##
##   This function uses the system() function to open a text editor.
##   The system() function returns the exit status of the command.
##   If the exit status is 0, the text editor was opened successfully.
##   If the exit status is non-zero, the text editor was not opened.
##
## See Also
##
##   edit()
##
## Source
##
##   https://www.rdocumentation.org/packages/iris/versions/0.1-10
```

# readr from Hadley

The goal of *readr* is to provide a fast and friendly way to read tabular data into R. The most important functions are:

- Read delimited files: `read_delim()`, `read_csv()`, `read_tsv()`, `read_csv2()`.
- Read fixed width files: `read_fwf()`, `read_table()`.
- Read lines: `read_lines()`.
- Read whole file: `read_file()`.
- Re-parse existing data frame: `type_convert()`.

# readr from Hadley

Source Data - <https://bit.ly/dsdata>

```
> library(readr)
> system.time(read_csv("BigDiamonds.csv"))
|=====| 100% 49 MB
  user system elapsed
 2.396  0.068  2.448
Warning message:
597311 problems parsing 'BigDiamonds.csv'. See problems(...) for more details.
```

# readxl from Hadley

Readxl supports both the legacy `.xls` format and the modern xml-based `.xlsx` format. `.xls` support is made possible the with `libxls` C library, which abstracts away many of the complexities of the underlying binary format. To parse `.xlsx`, we use the `RapidXML` C++ library.

```
read_excel("my-old-spreadsheet.xls")
read_excel("my-new-spreadsheet.xlsx")

read_excel("my-spreadsheet.xls", sheet = "data")
read_excel("my-spreadsheet.xls", sheet = 2)

read_excel("my-spreadsheet.xls", na = "NA")
```

<https://github.com/hadley/readxl>

# data.table

fread is the fastest way to read data

```
> b=fread("BigDiamonds.csv")  
Read 598024 rows and 13 (of 13) columns from 0.049 GB file in 00:00:04
```

# data.table

fread is the fastest way to read data

```
> b=fread("BigDiamonds.csv")  
Read 598024 rows and 13 (of 13) columns from 0.049 GB file in 00:00:04
```



# data.table

fread is the fastest way to read data

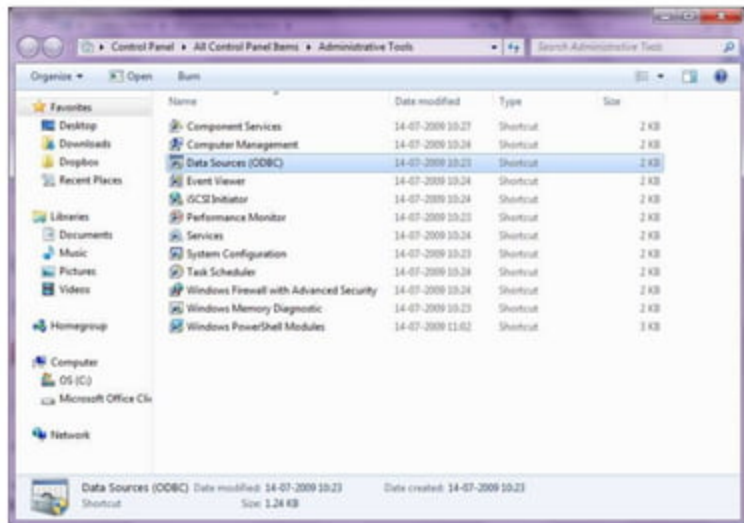
```
> system.time(read_csv("BigDiamonds.csv"))
  user system elapsed
2.552  0.028  2.581
Warning message:
597311 problems parsing 'BigDiamonds.csv'. See problems(...) for more details.
> system.time(fread("BigDiamonds.csv"))
  user system elapsed
1.532  0.012  1.540
> system.time(read.csv("BigDiamonds.csv"))
  user system elapsed
10.892  0.032 10.922
|
```

# Some learnings

1. Multiple packages can do the same thing faster or slower in R
2. Knowing the right package is the essential difference as a data scientist
3. Putting code within `system.time()` helps measure speed

also see <http://adv-r.had.co.nz/Profiling.html> for advanced ways to speed up code

# Creating DSN (Optional)



# Creating DSN (in Windows)

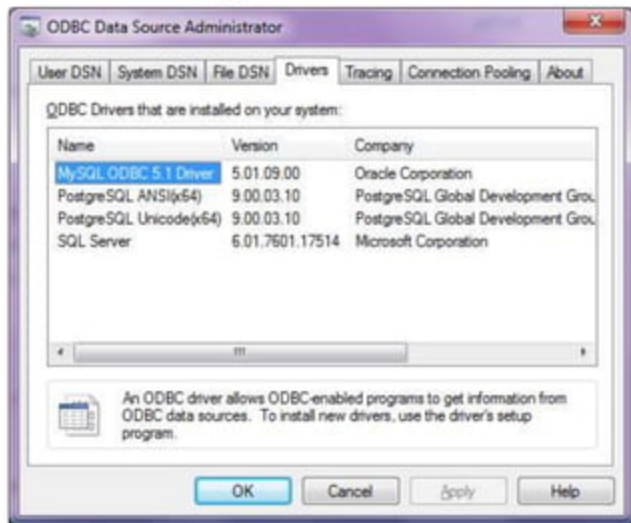
A Data Source Name (DSN) is the logical name that is used by Open Database Connectivity (ODBC) to refer to the drive and other information that is required to access data. The name is used by Internet Information Services for a connection to an ODBC data source, such as a Microsoft SQL Server database.

<https://support.microsoft.com/en-us/kb/kbview/300596>

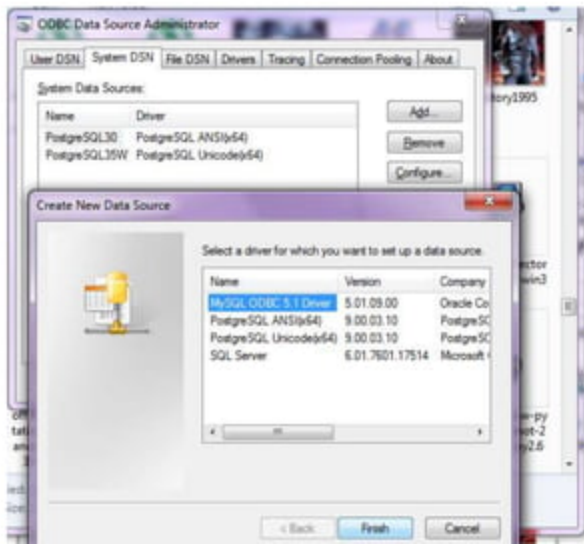
# Creating DSN (in Windows)

1. Click **Start**, point to **Control Panel**, double-click **Administrative Tools**, and then double-click **Data Sources(ODBC)**.
2. Click the **System DSN** tab, and then click **Add**.
3. Click the database driver that corresponds with the database type to which you are connecting, and then click **Finish**.
4. Type the data source name. Make sure that you choose a name that you can remember. You will need to use this name later.
5. Click **Select**.
6. Click the correct database, and then click **OK**.
7. Click **OK**, and then click **OK**.

# Creating DSN



# Creating DSN

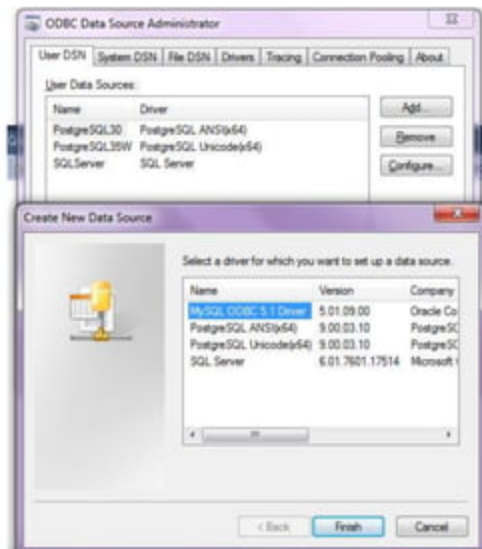


# Creating DSN





# Creating DSN



# RODBC

```
> library(RODBC)
> odbcDataSources()
> ajay=odbcConnect("MySQL",uid="root",pwd="XX")
> ajay
> sqlTables(ajay)
> tested=sqlFetch(ajay,"host")
```

# From Databases

The [RODBC](#) package provides access to databases through an **ODBC** interface.

The primary functions are

- **odbcConnect(*dsn*, uid="", pwd="")** Open a connection to an ODBC database
- **sqlFetch(*channel*, *sqltable*)** Read a table from an ODBC database into a data frame

Hint- a good site to revise R

<http://www.statmethods.net>

# sqlite

<http://cran.r-project.org/web/packages/RSQLite/RSQLite.pdf> embeds the SQLite database engine in R and provides an interface compliant with the DBI package.

SQLite is a software library that implements a [self-contained](#), [serverless](#), [zero-configuration](#), [transactional](#) SQL database engine.

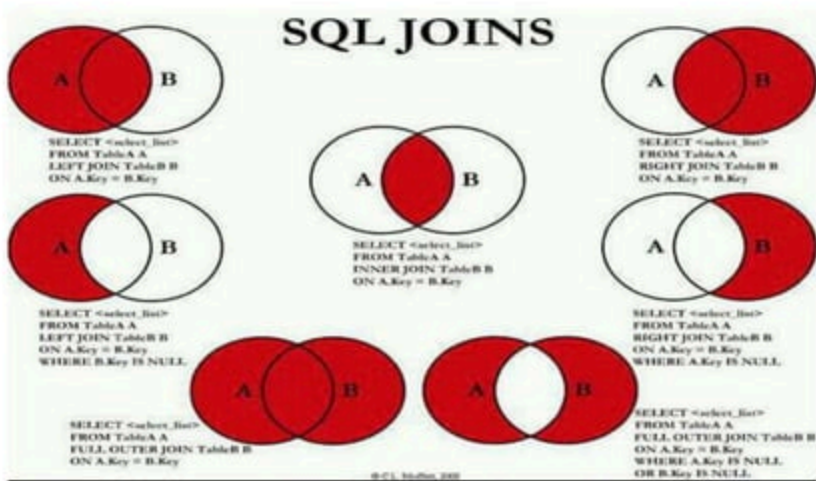
SQLite is the [most widely deployed](#) database engine in the world

```
library(RSQLite)
con <- dbConnect("SQLite", dbname = "sample_db")
# read csv file into sql database
dbWriteTable(con, name="sample_data", value="sample_data.csv", row.names=FALSE, header=TRUE, sep = ",")
```

<http://cran.r-project.org/web/packages/sqlf/index.html> Manipulate R data frames using SQL

`read.csv.sql` in the `sqlf` package imports data into a temporary SQLite database and then reads it into R.

# A Detour to SQL Joins (Optional)



# RMySQL

```
install.packages("RMySQL")
```

```
library(RMySQL)
```

```
mydb = dbConnect(MySQL(), user='user', password='password', dbname='database_name', host='host')
```

```
dbListTables(mydb)
```

```
dbListFields(mydb, 'some_table')
```

```
dbSendQuery(mydb, 'drop table if exists some_table, some_other_table')
```

```
dbWriteTable(mydb, name='table_name', value=data.frame.name)
```

# Other databases

Teradata <https://github.com/Teradata/teradataR>

PostgreSQL <http://cran.r-project.org/web/packages/RPostgreSQL/>

MongoDB <http://cran.r-project.org/web/packages/mongolite/index.html>

couchDB <http://cran.r-project.org/web/packages/couchDB/index.html>

MonetDB <http://cran.r-project.org/web/packages/MonetDB.R/index.html>

# Other data sources

**Cassandra with R** <http://cran.r-project.org/web/packages/RCassandra/RCassandra.pdf>

## Neo4j with R

<http://things-about-r.tumblr.com/post/47392314578/venue-recommendation-a-simple-use-case-connecting-r>

**R with Hadoop Stack** <https://github.com/RevolutionAnalytics/RHadoop/wiki>

- NEW! [ravro](#) - read and write files in avro format
- [plymr](#) - higher level plyr-like data processing for structured data, powered by `rmr`
- [rmr](#) - functions providing Hadoop MapReduce functionality in R
- [rhdfs](#) - functions providing file management of the HDFS from within R
- [rhbase](#) - functions providing database management for the HBase distributed database from within R

<https://amplab-extras.github.io/SparkR-pkg/> SparkR is an R package to use Spark from R.



# Web Scraping

**Web scraping** (web harvesting or **web data extraction**) is a computer software technique of extracting information from websites.

example - python (scrapy and beautiful soup)



The screenshot shows the homepage of the Beautiful Soup project. At the top, there is a navigation bar with links for 'Home', 'Documentation', 'FAQ', 'Examples', 'API', and 'Contributing'. The main heading is 'Beautiful Soup' in a purple font, followed by the subtitle 'A Python library for parsing HTML and XML documents'. Below this, there is a list of features and a 'Download Beautiful Soup' button. On the right side, there is a black and white illustration of a bird, possibly a crow or raven, perched on a branch.



The screenshot shows the homepage of the Scrapy project. At the top, there is a navigation bar with links for 'Home', 'Documentation', 'FAQ', 'Examples', 'API', and 'Contributing'. The main heading is 'Meet Scrapy' in a green font, followed by the subtitle 'A fast, high-level web crawling and web scraping framework, used to crawl websites and extracting structured data from them. It has built-in support for distributed crawling, using Redis and Amazon EC2 to scale up crawling of large websites. It features a rich ecosystem of extensions, making it the most powerful web scraping framework for Python.

Below the heading, there is a green button that says 'Install latest version: Scrapy 0.24'. To the left of the button, there is a green icon of a shovel, representing the Scrapy logo. Below the icon, there is a code block showing a snippet of Python code. At the bottom of the page, there are three green circular icons: a lightning bolt, a person, and a gear.

# Web Scraping

- readlines

```
> url="http://nytimes.com"
> ajoy=readlines(url)
Error: could not find function "readlines"
> ajoy=readlines(url)
> head(ajoy)
[1] "<!DOCTYPE html>"
[2] "<!--[if (gt IE 9)]!(IE)]> <!--> <html lang=\"en\" class=\"no-js edition-domestic app-homepage\" itemscope xmlns:og$
[3] "<!--[if IE 9]> <html lang=\"en\" class=\"no-js ie9 lt-ie10 edition-domestic app-homepage\" xmlns:og=\"http://opengr$
[4] "<!--[if IE 8]> <html lang=\"en\" class=\"no-js ie8 lt-ie10 lt-ie9 edition-domestic app-homepage\" xmlns:og=\"http://$
[5] "<!--[if (lt IE 8)]> <html lang=\"en\" class=\"no-js lt-ie10 lt-ie9 lt-ie8 edition-domestic app-homepage\" xmlns:og=$
[6] "<head>"
> tail(ajoy)
[1] "<div id=\"ab3\" class=\"ad ab3-ad hidden\"></div>"
[2] "<div id=\"prop1\" class=\"ad prop1-ad hidden\"></div>"
[3] "<div id=\"prop2\" class=\"ad prop2-ad hidden\"></div>"
[4] "<div id=\"Anchor\" class=\"ad anchor-ad hidden\"></div>"
[5] "<script type=\"text/javascript\">window.NREUM||(NREUM={});NREUM.info={\"beacon\":\"beacon-6.newrelic.com\", \"licens$
[6] "</html>"
> |
```

Hint : R is case sensitive

readlines is not the same as readLines

Hint : Use head() and tail() to inspect objects

Other packages are XML and Curl

Case Study- <http://decisionstats.com/2013/04/14/using-r-for-cricket-analysis-rstats/>

# curl

cURL is a computer software project providing a library and command-line tool for transferring data using various protocols. The **cURL** project produces two products, libcurl and **cURL**.

The RCurl package is an R-interface to the [libcurl](#) library that provides HTTP facilities. This allows us to download files from Web servers, post forms, use HTTPS (the secure HTTP), use persistent connections, upload files, use binary content, handle redirects, password authentication, etc.

The primary top-level entry points are

- [getURL\(\)](#)
- [getURLContent\(\)](#)
- [getForm\(\)](#)
- [postForm\(\)](#)

<http://www.omegahat.org/RCurl/RCurlJSS.pdf>

# Rcurl

```
File Edit Code View Plots Session Build Debug Tools Help
Untitled1
1 library(Rcurl)
2 h = getCurlHandle()
3 getURI("http://www.omegahat.org/Rcurl/index.html", curl = h)
4 names(getCurlInfo(h))
```

1:1 (Top Level) 1

Console

```
oop\>REventLoop</a>).\nWe can potentially turn them into regular
).\n\n\n<h2>License</h2>\nThis is distributed under the <a href=
e</a>\nin the same spirit as libcurl itself.\n\n<hr>\n<address><a
Temple Lang</a>\n<a href=mailto:duncan@wald.ucdavis.edu>&t;dunca
t -->\nLast modified: Mon May 25 11:35:38 PDT 2009\n<!-- hhmts en
> names(getCurlInfo(h))
[1] "effective.url"           "response.code"         "total.t
[5] "connect.time"           "pretransfer.time"     "size.up
[9] "speed.download"        "speed.upload"         "header.
[13] "ssl.verifyresult"       "filetime"             "content
[17] "starttransfer.time"     "content.type"         "redirec
[21] "private"                "http.connectcode"     "httpaut
[25] "os.errno"               "num.connects"         "ssl.eng
[29] "lastsocket"             "ftp.entry.path"       "redirec
[33] "appconnect.time"        "certinfo"             "conditi
> |
```

# XML

```
File Edit Code View Plots Session Build Debug Tools Help
Source on Save Run
1 library(XML)
2 theurl="http://stats.espncricinfo.com/ci/engine/stats/index.html?class=1;team=6;template=results;type=batting"
3 #Note I can also break the url string and use paste command to modify this url with parameters
4 table2 <- readHTMLTable(theurl)
5 table_cricket=table2$"Overall figures"
6 head(table_cricket)
```

1.1 (Top Level)

Console

```
> library(XML)
> theurl="http://stats.espncricinfo.com/ci/engine/stats/index.html?class=1;team=6;template=results;type=batting"
> #Note I can also break the url string and use paste command to modify this url with parameters
> table2 <- readHTMLTable(theurl)
> table_cricket=table2$"Overall figures"
> head(table_cricket)
```

|   | Player       | Span      | Mat | Inns | NO | Runs  | HS   | Ave   | 100 | 50 | 0  |
|---|--------------|-----------|-----|------|----|-------|------|-------|-----|----|----|
| 1 | SR Tendulkar | 1989-2013 | 200 | 329  | 33 | 15921 | 248* | 53.78 | 51  | 68 | 14 |
| 2 | R Dravid     | 1996-2012 | 163 | 284  | 32 | 13265 | 270  | 52.63 | 36  | 63 | 7  |
| 3 | SM Gavaskar  | 1971-1987 | 125 | 214  | 16 | 10122 | 236* | 51.12 | 34  | 45 | 12 |
| 4 | VVS Laxman   | 1996-2012 | 134 | 225  | 34 | 8781  | 201  | 45.97 | 17  | 56 | 14 |
| 5 | V Sehwag     | 2001-2013 | 103 | 178  | 6  | 8503  | 319  | 49.43 | 23  | 31 | 16 |
| 6 | SC Ganguly   | 1996-2008 | 113 | 188  | 17 | 7212  | 239  | 42.17 | 16  | 35 | 13 |

# json format

## jsonlite for json data

<http://arxiv.org/abs/1403.2805>

```
> library(jsonlite)
```

```
Attaching package: 'jsonlite'
```

```
The following object is masked from 'package:utils':
```

```
View
```

```
> library(httr)
```

```
> library(curl)
```

```
> zips <- stream_in(curl("https://media.mongodb.org/zips.json"))
```

```
opening curl input connection.
```

```
Found 29353 lines...
```

```
binding pages together (no custom handler).
```

```
closing curl input connection.
```

```
>
```

```
> head(zips)
```

|   | _id   | city        | loc                 | pop   | state |
|---|-------|-------------|---------------------|-------|-------|
| 1 | 01001 | AGAWAM      | -72.62274, 42.07021 | 15338 | MA    |
| 2 | 01002 | CUSHMAN     | -72.51565, 42.37702 | 36963 | MA    |
| 3 | 01005 | BARRE       | -72.10835, 42.40970 | 4546  | MA    |
| 4 | 01007 | BELCHERTOWN | -72.41095, 42.27510 | 10579 | MA    |
| 5 | 01008 | BLANDFORD   | -72.93611, 42.18295 | 1240  | MA    |
| 6 | 01010 | BRIMFIELD   | -72.18846, 42.11654 | 3706  | MA    |

```
{ "id": "01001", "city": "AGAWAM", "loc": [ -72.622739, 42.070200 ], "pop": 15338, "state": "MA" }
{ "id": "01002", "city": "CUSHMAN", "loc": [ -72.51564999999999, 42.377017 ], "pop": 36963, "state": "MA" }
{ "id": "01005", "city": "BARRE", "loc": [ -72.10835400000001, 42.409698 ], "pop": 4546, "state": "MA" }
{ "id": "01007", "city": "BELCHERTOWN", "loc": [ -72.41095300000001, 42.275103 ], "pop": 10579, "state": "MA" }
{ "id": "01008", "city": "BLANDFORD", "loc": [ -72.936114, 42.182949 ], "pop": 1240, "state": "MA" }
{ "id": "01010", "city": "BRIMFIELD", "loc": [ -72.188455, 42.116543 ], "pop": 3706, "state": "MA" }
{ "id": "01011", "city": "CHESTER", "loc": [ -72.908761, 42.279421 ], "pop": 1608, "state": "MA" }
{ "id": "01012", "city": "CHESTERFIELD", "loc": [ -72.833309, 42.38167 ], "pop": 177, "state": "MA" }
{ "id": "01013", "city": "CHICOPEE", "loc": [ -72.607962, 42.162046 ], "pop": 23396, "state": "MA" }
{ "id": "01020", "city": "CHICOPEE", "loc": [ -72.576142, 42.176443 ], "pop": 31495, "state": "MA" }
{ "id": "01022", "city": "WESTOVER AFB", "loc": [ -72.558657, 42.196672 ], "pop": 1764, "state": "MA" }
{ "id": "01026", "city": "CUMMINGTON", "loc": [ -72.905767, 42.435296 ], "pop": 1484, "state": "MA" }
{ "id": "01027", "city": "MOUNT TOM", "loc": [ -72.67992099999999, 42.264319 ], "pop": 16864, "state": "MA" }
{ "id": "01028", "city": "EAST LONGMEADOW", "loc": [ -72.505565, 42.067203 ], "pop": 13367, "state": "MA" }
{ "id": "01030", "city": "FEEDING HILLS", "loc": [ -72.675077, 42.07182 ], "pop": 11985, "state": "MA" }
{ "id": "01031", "city": "GILBERTVILLE", "loc": [ -72.19858499999999, 42.332194 ], "pop": 2385, "state": "MA" }
{ "id": "01032", "city": "GOSHEN", "loc": [ -72.844097, 42.466734 ], "pop": 122, "state": "MA" }
```

# json format

## jsonlite for json data

<http://arxiv.org/abs/1403.2805>

```
library(jsonlite)
iris2=toJSON(iris)
head(iris2)
iris3=fromJSON(iris2)
head(iris3)
```

5.12 (Top Level) 2

Console

```
idth":2.3,"Species":"virginica"),{"Sepal.Length":6.7,
.5,"Species":"virginica"},{"Sepal.Length":6.7,"Sepal.
es":"virginica"},{"Sepal.Length":6.3,"Sepal.Width":2.
ginica"},{"Sepal.Length":6.5,"Sepal.Width":3,"Petal.Le
"Sepal.Length":6.2,"Sepal.Width":3.4,"Petal.Length":5.4
ength":5.9,"Sepal.Width":3,"Petal.Length":5.1,"Petal.k
> iris3=fromJSON(iris2)
> head(iris3)
```

|   | Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
|---|--------------|-------------|--------------|-------------|---------|
| 1 | 5.1          | 3.5         | 1.4          | 0.2         | setosa  |
| 2 | 4.9          | 3.0         | 1.4          | 0.2         | setosa  |
| 3 | 4.7          | 3.2         | 1.3          | 0.2         | setosa  |
| 4 | 4.6          | 3.1         | 1.5          | 0.2         | setosa  |
| 5 | 5.0          | 3.6         | 1.4          | 0.2         | setosa  |
| 6 | 5.4          | 3.9         | 1.7          | 0.4         | setosa  |

# Using APIs for data

CRAN Task View: Web Technologies and Services

<https://ropensci.org/>

Maintainers: Scott Chamberlain, Thomas Leeper, Patrick Main, Karthik Ram, Christopher Gandrud

Contact: [scott@ropensci.org](mailto:scott@ropensci.org)

Version: 2015-05-20

This task view contains information about using R to obtain and parse data from the web. The base version of R does not ship with many tools for interacting with the web. Thankfully, there are an increasingly large number of tools for interacting with the web. A list of available packages and functions is presented below, grouped by the type of activity. If you have any comments or suggestions for additions or improvements for this taskview, go to GitHub and [submit an issue](#), or make some changes and [submit a pull request](#). If you can't contribute on GitHub, [send Scott an email](#). If you have an issue with one of the packages discussed below, please contact the maintainer of that package. If you know of a web service, API, data source, or other online resource that is not yet supported by an R package, consider adding it to [the package development to do list on GitHub](#).

Tools for Working with the Web from R

## Fetching Data from the Web

- `download.file()` / `download.file()` is in base R and commonly used way to download a file. However, downloading files over HTTPS is not supported in R's internal method for `download.file()`. The `download()` function in the package [downloadr](#) wraps `download.file()`, and takes all the same arguments, but works for https across platforms.
- `webshot` / `webshot`: You can use `read.table()`, `read.csv()`, and friends to read a table directly from a URL, or after acquiring the csv file from the web via e.g., `getURL()` from RCurl. `read.csv()` works with http but not https, i.e., `read.csv("http://...")`, but not `read.csv("https://...")`. You can download a file first before reading the file in R, and you can use [downloadr](#) to download over https. `read.table()` and friends also have a `text` parameter so you can read a table if a table is encoded as a string with line breaks, etc.
- `RJSONIO` / `RJSON` is *javascript object notation*. There are three packages for reading and writing JSON: `RJSON`, `RJSONIO`, and `jsonlite`. `jsonlite` includes a different parser from `RJSONIO` called `yaml`. We recommend using `jsonlite`. Check out the paper describing `jsonlite` by James Oros <http://ropensci.org/pubs/1403.2807>.
- `XML` / `RXML` / `XML`: The package `XML` contains functions for parsing XML and HTML, and supports XPath for searching XML (think regex for strings). A helpful function to read data from one or more HTML tables is `readHTMLTable()`. `XML` also includes XPath parsing ability, see `applyXPath()` and `applyXPath2()`. The `XML2R` package is a collection of convenient functions for converting XML into data frames (development version [on GitHub](#)). An alternative to `XML` is `selectr`, which parses CSS Selectors and translates them to XPath 1.0 expressions. `XML` package is often used for parsing xml and html, but `selectr` translates CSS selectors to XPath, so you can use the CSS selectors instead of XPath. The [selectr/gadget:here-our-extension](#) can be used to identify page elements. `RHTMLForms` reads HTML documents and obtains a description of each of the forms it contains, along with the different elements and hidden fields. `scraper` provides additional tools for scraping data from HTML and XML documents.
- `rcurl`: `rcurl` scrapes html from web pages, and is designed to work with `magrittr` to make it easy to express common web scraping tasks.
- The `htcextract` package extract top level domains and subdomains from a host name. It's a part of a [Python library of the same name](#).
- `urltools`: Utility functions for developing web applications. Parses for `application/*-form-urlencoded` as well as `multipart/form-data`. [Source on GitHub](#)
- `urltools`: URL encoding, decoding, parsing, and parameter extraction. [Source on GitHub](#)
- The `rsync` package contains a `source_data()` command to load and cache plain-text data from a URL (either http or https). It also includes `source_to_cache()` for downloading/caching plain-text data from non-public Dropbox folders and `source_to_cache2()` for downloading/caching Excel files sheets.
- `sdmx` provides tools to read data and metadata documents exchanged through the Statistical Data and Metadata Exchange (SDMX) framework. The package currently focuses on the SDMX XML standard format (SDMX-ML). [project website \(GitHub\)](#).

## Curl, HTTP, FTP, HTML, XML, SOAP

- `RCurl`: A low level curl wrapper that allows one to compose general HTTP requests and provides convenient functions to fetch URLs, get/post forms, etc. and process the results returned by the Web server. This provides a great deal of control over the HTTP/FTP connection and the forms of the request while providing a higher-level interface than is available just using R socket connections. It also provide tools for Web authentication.



# ff package

<http://cran.r-project.org/web/packages/ff/index.html>

The ff package provides data structures that are stored on disk but behave (almost) as if they were in RAM by transparently mapping only a section (pagesize) in main memory - the effective virtual memory consumption per ff object.

<http://cran.r-project.org/web/packages/ffbase/index.html>

Basic (statistical) functionality for package ff

Example- <http://www.bnosac.be/index.php/blog/22-if-you-are-into-large-data-and-work-a-lot-package-ff>

```
> require(ffbase)
> hhp <- read.table.ffdf(file="/home/jan/Work/RForgeBNOSAC/github/RBelgium_HeritageHealthPrize/Data/Claims.csv",
FUN = "read.csv", na.strings = "")
```

Also see <http://cran.r-project.org/web/packages/bigmemory/index.html>

Create, store, access, and manipulate massive matrices. Matrices are allocated to shared memory and may use memory-mapped files. Packages biganalytics, bigtabulate, synchronicity, and bigalgebra provide advanced functionality

# RevoScaleR package

RevoScaleR has its own file format, XDF, which is able to rapidly access data by row or by column and to read some data sequentially. XDF file data is stored in the same binary format used in memory, which eliminates the need for conversion when it is brought into memory.

<http://www.revolutionanalytics.com/revolution-r-enterprise-scaler>

# rhdf5

This R/Bioconductor package provides an interface between HDF5 and R. HDF5's main features are the ability to store and access very large and/or complex datasets and a wide variety of metadata on mass storage (disk) through a completely portable file format.

<http://www.bioconductor.org/packages/release/bioc/html/rhdf5.html>

HDF5 is a data model, library, and file format for storing and managing data. It supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data. HDF5 is portable and is extensible, allowing applications to evolve in their use of HDF5.

<https://www.hdfgroup.org/HDF5/>

HDF5 simplifies the file structure to include only two major types of object:

- Datasets, which are multidimensional arrays of a homogeneous type
- Groups, which are container structures which can hold datasets and other groups



## HDF5 interface to R

Bioconductor version: Release (3.1)

This R/Bioconductor package provides an interface between HDF5 and R. HDF5's main features are the ability to store and access very large and/or complex datasets and a wide variety of metadata on mass storage (disk) through a completely portable file format. The rhdf5 package is thus suited for the exchange of large and/or complex datasets between R and other software packages, and for letting R applications work on datasets that are larger than the available RAM.

Author: Bernd Fischer, Gregoire Pau

Maintainer: Bernd Fischer <b.fischer@dkfz.de>

Citation (from within R, enter `citation("rhdf5")`):

Fischer B and Pau G. rhdf5: HDF5 interface to R. R package version 2.12.0.

### Installation

To install this package, start R and enter:

```
source("http://bioconductor.org/BiocLite.R")
biocLite("rhdf5")
```

### Documentation

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("rhdf5")
```

### Documentation =>

#### Bioconductor

- Package [page](#) and manuals.
- [Workflows](#) for learning and use.
- [Course and conference](#) material.
- [Videos](#).
- Community [resources](#) and [forums](#).

R / [CRAN](#) packages and [documentation](#)

### Support =>

Please read the [getting started](#). Post questions about Bioconductor to one of the following locations:

- [Support site](#) - for questions about Bioconductor packages.
- [BioC-devel](#) mailing list - for package developers.

## Functions Used in this lesson

- toJSON and fromJSON

# Packages

- data.table
- jsonlite
- rvest

# Revision

|                  |                    |
|------------------|--------------------|
| getwd            | fread vs read, csv |
| setwd            | Df[i,j]            |
| dir              | Df\$column         |
| ls               | str                |
| rm               | summary            |
| Install.packages | table              |
| library          | citation           |
|                  | help               |

# Revision

mean

std

median

length

Vector

data.frame

Indexing

class

nrow

ncol

head

tail



# Citations and References

M Dowle, T Short, S Lianoglou, A Srinivasan with contributions from R Saporta and E Antonyan (2014) data.table: Extension of data.frame. R package version 1.9.4. <http://CRAN.R-project.org/package=data.table>

Jeroen Ooms (2014). The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. arXiv:1403.2805 [stat.CO] URL <http://arxiv.org/abs/1403.2805>

Hadley Wickham (2015). rvest: Easily Harvest (Scrape) Web Pages. R package version 0.2.0. <http://CRAN.R-project.org/package=rvest>

