RStudio is good for you!

Julyan Arbel (CREST - INSEE) - Statisfaction

Semin-R

7 February 2012

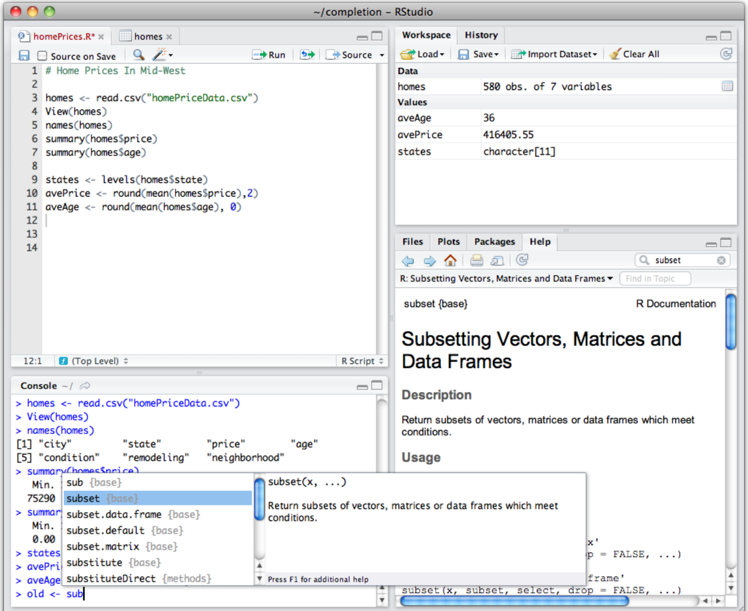
Useful links

- RStudio (download, blog)
- R-bloggers
- Statisfaction on Google + and Facebook

- RStudio is good for you!

- Running R on an iPhone/iPad with RStudio

Multi platform IDE



```
# Home Prices In Mid-West

homes <- read.csv("homePriceData.csv")
View(homes)
names(homes)
summary(homes$price)
summary(homes$age)

states <- levels(homes$state)
avePrice <- round(mean(homes$price),2)
aveAge <- round(mean(homes$age), 0)
```

Data	
homes	580 obs. of 7 variables
Values	
aveAge	36
avePrice	416405.55
states	character[11]

```
> homes <- read.csv("homePriceData.csv")
> View(homes)
> names(homes)
[1] "city"         "state"        "price"        "age"
[5] "condition"    "remodeling"   "neighborhood"
> summary(homes$price)
   Min.
  75290
> summar
   Min.
   0.00
> states
> avePri
> aveAge
> old <- sub
```

```
sub {base}
subset {base}
subset.data.frame {base}
subset.default {base}
subset.matrix {base}
substitute {base}
substituteDirect {methods}
```

```
subset(x, ...)

Return subsets of vectors, matrices or data frames which meet
conditions.
```

subset {base} R Documentation

Subsetting Vectors, Matrices and Data Frames

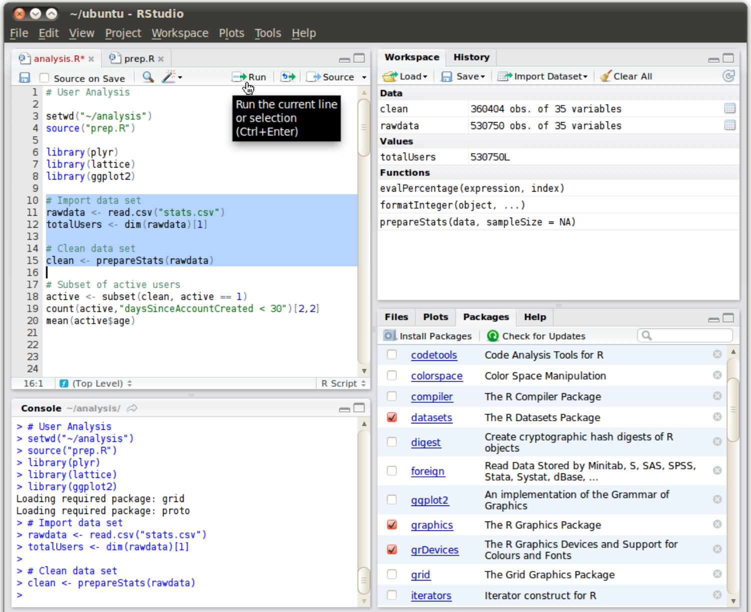
Description

Return subsets of vectors, matrices or data frames which meet conditions.

Usage

```
x'
p = FALSE, ...)
frame'
subset(x, subset, select, drop = FALSE, ...)
```

Multi platform IDE


~/ubuntu - RStudio
File Edit View Project Workspace Plots Tools Help
analysis.R* prep.R
Source on Save
Run
Source
16:1
(Top Level)
R Script
Workspace History
Load Save Import Dataset Clear All
Files Plots Packages Help
Install Packages Check for Updates
Console ~/analysis/

```
# User Analysis

setwd("~/analysis")
source("prep.R")

library(plyr)
library(lattice)
library(ggplot2)

# Import data set
rawdata <- read.csv("stats.csv")
totalUsers <- dim(rawdata)[1]

# Clean data set
clean <- prepareStats(rawdata)

# Subset of active users
active <- subset(clean, active == 1)
count(active,"daysSinceAccountCreated < 30")[2,2]
mean(active$age)
```

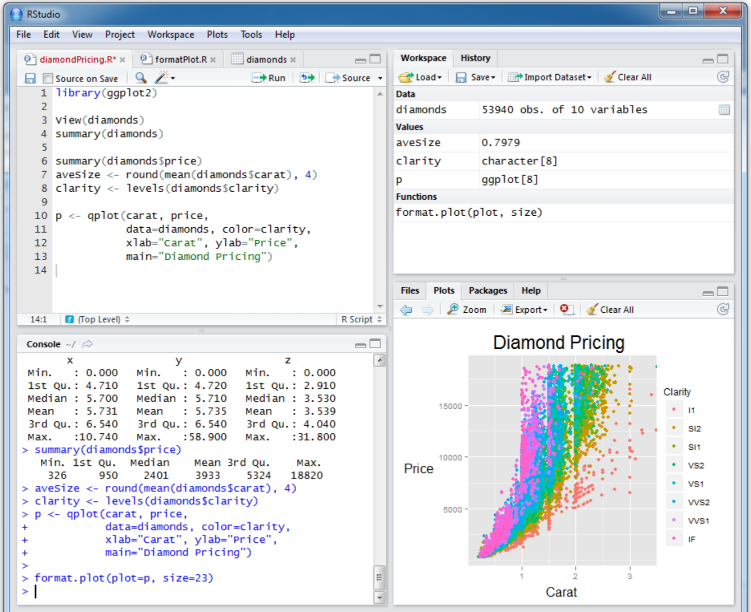
Run the current line or selection (Ctrl+Enter)

Data	
clean	360404 obs. of 35 variables
rawdata	530750 obs. of 35 variables
Values	
totalUsers	530750L
Functions	
evalPercentage(expression, index)	
formatInteger(object, ...)	
prepareStats(data, sampleSize = NA)	

☐	codetools	Code Analysis Tools for R
☐	colorspace	Color Space Manipulation
☐	compiler	The R Compiler Package
✓	datasets	The R Datasets Package
☐	digest	Create cryptographic hash digests of R objects
☐	foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...
☐	ggplot2	An implementation of the Grammar of Graphics
✓	graphics	The R Graphics Package
✓	grDevices	The R Graphics Devices and Support for Colours and Fonts
☐	grid	The Grid Graphics Package
☐	iterators	Iterator construct for R

```
> # User Analysis
> setwd("~/analysis")
> source("prep.R")
> library(plyr)
> library(lattice)
> library(ggplot2)
Loading required package: grid
Loading required package: proto
> # Import data set
> rawdata <- read.csv("stats.csv")
> totalUsers <- dim(rawdata)[1]
>
> # Clean data set
> clean <- prepareStats(rawdata)
>
```

Multi platform IDE


RStudio
File Edit View Project Workspace Plots Tools Help
diamondPricing.R* formatPlot.R diamonds
Source on Save Run Source
14:1 (Top Level) R Script
Console ~/
Workspace History
Load Save Import Dataset Clear All
Files Plots Packages Help
Zoom Export Clear All
Diamond Pricing
Price
Carat
15000
10000
5000
1
2
3
Clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF

```r
library(ggplot2)

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")
```

Data

diamonds	53940 obs. of 10 variables

Values

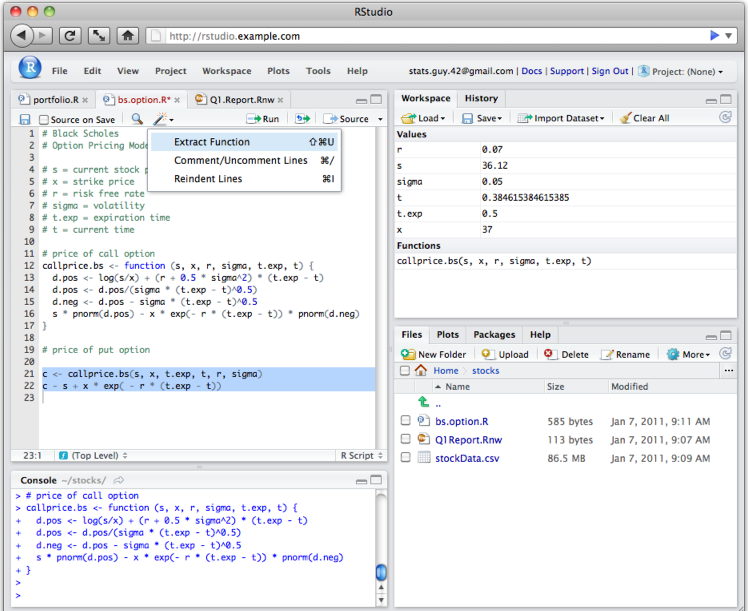
aveSize	0.7979
clarity	character[8]
p	ggplot[8]

Functions

format.plot(plot, size)

```
        x                 y                 z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(plot=p, size=23)
>
```

RStudio over the web


RStudio
http://rstudio.example.com
File Edit View Project Workspace Plots Tools Help
stats.guy.42@gmail.com | Docs | Support | Sign Out | Project: (None)
portfolio.R bs.option.R* Q1.Report.Rnw
Source on Save
Run
Source
Extract Function ⇧⌘U
Comment/Uncomment Lines ⌘/
Reindent Lines ⌘I
23:1
(Top Level)
R Script
Console ~/stocks/
Workspace History
Load Save Import Dataset Clear All
Files Plots Packages Help
New Folder Upload Delete Rename More
Home › stocks

```
# Black Scholes
# Option Pricing Mode

# s = current stock p
# x = strike price
# r = risk free rate
# sigma = volatility
# t.exp = expiration time
# t = current time

# price of call option
callprice.bs <- function (s, x, r, sigma, t.exp, t) {
  d.pos <- log(s/x) + (r + 0.5 * sigma^2) * (t.exp - t)
  d.pos <- d.pos/(sigma * (t.exp - t)^0.5)
  d.neg <- d.pos - sigma * (t.exp - t)^0.5
  s * pnorm(d.pos) - x * exp(- r * (t.exp - t)) * pnorm(d.neg)
}

# price of put option

c <- callprice.bs(s, x, t.exp, t, r, sigma)
c - s + x * exp( - r * (t.exp - t))

```

```
> # price of call option
> callprice.bs <- function (s, x, r, sigma, t.exp, t) {
+   d.pos <- log(s/x) + (r + 0.5 * sigma^2) * (t.exp - t)
+   d.pos <- d.pos/(sigma * (t.exp - t)^0.5)
+   d.neg <- d.pos - sigma * (t.exp - t)^0.5
+   s * pnorm(d.pos) - x * exp(- r * (t.exp - t)) * pnorm(d.neg)
+ }
>
>
```

Values	
r	0.07
s	36.12
sigma	0.05
t	0.384615384615385
t.exp	0.5
x	37
Functions	
callprice.bs(s, x, r, sigma, t.exp, t)	

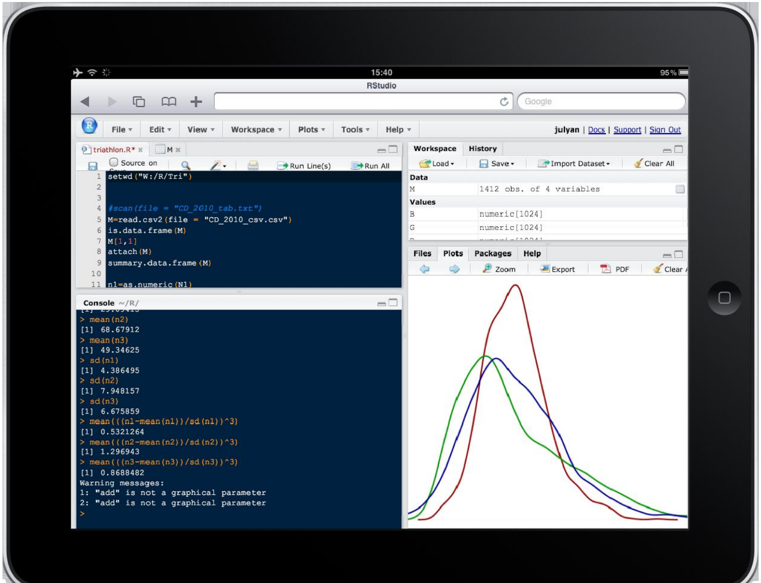
Name	Size	Modified
..		
bs.option.R	585 bytes	Jan 7, 2011, 9:11 AM
Q1Report.Rnw	113 bytes	Jan 7, 2011, 9:07 AM
stockData.csv	86.5 MB	Jan 7, 2011, 9:09 AM

RStudio over the web

Need to create a server

- on Linux
- or on a Mac

R accessed from a mobile/tablet

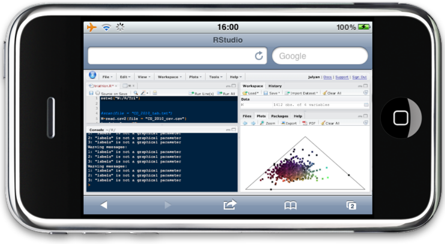


```
setwd("W:/R/Tri")


#scan(file = "CD_2010_tab.txt")
M=read.csv2(file = "CD_2010_csv.csv")
is.data.frame(M)
M[1,1]
attach(M)
summary.data.frame(M)

n1=as.numeric(N1)
```

```
Data
M        1412 obs. of 4 variables
Values
B        numeric[1024]
G        numeric[1024]
```

```
> mean(n2)
[1] 68.67912
> mean(n3)
[1] 49.34625
> sd(n1)
[1] 4.386495
> sd(n2)
[1] 7.948157
> sd(n3)
[1] 6.675859
> mean(((n1-mean(n1))/sd(n1))^3)
[1] 0.5321264
> mean(((n2-mean(n2))/sd(n2))^3)
[1] 1.296943
> mean(((n3-mean(n3))/sd(n3))^3)
[1] 0.8688482
Warning messages:
1: "add" is not a graphical parameter
2: "add" is not a graphical parameter
>
```

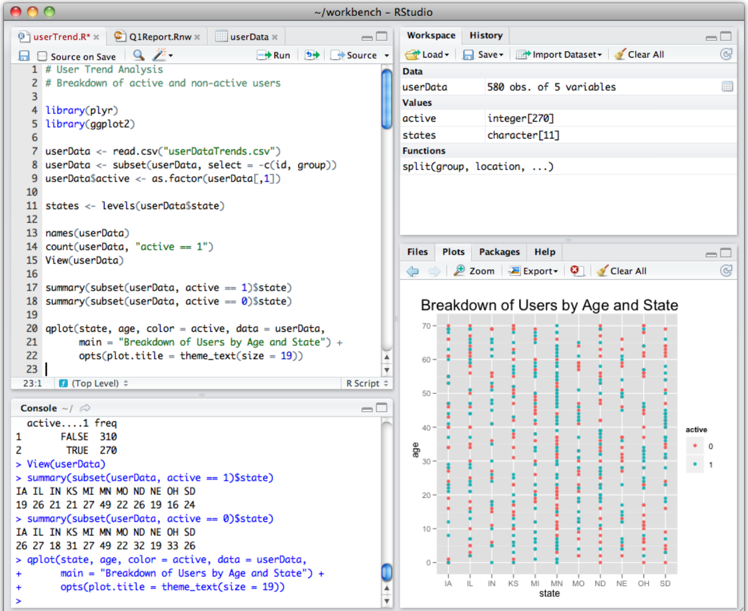
R accessed from a mobile/tablet



```
setwd("W:/R/Tri")


#scan(file = "CD_2010_tab.txt")
M=read.csv2(file = "CD_2010_csv.csv")
```

```
2: "labels" is not a graphical parameter
3: "labels" is not a graphical parameter
Warning messages:
1: "labels" is not a graphical parameter
2: "labels" is not a graphical parameter
3: "labels" is not a graphical parameter
Warning messages:
1: "labels" is not a graphical parameter
2: "labels" is not a graphical parameter
3: "labels" is not a graphical parameter
>
```

Source, Console, Workspace, and Plots

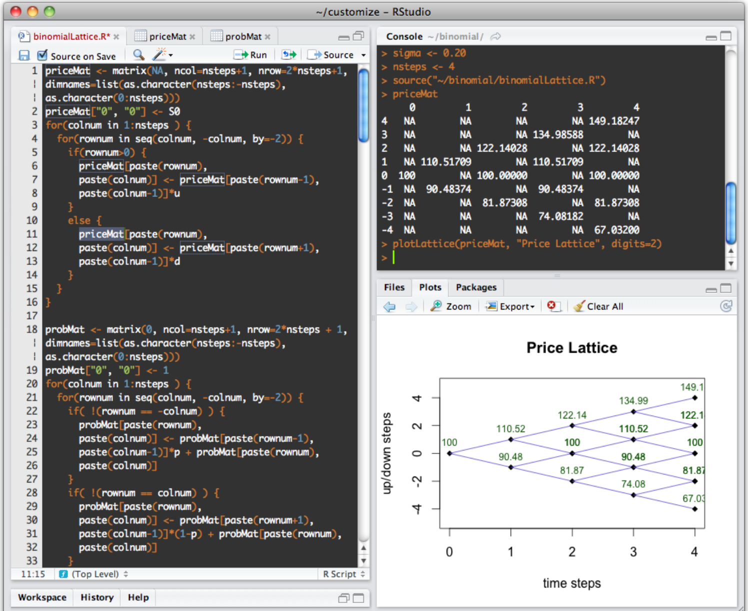


```r
# User Trend Analysis
# Breakdown of active and non-active users

library(plyr)
library(ggplot2)

userData <- read.csv("userDataTrends.csv")
userData <- subset(userData, select = -c(id, group))
userData$active <- as.factor(userData[,1])

states <- levels(userData$state)

names(userData)
count(userData, "active == 1")
View(userData)

summary(subset(userData, active == 1)$state)
summary(subset(userData, active == 0)$state)

qplot(state, age, color = active, data = userData,
      main = "Breakdown of Users by Age and State") +
      opts(plot.title = theme_text(size = 19))

```

```
  active....1 freq
1       FALSE  310
2        TRUE  270
> View(userData)
> summary(subset(userData, active == 1)$state)
IA IL IN KS MI MN MO ND NE OH SD
19 26 21 21 27 49 22 26 19 16 24
> summary(subset(userData, active == 0)$state)
IA IL IN KS MI MN MO ND NE OH SD
26 27 18 31 27 49 22 32 19 33 26
> qplot(state, age, color = active, data = userData,
+       main = "Breakdown of Users by Age and State") +
+       opts(plot.title = theme_text(size = 19))
>
```

Data	
userData	580 obs. of 5 variables
Values	
active	integer[270]
states	character[11]
Functions	
split(group, location, ...)	

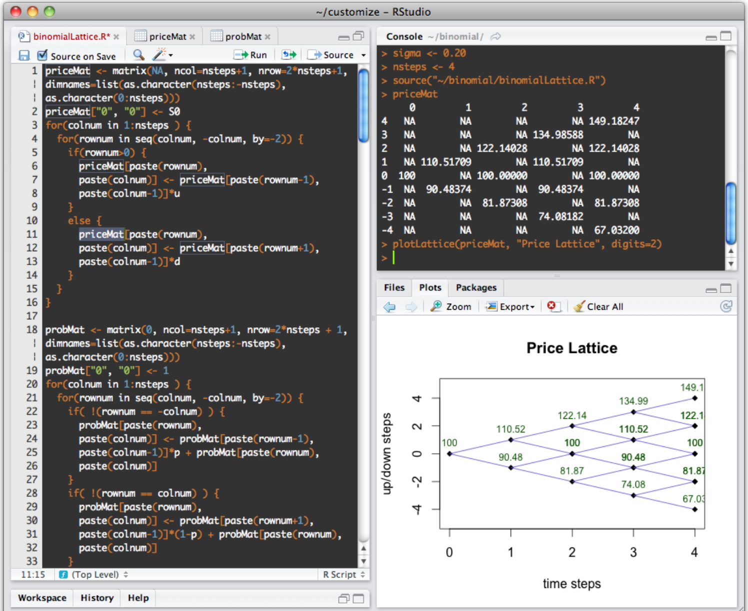
Custom Theme and Layout



```r
priceMat <- matrix(NA, ncol=nsteps+1, nrow=2*nsteps+1,
dimnames=list(as.character(nsteps:-nsteps),
as.character(0:nsteps)))
priceMat["0", "0"] <- S0
for(colnum in 1:nsteps ) {
  for(rownum in seq(colnum, -colnum, by=-2)) {
    if(rownum>0) {
      priceMat[paste(rownum),
      paste(colnum)] <- priceMat[paste(rownum-1),
      paste(colnum-1)]*u
    }
    else {
      priceMat[paste(rownum),
      paste(colnum)] <- priceMat[paste(rownum+1),
      paste(colnum-1)]*d
    }
  }
}

probMat <- matrix(0, ncol=nsteps+1, nrow=2*nsteps + 1,
dimnames=list(as.character(nsteps:-nsteps),
as.character(0:nsteps)))
probMat["0", "0"] <- 1
for(colnum in 1:nsteps ) {
  for(rownum in seq(colnum, -colnum, by=-2)) {
    if( !(rownum == -colnum) ) {
      probMat[paste(rownum),
      paste(colnum)] <- probMat[paste(rownum-1),
      paste(colnum-1)]*p + probMat[paste(rownum),
      paste(colnum)]
    }
    if( !(rownum == colnum) ) {
      probMat[paste(rownum),
      paste(colnum)] <- probMat[paste(rownum+1),
      paste(colnum-1)]*(1-p) + probMat[paste(rownum),
      paste(colnum)]
    }
```

```
> sigma <- 0.20
> nsteps <- 4
> source("~/binomial/binomialLattice.R")
> priceMat
     0         1         2         3         4
4   NA        NA        NA        NA 149.18247
3   NA        NA        NA 134.98588        NA
2   NA        NA 122.14028        NA 122.14028
1   NA 110.51709        NA 110.51709        NA
0  100        NA 100.00000        NA 100.00000
-1  NA  90.48374        NA  90.48374        NA
-2  NA        NA  81.87308        NA  81.87308
-3  NA        NA        NA  74.08182        NA
-4  NA        NA        NA        NA  67.03200
> plotLattice(priceMat, "Price Lattice", digits=2)
>
```

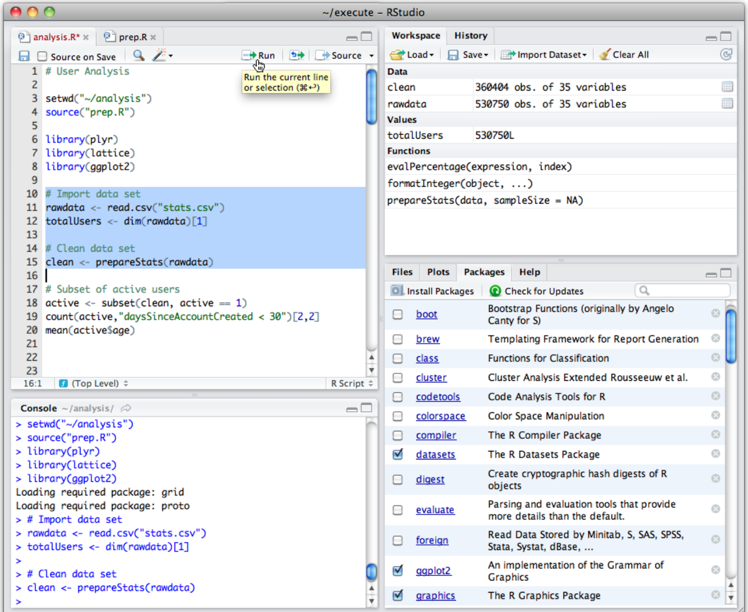
Code Completion



```
priceMat <- matrix(NA, ncol=nsteps+1, nrow=2*nsteps+1,
dimnames=list(as.character(nsteps:-nsteps),
as.character(0:nsteps)))
priceMat["0", "0"] <- S0
for(colnum in 1:nsteps ) {
  for(rownum in seq(colnum, -colnum, by=-2)) {
    if(rownum>0) {
      priceMat[paste(rownum),
      paste(colnum)] <- priceMat[paste(rownum-1),
      paste(colnum-1)]*u
    }
    else {
      priceMat[paste(rownum),
      paste(colnum)] <- priceMat[paste(rownum+1),
      paste(colnum-1)]*d
    }
  }
}

probMat <- matrix(0, ncol=nsteps+1, nrow=2*nsteps + 1,
dimnames=list(as.character(nsteps:-nsteps),
as.character(0:nsteps)))
probMat["0", "0"] <- 1
for(colnum in 1:nsteps ) {
  for(rownum in seq(colnum, -colnum, by=-2)) {
    if( !(rownum == -colnum) ) {
      probMat[paste(rownum),
      paste(colnum)] <- probMat[paste(rownum-1),
      paste(colnum-1)]*p + probMat[paste(rownum),
      paste(colnum)]
    }
    if( !(rownum == colnum) ) {
      probMat[paste(rownum),
      paste(colnum)] <- probMat[paste(rownum+1),
      paste(colnum-1)]*(1-p) + probMat[paste(rownum),
      paste(colnum)]
    }
```

```
> sigma <- 0.20
> nsteps <- 4
> source("~/binomial/binomialLattice.R")
> priceMat
     0         1         2         3         4
4   NA        NA        NA        NA 149.18247
3   NA        NA        NA 134.98588        NA
2   NA        NA 122.14028        NA 122.14028
1   NA 110.51709        NA 110.51709        NA
0  100        NA 100.00000        NA 100.00000
-1  NA  90.48374        NA  90.48374        NA
-2  NA        NA  81.87308        NA  81.87308
-3  NA        NA        NA  74.08182        NA
-4  NA        NA        NA        NA  67.03200
> plotLattice(priceMat, "Price Lattice", digits=2)
>
```

Execute From Source



```
# User Analysis

setwd("~/analysis")
source("prep.R")

library(plyr)
library(lattice)
library(ggplot2)

# Import data set
rawdata <- read.csv("stats.csv")
totalUsers <- dim(rawdata)[1]

# Clean data set
clean <- prepareStats(rawdata)

# Subset of active users
active <- subset(clean, active == 1)
count(active,"daysSinceAccountCreated < 30")[2,2]
mean(active$age)
```

Data

clean	360404 obs. of 35 variables
rawdata	530750 obs. of 35 variables

Values

totalUsers	530750L

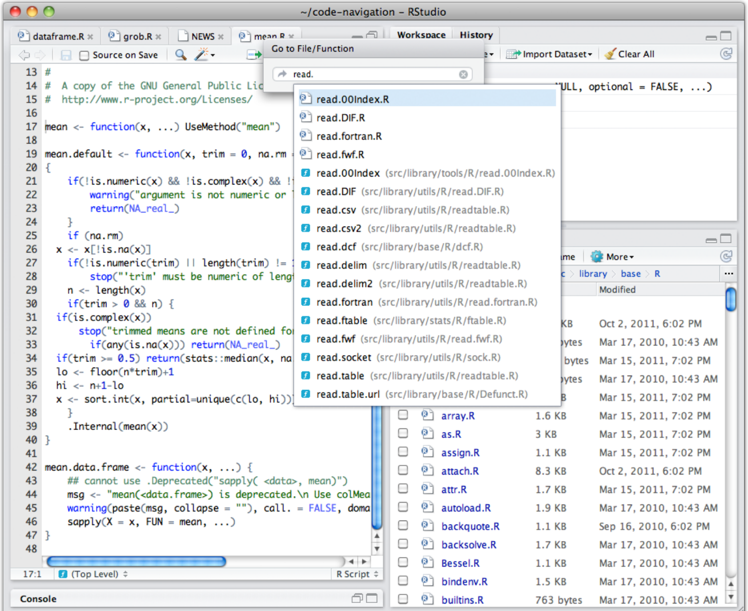
Functions

evalPercentage(expression, index)
formatInteger(object, ...)
prepareStats(data, sampleSize = NA)

☐	boot	Bootstrap Functions (originally by Angelo Canty for S)
☐	brew	Templating Framework for Report Generation
☐	class	Functions for Classification
☐	cluster	Cluster Analysis Extended Rousseeuw et al.
☐	codetools	Code Analysis Tools for R
☐	colorspace	Color Space Manipulation
☐	compiler	The R Compiler Package
☑	datasets	The R Datasets Package
☐	digest	Create cryptographic hash digests of R objects
☐	evaluate	Parsing and evaluation tools that provide more details than the default.
☐	foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...
☑	ggplot2	An implementation of the Grammar of Graphics
☑	graphics	The R Graphics Package

```
> setwd("~/analysis")
> source("prep.R")
> library(plyr)
> library(lattice)
> library(ggplot2)
Loading required package: grid
Loading required package: proto
> # Import data set
> rawdata <- read.csv("stats.csv")
> totalUsers <- dim(rawdata)[1]
>
> # Clean data set
> clean <- prepareStats(rawdata)
>
```

Go to File/Function



read.

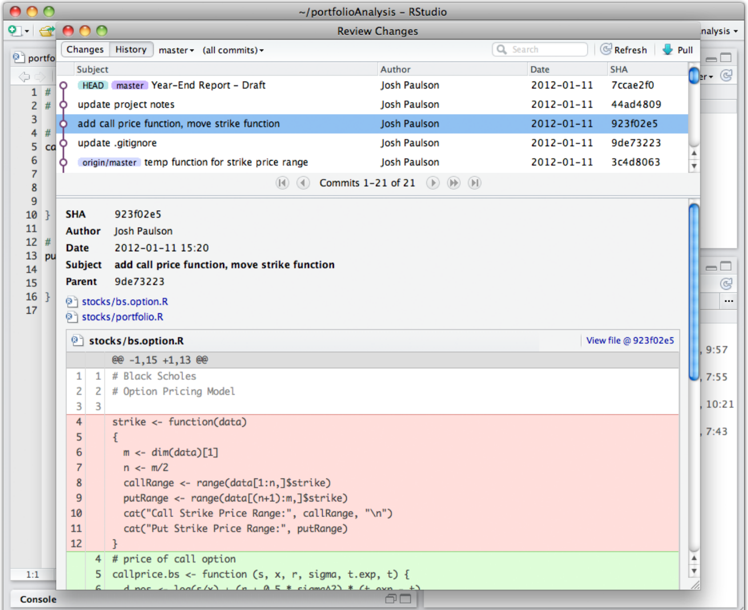
read.00Index.R
read.DIF.R
read.fortran.R
read.fwf.R
read.00Index (src/library/tools/R/read.00Index.R)
read.DIF (src/library/utils/R/read.DIF.R)
read.csv (src/library/utils/R/readtable.R)
read.csv2 (src/library/utils/R/readtable.R)
read.dcf (src/library/base/R/dcf.R)
read.delim (src/library/utils/R/readtable.R)
read.delim2 (src/library/utils/R/readtable.R)
read.fortran (src/library/utils/R/read.fortran.R)
read.ftable (src/library/stats/R/ftable.R)
read.fwf (src/library/utils/R/read.fwf.R)
read.socket (src/library/utils/R/sock.R)
read.table (src/library/utils/R/readtable.R)
read.table.url (src/library/base/R/Defunct.R)

```
#
#  A copy of the GNU General Public Lic
#  http://www.r-project.org/Licenses/

mean <- function(x, ...) UseMethod("mean")

mean.default <- function(x, trim = 0, na.rm
{
    if(!is.numeric(x) && !is.complex(x) && !
        warning("argument is not numeric or
        return(NA_real_)
    }
    if (na.rm)
  x <- x[!is.na(x)]
    if(!is.numeric(trim) || length(trim) != 1
        stop("'trim' must be numeric of leng
    n <- length(x)
    if(trim > 0 && n) {
  if(is.complex(x))
      stop("trimmed means are not defined fo
        if(any(is.na(x))) return(NA_real_)
  if(trim >= 0.5) return(stats::median(x, na
  lo <- floor(n*trim)+1
  hi <- n+1-lo
  x <- sort.int(x, partial=unique(c(lo, hi))
    }
    .Internal(mean(x))
}

mean.data.frame <- function(x, ...) {
    ## cannot use .Deprecated("sapply( <data>, mean)")
    msg <- "mean(<data.frame>) is deprecated.\n Use colMea
    warning(paste(msg, collapse = ""), call. = FALSE, doma
    sapply(X = x, FUN = mean, ...)
}
```

```
ULL, optional = FALSE, ...)
```

		Modified
	KB	Oct 2, 2011, 6:02 PM
	ytes	Mar 17, 2010, 10:43 AM
	bytes	Mar 15, 2011, 7:02 PM
	ytes	Mar 17, 2010, 10:43 AM
	B	Mar 15, 2011, 7:02 PM
array.R	1.6 KB	Mar 15, 2011, 7:02 PM
as.R	3 KB	Mar 15, 2011, 7:02 PM
assign.R	1.1 KB	Mar 15, 2011, 7:02 PM
attach.R	8.3 KB	Oct 2, 2011, 6:02 PM
attr.R	1.7 KB	Mar 15, 2011, 7:02 PM
autoload.R	1.9 KB	Mar 17, 2010, 10:43 AM
backquote.R	1.1 KB	Sep 16, 2010, 6:02 PM
backsolve.R	1.7 KB	Mar 17, 2010, 10:43 AM
Bessel.R	1.1 KB	Mar 17, 2010, 10:43 AM
bindenv.R	1.5 KB	Mar 17, 2010, 10:43 AM
builtins.R	763 bytes	Mar 17, 2010, 10:43 AM

Versionning



Subject	Author	Date	SHA
HEAD master Year-End Report – Draft	Josh Paulson	2012-01-11	7ccae2f0
update project notes	Josh Paulson	2012-01-11	44ad4809
add call price function, move strike function	Josh Paulson	2012-01-11	923f02e5
update .gitignore	Josh Paulson	2012-01-11	9de73223
origin/master temp function for strike price range	Josh Paulson	2012-01-11	3c4d8063

Commits 1–21 of 21

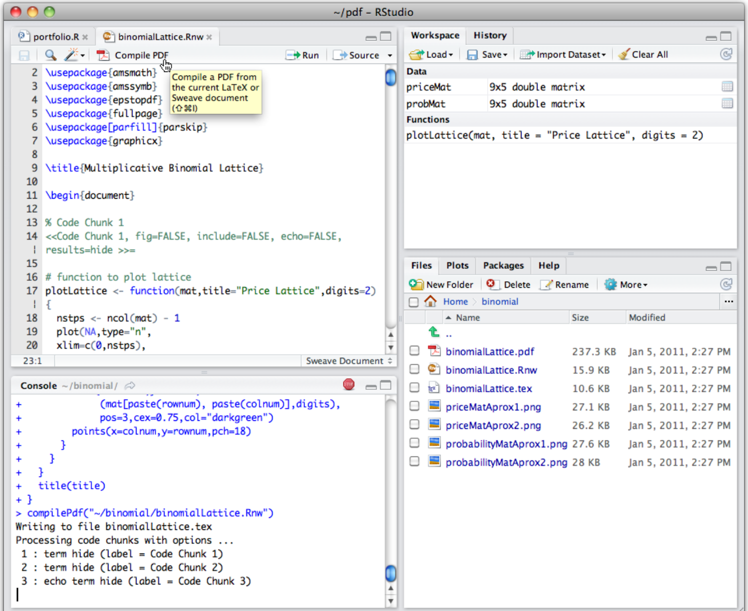
SHA 923f02e5
Author Josh Paulson
Date 2012-01-11 15:20
Subject **add call price function, move strike function**
Parent 9de73223

stocks/bs.option.R
stocks/portfolio.R

stocks/bs.option.R View file @ 923f02e5

```
@@ -1,15 +1,13 @@
# Black Scholes
# Option Pricing Model

strike <- function(data)
{
  m <- dim(data)[1]
  n <- m/2
  callRange <- range(data[1:n,]$strike)
  putRange <- range(data[(n+1):m,]$strike)
  cat("Call Strike Price Range:", callRange, "\n")
  cat("Put Strike Price Range:", putRange)
}
# price of call option
callprice.bs <- function (s, x, r, sigma, t.exp, t) {
```

Easy creation of Sweave documents



```
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{epstopdf}
\usepackage{fullpage}
\usepackage[parfill]{parskip}
\usepackage{graphicx}

\title{Multiplicative Binomial Lattice}

\begin{document}

% Code Chunk 1
<<Code Chunk 1, fig=FALSE, include=FALSE, echo=FALSE,
results=hide >>=

# function to plot lattice
plotLattice <- function(mat,title="Price Lattice",digits=2)
{
  nstps <- ncol(mat) - 1
  plot(NA,type="n",
  xlim=c(0,nstps),
```

Data

priceMat	9x5 double matrix
probMat	9x5 double matrix

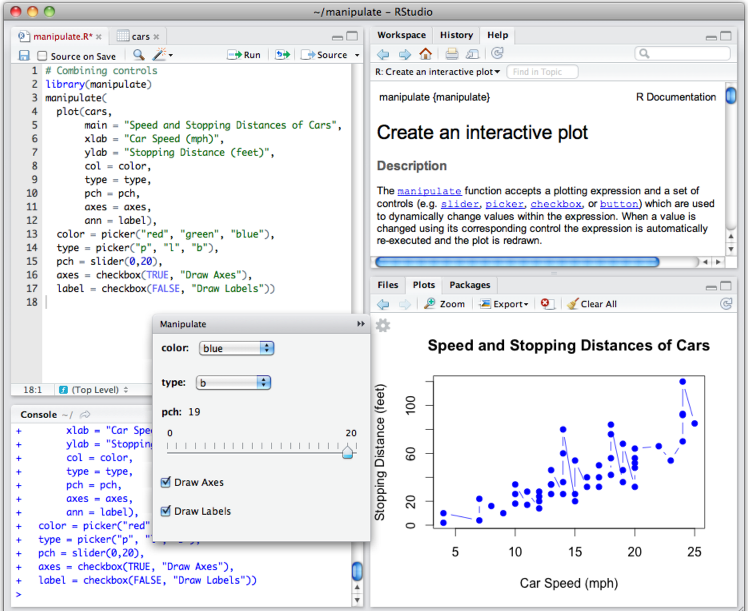
Functions

plotLattice(mat, title = "Price Lattice", digits = 2)

```
+              (mat[paste(rownum), paste(colnum)],digits),
+              pos=3,cex=0.75,col="darkgreen")
+         points(x=colnum,y=rownum,pch=18)
+       }
+     }
+   }
+   title(title)
+ }
> compilePdf("~/binomial/binomialLattice.Rnw")
Writing to file binomialLattice.tex
Processing code chunks with options ...
 1 : term hide (label = Code Chunk 1)
 2 : term hide (label = Code Chunk 2)
 3 : echo term hide (label = Code Chunk 3)
```

Name	Size	Modified
..		
binomialLattice.pdf	237.3 KB	Jan 5, 2011, 2:27 PM
binomialLattice.Rnw	15.9 KB	Jan 5, 2011, 2:27 PM
binomialLattice.tex	10.6 KB	Jan 5, 2011, 2:27 PM
priceMatAprox1.png	27.1 KB	Jan 5, 2011, 2:27 PM
priceMatAprox2.png	26.2 KB	Jan 5, 2011, 2:27 PM
probabilityMatAprox1.png	27.6 KB	Jan 5, 2011, 2:27 PM
probabilityMatAprox2.png	28 KB	Jan 5, 2011, 2:27 PM

Interactive graphics via the manipulate package



```r
# Combining controls
library(manipulate)
manipulate(
  plot(cars,
       main = "Speed and Stopping Distances of Cars",
       xlab = "Car Speed (mph)",
       ylab = "Stopping Distance (feet)",
       col = color,
       type = type,
       pch = pch,
       axes = axes,
       ann = label),
  color = picker("red", "green", "blue"),
  type = picker("p", "l", "b"),
  pch = slider(0,20),
  axes = checkbox(TRUE, "Draw Axes"),
  label = checkbox(FALSE, "Draw Labels"))
```

manipulate {manipulate} R Documentation

Create an interactive plot

Description

The manipulate function accepts a plotting expression and a set of controls (e.g. slider, picker, checkbox, or button) which are used to dynamically change values within the expression. When a value is changed using its corresponding control the expression is automatically re-executed and the plot is redrawn.

color: blue

type: b

pch: 19

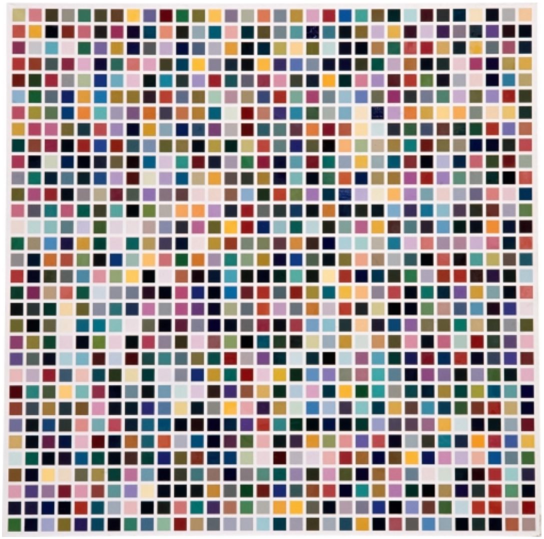
0 20

☑ Draw Axes

☑ Draw Labels

```
+        xlab = "Car Spe
+        ylab = "Stoppin
+        col = color,
+        type = type,
+        pch = pch,
+        axes = axes,
+        ann = label),
+   color = picker("red"
+   type = picker("p", "
+   pch = slider(0,20),
+   axes = checkbox(TRUE, "Draw Axes"),
+   label = checkbox(FALSE, "Draw Labels"))
>
```

An example



RStudio is good for you!

Thank you for your attention!