



COMPUTATIONAL FINANCE & RISK MANAGEMENT

UNIVERSITY *of* WASHINGTON

Department of Applied Mathematics

R Programming for Quantitative Finance


Guy Yollin


Applied Mathematics
University of Washington

Outline

- 1 R language overview and history
- 2 R language references
- 3 Short R Tutorial
- 4 The R help system
- 5 Web resources for R
- 6 IDE editors for R

Lecture references

- 
- J. Adler.
R in a Nutshell: A Desktop Quick Reference.
O'Reilly Media, 2010.
- Chapters 1-3

- 
- W. N. Venables and D. M. Smith.
An Introduction to R.
2013.
- Sections 1-3

Outline

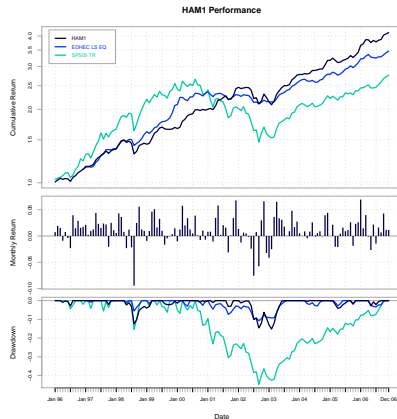
- 1 R language overview and history
- 2 R language references
- 3 Short R Tutorial
- 4 The R help system
- 5 Web resources for R
- 6 IDE editors for R

The R programming language

- R is a *language* and *environment* for statistical computing and graphics
- R is based on the *S language* originally developed by John Chambers and colleagues at AT&T Bell Labs in the late 1970s and early 1980s
- R (sometimes called “*GNU S*”) is free open source software licensed under the GNU general public license (GPL 2)
- R development was initiated by Robert Gentleman and Ross Ihaka at the University of Auckland, New Zealand in the 1990s
- R is formally known as The R Project for Statistical Computing
 - www.r-project.org

Strengths of the R programming language

- Data manipulation
- Data analysis
- Statistical modeling
- Data visualization



Plot from the PerformanceAnalytics package

S language implementations

R is the most recent and full-featured implementation of the S language

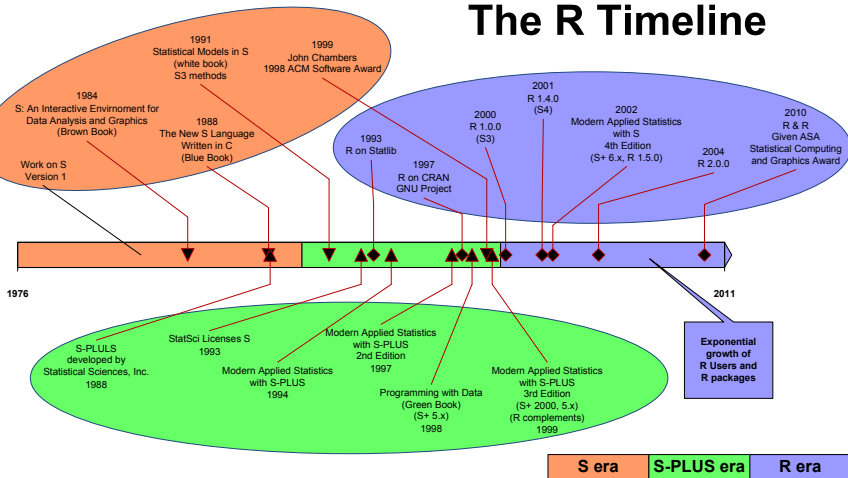
- Original S - AT & T Bell Labs
- S-PLUS (S plus a GUI)
 - Statistical Sciences, Inc.[†]
 - Mathsoft, Inc.
 - Insightful, Inc.
 - Tibco, Inc.
- R - The R Project for Statistical Computing



Figure from *The History of S and R*, John Chambers, 2006

[†]Founded by UW Professor Doug Martin, CompFin Program Director

The R Timeline



Recognition of software excellence

Association for Computing Machinery

John Chambers received the 1998 ACM Software System Award

*Dr. Chambers' work
will forever alter the
way people analyze,
visualize, and
manipulate data*

American Statistical Association

Robert Gentleman and Ross Ihaka received the 2009 ASA Statistical Computing and Graphics Award

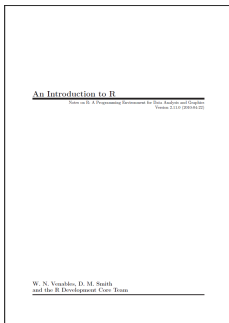
*In recognition for their
work in initiating the R
Project for Statistical
Computing*

Outline

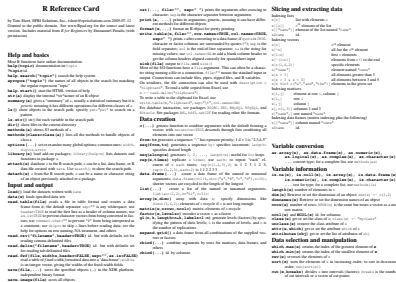
- 1 R language overview and history
- 2 R language references
- 3 Short R Tutorial
- 4 The R help system
- 5 Web resources for R
- 6 IDE editors for R

Essential web resources

- An Introduction to R
 - W.N. Venables, D.M. Smith
 - R Development Core Team



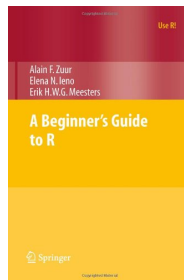
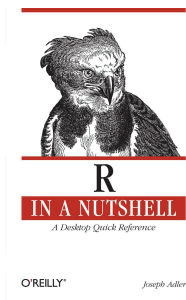
- R Reference Card 2.0
 - Baggott & Short



Definitely obtain these PDF files from the R homepage or a CRAN mirror

Introductory texts

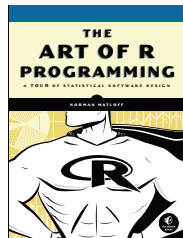
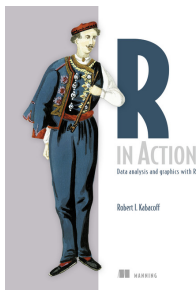
- R in a Nutshell: A Desktop Quick Reference
 - Joseph Adler
 - O'Reilly Media, 2009
- A Beginner's Guide to R
 - Zuur, Ieno, Meesters
 - Springer, 2009



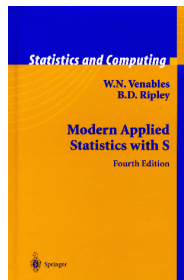
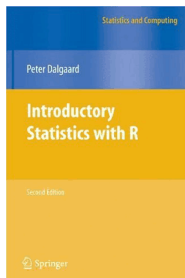
Both texts available online through UW library

Introductory texts

- R in Action
 - Robert Kabacoff
 - Manning Publications, 2011
- The Art of R Programming
 - Norman Matloff
 - No Starch Press, 2011



- Introductory Statistics with R
2nd Edition
 - P. Dalgaard
 - Springer, 2008
- Modern Applied Statistics
with S, 4th Edition
 - Venables and Ripley
 - Springer, 2002



Experience with other statistical computing languages

For those with experience in MATLAB, David Hiebeler has created a MATLAB/R cross reference document:

- <http://www.math.umaine.edu/~hiebler/comp/matlabR.pdf>

For those with experience in SAS, SPSS, or Stata, Robert Muenchen has written R books for this audience:

- <http://r4stats.com>

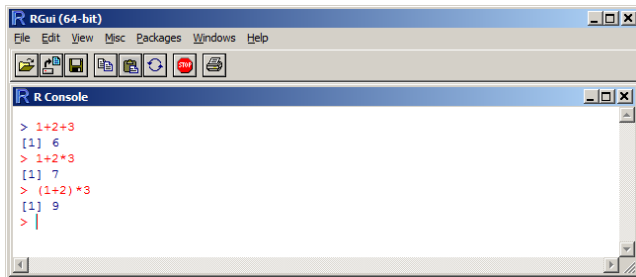
Outline

- 1 R language overview and history
- 2 R language references
- 3 Short R Tutorial**
- 4 The R help system
- 5 Web resources for R
- 6 IDE editors for R

Interacting with R

R is an *interpreted language*[†]

- An R interpreter must be running in order to evaluate R commands or execute R scripts
 - RGui which includes an R Console window
 - RStudio which includes an R Console window

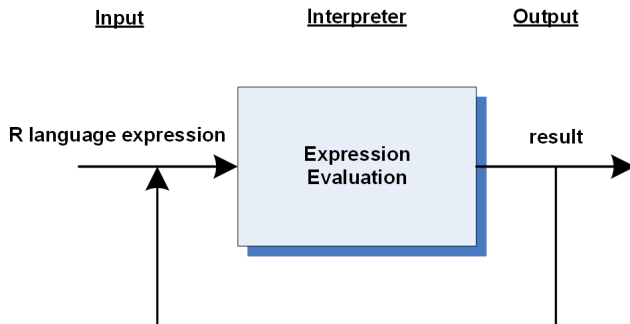


[†]http://en.wikipedia.org/wiki/Interpreted_language

R expression evaluation

R expressions are processed via R's Read-eval-print loop [†]:

The Read-Evaluate-Print Loop (REPL) for R

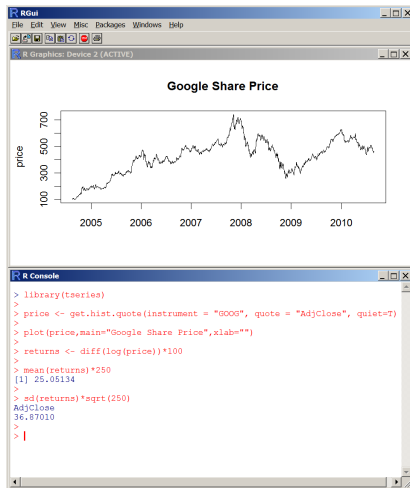


[†]http://en.wikipedia.org/wiki/Read-eval-print_loop

Interacting with the RGui

The RGui is an *interactive* command driven *environment*:

- Type R commands (expressions) into the R Console
- Copy/Paste multiple R commands into the R Console
- Source an R script
 - An R script is simply a text file of multiple R commands

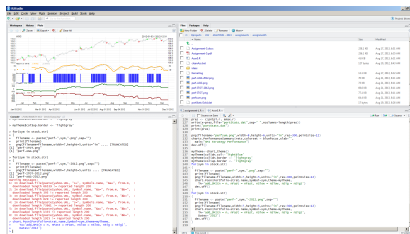


Commands entered interactively into the R console

Interacting with RStudio

The RStudio is an *Integrated Development Environment (IDE)* R:

- Embedded R Console
 - RStudio runs an R interpreter automatically
- Program editor for R
- Plot window
- File browser
- Integrated version control
- R debugger



RStudio includes an embedded R Console

Calling functions

R makes extensive use of functions[†]

- Functions can be defined to take zero or more arguments
- Functions typically return a value
 - a return value is not required
- Functions are called by name with any arguments enclosed in parentheses
 - even if the function has no arguments the parentheses are required

```
sin(pi/2)

## [1] 1

print("Hello, world")

## [1] "Hello, world"

abs(-8)

## [1] 8

cos(2*sqrt(2))

## [1] -0.95136313

date()

## [1] "Sun Aug 31 17:08:42 2014"
```

[†]http://en.wikipedia.org/wiki/Functional_programming

Assigning values to variables

Like other programming languages, values can be stored in variables

- Variables are typically assigned in 1 of 3 ways:
 - assignment operator: `<-`
 - assignment function: `assign`
 - equal sign: `=`
 - must be used to assign arguments in a function call

```
y <- 5
y

## [1] 5

assign("e",2.7183)
e

## [1] 2.7183

s = sqrt(2)
s

## [1] 1.4142136

r <- rnorm(n=2)
r

## [1] -1.0067110533 -0.0020828847

s*e+y

## [1] 8.8442567
```

Object orientation in R

Everything in R is an Object[†]

- Use functions `ls` and `objects` to list all objects in the current workspace

```
x <- c(3.1416,2.7183)
m <- matrix(rnorm(9),nrow=3)
tab <- data.frame(store=c("downtown","eastside","airport"),sales=c(32,17,24))
cities <- c("Seattle","Portland","San Francisco")
ls()

## [1] "cities"      "e"           "filename"    "m"           "r"           "s"
## [7] "tab"        "x"           "y"
```

[†]http://en.wikipedia.org/wiki/Object-oriented_programming

Object classes

All R objects have a *class*

The class of an object determines what it can do and what you can do with it

- Use function `class` to display an object's class
- There are many R classes; basic classes are:
 - numeric
 - character
 - data.frame
 - matrix

```
m

##           [,1]           [,2]           [,3]
## [1,]  0.374352397  0.586864810 -0.73778598
## [2,] -0.071532765 -0.262264339 -0.19904931
## [3,]  0.790144078  0.012603635  1.96472235

class(m)

## [1] "matrix"

tab

##      store sales
## 1 downtown    32
## 2 eastside    17
## 3  airport    24

class(tab)

## [1] "data.frame"
```


Vectors

R is a vector/matrix programming language (also known as an array programming language[†])

- vectors can easily be created with `c`, the combine function
- most places where single value can be supplied, a vector can be supplied and R will perform a vectorized operation

```
my.vector <- c(2, 4, 3, 7, 10)
my.vector
```

```
## [1] 2 4 3 7 10
```

```
my.vector^2
```

```
## [1] 4 16 9 49 100
```

```
sqrt(my.vector)
```

```
## [1] 1.4142136 2.0000000 1.7320508 2.6457513 3.1622777
```

[†]http://en.wikipedia.org/wiki/Array_programming

Creating vectors with the c function

```
constants <- c(3.1416,2.7183,1.4142,1.6180)
constants

## [1] 3.1416 2.7183 1.4142 1.6180

my.labels <- c("pi","euler","sqrt2","golden")
my.labels

## [1] "pi"      "euler"   "sqrt2"   "golden"

names(constants) <- my.labels
constants

##      pi  euler  sqrt2  golden
## 3.1416 2.7183 1.4142 1.6180
```

- The [1] in the above output is labeling the first element of the vector
- The c function can be used to create character vectors, numeric vectors, as well as other types of vectors

Indexing vectors

Vectors indices are placed with square brackets: `[]`

Vectors can be indexed in any of the following ways:

- vector of positive integers
- vector of negative integers
- vector of named items
- logical vector

```
constants[c(1,3,4)]

##      pi  sqrt2 golden
## 3.1416 1.4142 1.6180

constants[c(-1,-2)]

##      sqrt2 golden
## 1.4142 1.6180

constants[c("pi","golden")]

##      pi golden
## 3.1416 1.6180

constants > 2

##      pi  euler  sqrt2 golden
##   TRUE   TRUE  FALSE  FALSE

constants[constants > 2]

##      pi  euler
## 3.1416 2.7183
```

Creating integer sequences with the a:b operator

The sequence operator will generate a vector of integers between a and b
Sequences of this type are particularly useful for indexing vectors, matrices, data.frames etc.

```
1:5

## [1] 1 2 3 4 5

-(1:4)

## [1] -1 -2 -3 -4

letters[1:15]

## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o"

letters[16:26]

## [1] "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

letters[-(1:15)]

## [1] "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"
```

Comparing vector and non-vector computing

```
# vectorized operation
# taking the log of each element in a vector
x <- c(97.87,96.18,95,86.39,88.18,90.8,86.06,82.27,83.32,85.3,83.25,82.13,78.54)
log(x)

## [1] 4.5836401 4.5662214 4.5538769 4.4588719 4.4793802 4.5086593 4.4550447
## [8] 4.4100065 4.4226886 4.4461745 4.4218481 4.4083034 4.3636080

# non-vectorized computation
# taking the log of each element in a vector
n <- length(x)
y <- rep(0,n)
for( i in 1:n )
  y[i] <- log(x[i])
y

## [1] 4.5836401 4.5662214 4.5538769 4.4588719 4.4793802 4.5086593 4.4550447
## [8] 4.4100065 4.4226886 4.4461745 4.4218481 4.4083034 4.3636080
```

Comparing vector and non-vector computing

```
# vectorized operation  
# taking the log of each element in a matrix  
x <- matrix(c(2,9,4,7,5,3,6,1,8),nrow=3)  
x^2
```

```
##      [,1] [,2] [,3]  
## [1,]    4  49  36  
## [2,]   81  25   1  
## [3,]   16   9  64
```

```
# non-vectorized computation  
# taking the log of each element in a matrix  
y <- x  
for( i in 1:nrow(x) )  
  for( j in 1:ncol(x) )  
    y[i,j] <- x[i,j]^2  
y
```

```
##      [,1] [,2] [,3]  
## [1,]    4  49  36  
## [2,]   81  25   1  
## [3,]   16   9  64
```

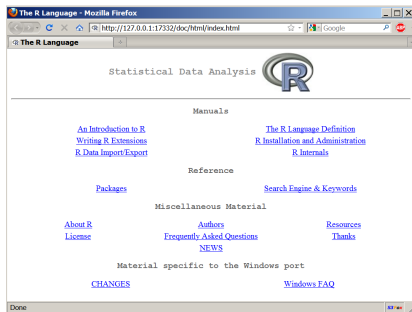
Outline

- 1 R language overview and history
- 2 R language references
- 3 Short R Tutorial
- 4 The R help system**
- 5 Web resources for R
- 6 IDE editors for R

The HTML help system

R has a comprehensive Html help facility

- Run the `help.start` function
- R GUI menu item
Help|Html help



```
help.start()
```

```
## If nothing happens, you should open
```

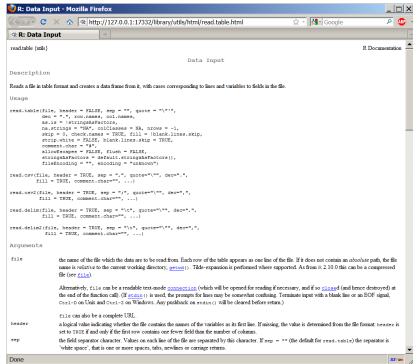
```
## 'http://127.0.0.1:28913/doc/html/index.html' yourself
```


The help function

Obtain help on a particular topic via the `help` function

- `help(topic)`
- `?topic`

```
help(read.table)
```

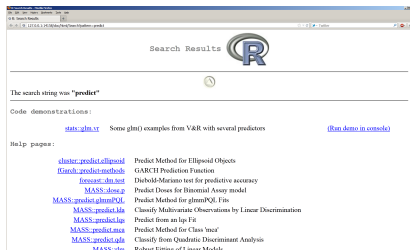


The help.search function

Search help for a particular topic via the `help.search` function

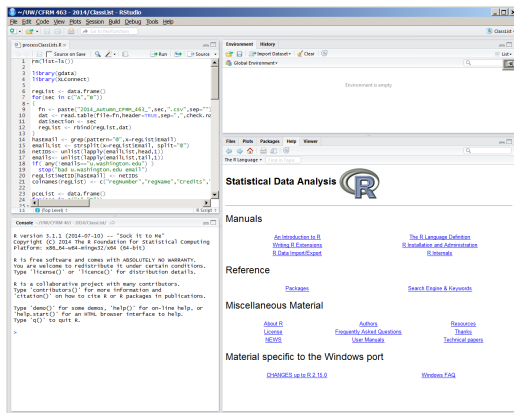
- `help.search(topic)`
- `??topic`

```
??predict
```



Help tab in RStudio

RStudio incorporates a dedicated help tab which facilitates accessing the R Html help system

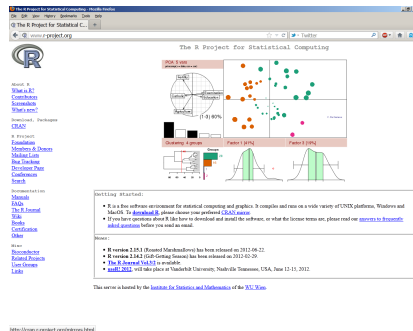


Outline

- 1 R language overview and history
- 2 R language references
- 3 Short R Tutorial
- 4 The R help system
- 5 Web resources for R**
- 6 IDE editors for R

<http://www.r-project.org>

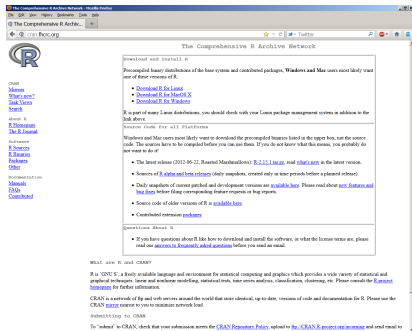
- List of CRAN mirror sites
- Manuals
- FAQs
- Site search
- Mailing lists
- Links



CRAN - Comprehensive R Archive Network

<http://cran.fhcrc.org>

- CRAN Mirrors
 - About 45 countries
 - About 100 sites worldwide
 - About 15 sites in US
- R Binaries
- R Packages
 - 5800+ packages
- R Sources
- Task Views



use your closest CRAN mirror site

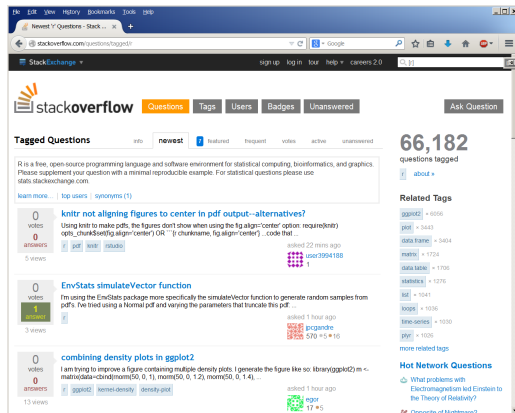
CRAN Task Views

Organizes the 5800+ R packages by application

- Finance
- Time Series
- Econometrics
- Optimization
- Machine Learning



Stackoverflow has become the primary resource for help with R



<http://stackoverflow.com/>

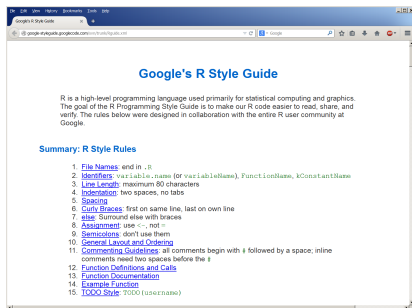
- Nerve center of the R finance community
- Daily must read
- Exclusively for Finance-specific questions, not general R questions

The screenshot shows a web browser window titled "R-SIG-Finance Info Page - Mozilla Firefox". The address bar displays the URL "https://stat.ethz.ch/mailman/listinfo/r-sig-finance". The page content includes a header "R-SIG-Finance -- Special Interest Group for 'R in Finance'", a language selector for "English (USA)", and sections for "About R-SIG-Finance", "Using R-SIG-Finance", and "Subscribing to R-SIG-Finance". The "Subscribing" section contains a form with fields for "Your email address:", "Your name (optional):" (with a dropdown menu showing "Guy Yollin"), "Pick a password:", "Reenter password to confirm:", "Which language do you prefer to display your messages?" (with a dropdown menu showing "English (USA)"), and "Would you like to receive list mail batched in a daily digest?" (with radio buttons for "No" and "Yes", where "Yes" is selected). A "Subscribe" button is at the bottom of the form.

<https://stat.ethz.ch/mailman/listinfo/r-sig-finance>

Google's R Style Guide

- 1 Naming convention
- 2 Coding Syntax
- 3 Program Organization



<http://google-styleguide.googlecode.com/svn/trunk/google-r-style.html>

<http://www.statmethods.net>

Introductory R Lessons

- R Interface
- Data Input
- Data Management
- Basic Statistics
- Advanced Statistics
- Basic Graphs
- Advanced Graphs



Site maintained by Robert Kabacoff, author of R in Action

Site of Earl Glynn of Stowers Institute for Medical Research

- R Graphics and other useful information
 - R Color Chart
 - Using Color in R (great presentation)
 - Plot area, margins, multiple figures
 - Mixture models
 - Distance measures and clustering
 - Using Windows Explorer to Start R with Specified Working Directory (under tech notes)

<http://research.stowers-institute.org/efg/R/index.htm>

Programming in R

Online R programming manual from UC Riverside:

- R Basics
- Finding Help
- Code Editors for R
- Control Structures
- Functions
- Object Oriented Programming
- Building R Packages

<http://manuals.bioinformatics.ucr.edu/home/programming-in-r>

Other useful R sites

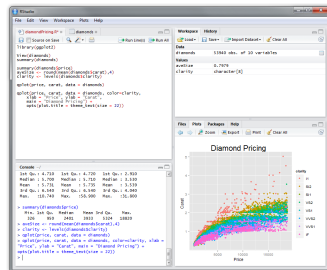
- | | |
|-----------------|---|
| R Bloggers | Aggregation of about 550 R blogs <ul style="list-style-type: none">• http://www.r-bloggers.com |
| R Site Search | Search R function help, vignettes, R-help <ul style="list-style-type: none">• http://finzi.psych.upenn.edu/search.html |
| R Seek | R specific search site <ul style="list-style-type: none">• http://www.rseek.org/ |
| Revolution Blog | Blog from David Smith of Revolution <ul style="list-style-type: none">• http://blog.revolutionanalytics.com |
| Inside-R | R community site by Revolution Analytics <ul style="list-style-type: none">• http://www.inside-r.org |

Outline

- 1 R language overview and history
- 2 R language references
- 3 Short R Tutorial
- 4 The R help system
- 5 Web resources for R
- 6 IDE editors for R**

RStudio is a fully-featured open-source IDE for R

- R language highlighting
- Paste/Source to R console
- object explorer
- tabbed graphics window
- integrated version control
- 1-click knitr/Sweave compilation



RStudio also provides a server-based version (R running in the cloud)

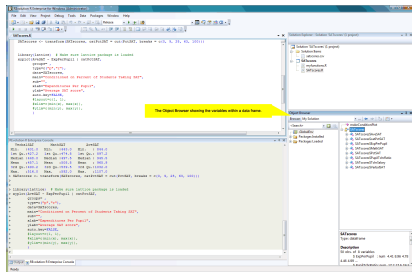
Revolution R Enterprise Visual Development Environment

Revolution Analytics is a company that sells a commercial distribution of R including a desktop IDE

Revolution R Enterprise is *free* to academic users

- R language highlighting
- Paste/Source code to R
- Source code debugger
- object explorer
- runs R in SDI mode

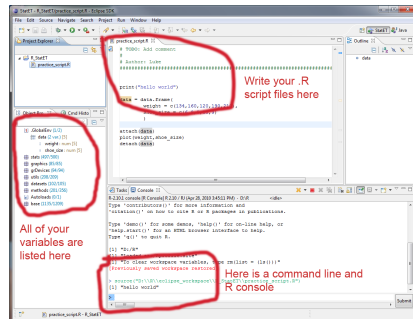
<http://www.revolutionanalytics.com>



StatET - An Eclipse Plug-In for R

StatET is a plug-in for the open-source Eclipse development environment

- R language highlighting
- Paste/Source code to R
- Source code debugger
- 1-click Sweave compilation
- Supports R in SDI mode
- Excellent documentation by Longhow Lam

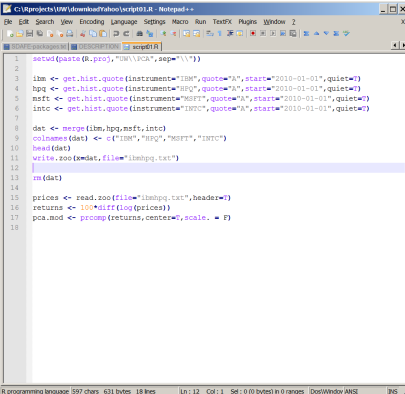


<http://www.walware.de/goto/statet>

Notepad++ and NpptoR

NpptoR is an automation widget (based on AutoHotkey) which allows the very useful program editor Notepad++ to interact with R

- R language highlighting
- Paste/Source code to R
- Supports R in SDI mode

A screenshot of the Notepad++ application window. The title bar reads "C:\Rprojects\IOW\download\Yahoo\script01.R - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, TextFX, Plugins, Window, and Help. The toolbar contains various icons for file operations and editing. The status bar at the bottom indicates "R programming language 597 chars 631 bytes 18 lines | Ln: 12 Col: 1 Sel: 0 (0 bytes) in 0 ranges | Doc/Window ANSI | RNS". The main text area contains the following R code:

```
1 setwd(paste(R.proj, "IOW\PCA", sep="\\"))
2
3 libm <- get.hist.quote(instrument="IBM", quote="A", start="2010-01-01", quiet=F)
4 hbpq <- get.hist.quote(instrument="HPQ", quote="A", start="2010-01-01", quiet=F)
5 msft <- get.hist.quote(instrument="MSFT", quote="A", start="2010-01-01", quiet=F)
6 intc <- get.hist.quote(instrument="INTC", quote="A", start="2010-01-01", quiet=F)
7
8 dat <- merge(libm, hbpq, msft, intc)
9 colnames(dat) <- c("IBM", "HPQ", "MSFT", "INTC")
10 head(dat)
11 write.zoo(x=dat, file="ibmhpq.txt")
12
13 rm(dat)
14
15 prices <- read.zoo(file="ibmhpq.txt", header=F)
16 returns <- 100*diff(log(prices))
17 pca.mod <- prcomp(returns, center=F, scale. = F)
18
```

<http://notepad-plus-plus.org>

<http://sourceforge.net/projects/npptor>

W COMPUTATIONAL FINANCE & RISK MANAGEMENT
UNIVERSITY *of* WASHINGTON
Department of Applied Mathematics

`http://depts.washington.edu/compfin`