R for MBA

Presented by MBA students

JKSHIM, NITTE

Prof. Mohsin Ahmed

TECHBUGS 2014-09-16

Topics

Part 1, Introduction to R, by

- Jovita Monteiro and
- Nikitha Jackline Fernandes

Part 2, The R Language, by

- Kavya M Nagraj and
- Nishita Rai

Part 3 and 4 Stats and Finance in R, by

- Laxmi Nayak and
- Meet Amrutia

Introduction to R

Part 1. Introduction by

by Jovita Monteiro and Nikitha Fernandes

What is R?

- R is a powerful language and environment for Statistical computing and graphics.
- R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand in 1993.
- It is the successor to S, the statistics language S, developed by John Chambers at Bell Labs in 1976.

Advantages of R

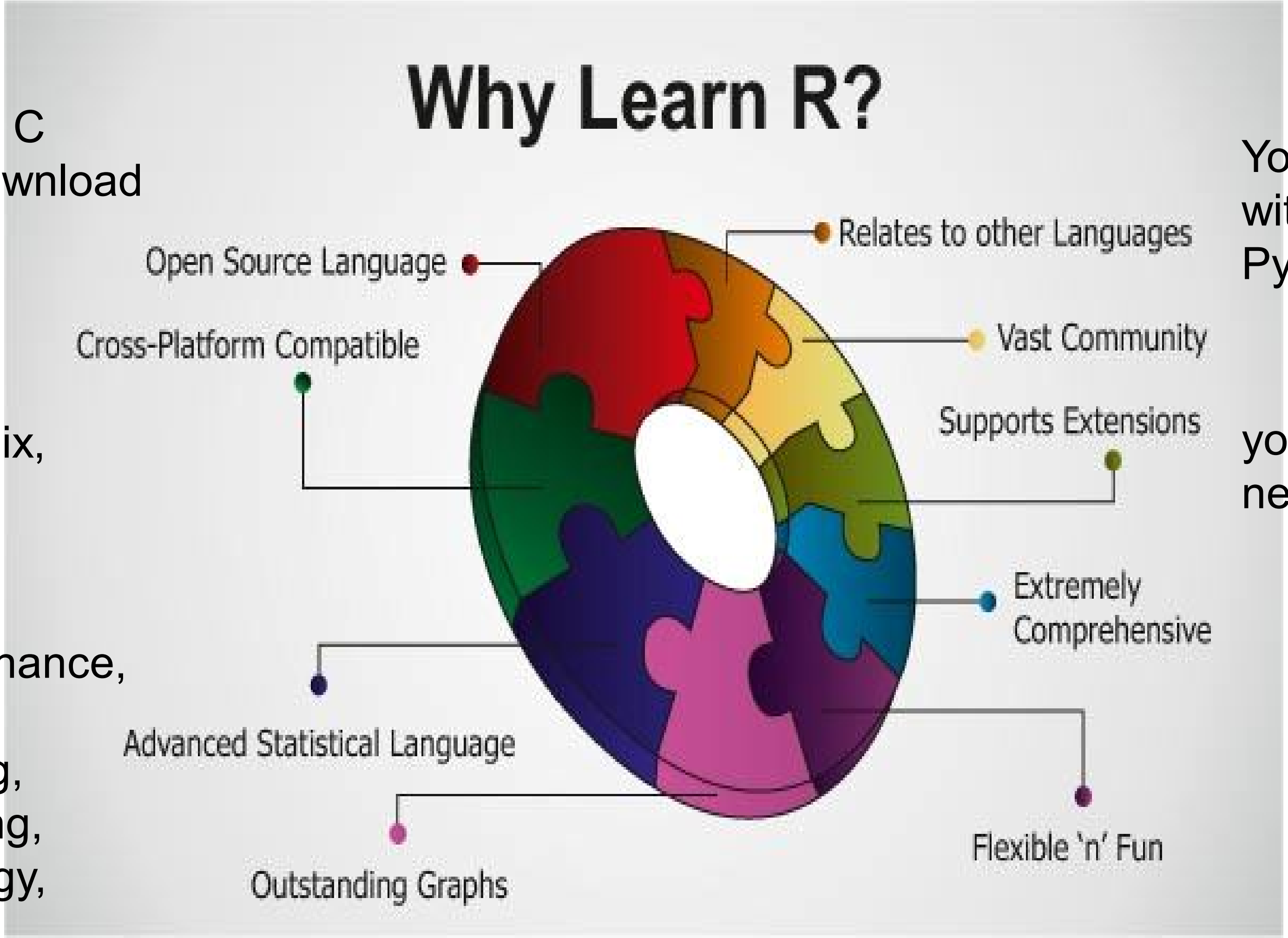
R is

- freeware
- runs on windows and linux
- lot of online help
- user friendly for basic users
- accurate for advanced users

Written in C
free to download

Runs on
Windows
Linux, Unix,
Macs

Use for finance,
statistics,
marketing,
advertising,
psychology,
research



Why Learn R?

- Open Source Language
- Cross-Platform Compatible
- Advanced Statistical Language
- Outstanding Graphs
- Relates to other Languages
- Vast Community
- Supports Extensions
- Extremely Comprehensive
- Flexible 'n' Fun

You can use
with C,
Python

you can add
new functions

Why MBA students should learn R?

- R is free open source language (free to download).
- R is cross platform compatible (can use on windows, linux, etc)
- Most advanced statistical package (better than spss, excel, stata).
- Outstanding graphical output.

Why MBA students should learn R?

Used in financial and fortune 500 companies for:

- Financial analysis on Wall St
- Pricing by sales teams
- Marketing and advertising
- HR for performance evaluation
- Researchers in Universities.

Why MBA students should learn R?

- R is extremely comprehensive.
- R support extensions - users can add new functions
- R has vast community.
- R can be used with other packages like Excel, SPSS, STATA.
- Higher Pay

The R Language

Part 2

by Kavya Nagraj and Nishita Rai

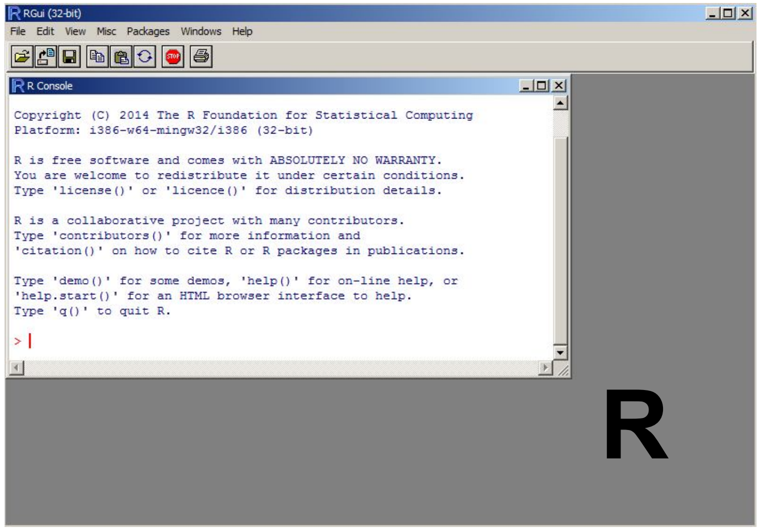
16/9/2014

History of R

- R is a Statistical Programming Language
- R is the programming language and environment that you write your commands and run in.
- R is the successor to the S language from Bell Labs in 1976.
- R created by Ross Ihaka and Robert Gentleman at University of Auckland, New Zealand in 1993.

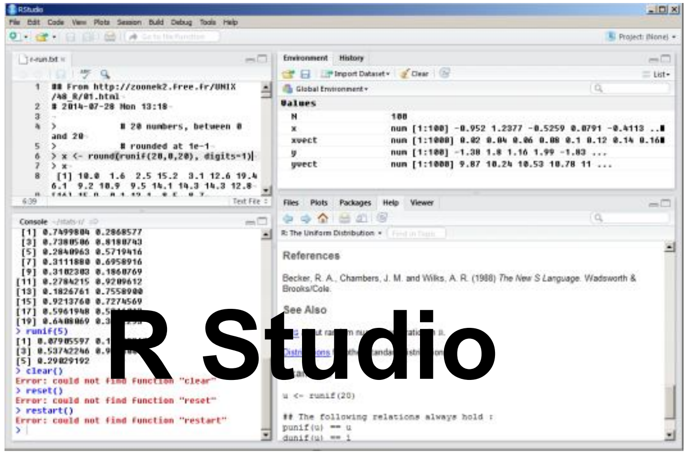
Installing R

1. Google “R stats windows download”
2. Download R and R-Studio
3. Install R and R-Studio
4. Start R by clicking on it



```
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

R


RStudio
File Edit Code View Plots Session Build Debug Tools Help
Project: (None)
r-run.txt
Environment History
Import Dataset
Clear
List
Global Environment
Files Plots Packages Help Viewer
R: The Uniform Distribution
Text File
Console

```
## From http://zoonek2.free.fr/UNIX
/48_R/01.html
# 2014-07-28 Mon 13:18

>            # 20 numbers, between 0
and 20
>            # rounded at 1e-1
> x <- round(runif(20,0,20), digits=1)
> x
 [1] 10.0  1.6  2.5 15.2  3.1 12.6 19.4
6.1  9.2 10.9  9.5 14.1 14.3 14.3 12.0
```

```
Values
  N          100
  x          num [1:100] -0.952 1.2377 -0.5259 0.0791 -0.4113 ...
  xvect      num [1:1000] 0.02 0.04 0.06 0.08 0.1 0.12 0.14 0.16
  y          num [1:100] -1.38 1.8 1.16 1.99 -1.03 ...
  yvect      num [1:1000] 9.87 10.24 10.53 10.78 11 ...
```

```
 [1] 0.7499804 0.2868577
 [3] 0.7380506 0.8180743
 [5] 0.2840963 0.5719416
 [7] 0.3111880 0.6958916
 [9] 0.3102303 0.1860769
[11] 0.2784215 0.9209612
[13] 0.1826761 0.7558900
[15] 0.9213760 0.7274569
[17] 0.5961948 0.5
[19] 0.6408069 0.3
> runif(5)
[1] 0.07905597 0.1
[3] 0.53742246 0.9
[5] 0.29029192
> clear()
Error: could not find function "clear"
> reset()
Error: could not find function "reset"
> restart()
Error: could not find function "restart"
> 
```

References

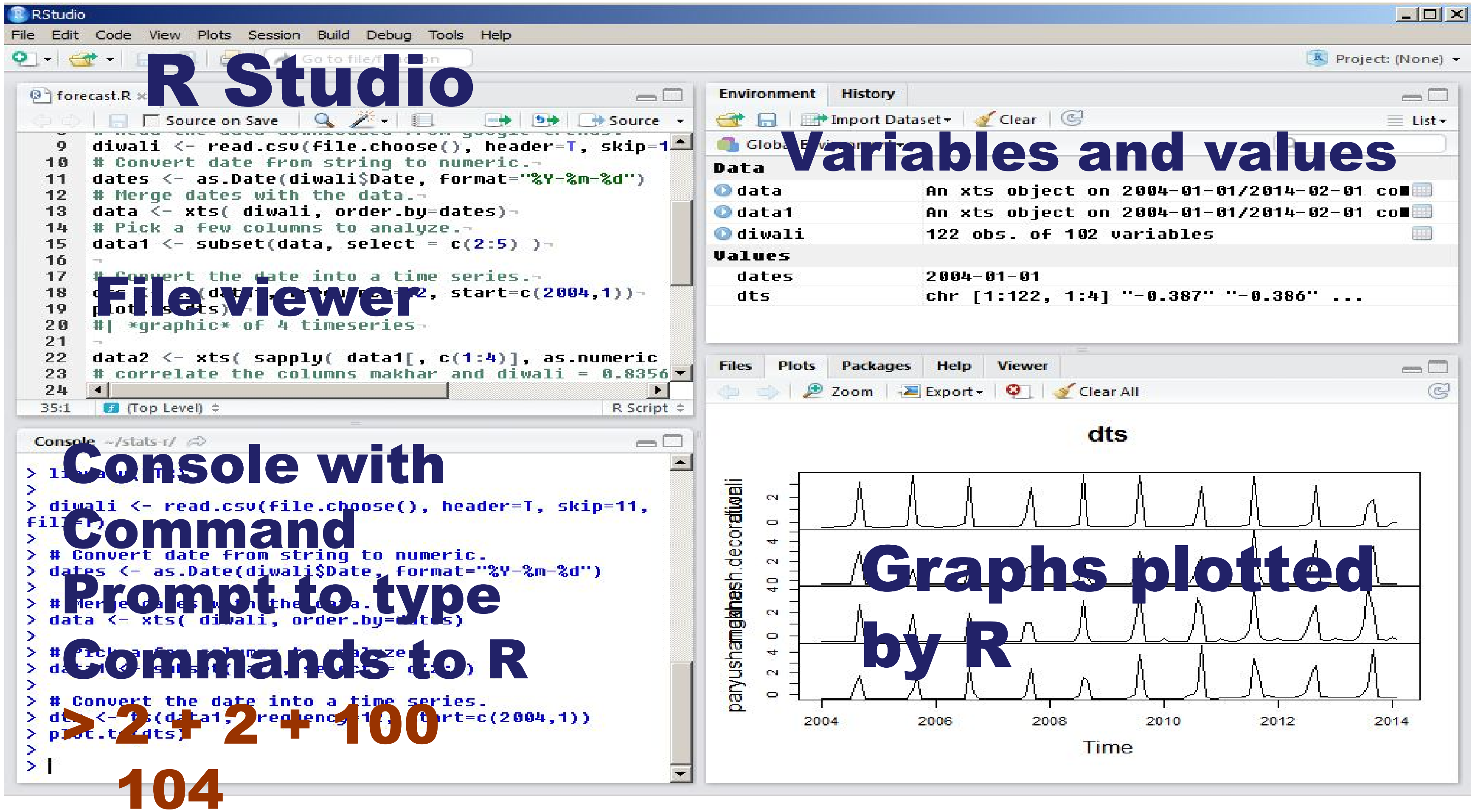
Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

```
u <- runif(20)

## The following relations always hold :
punif(u) == u
dunif(u) == 1
```

R Studio

R Studio



R Studio

File viewer

```
diwali <- read.csv(file.choose(), header=T, skip=1
# Convert date from string to numeric.
dates <- as.Date(diwali$Date, format="%Y-%m-%d")
# Merge dates with the data.
data <- xts( diwali, order.by=dates)
# Pick a few columns to analyze.
data1 <- subset(data, select = c(2:5) )

# Convert the date into a time series.
dts <- ts(data1, frequency=12, start=c(2004,1))
plot.ts(dts)
#| *graphic* of 4 timeseries

data2 <- xts( sapply( data1[, c(1:4)], as.numeric
# correlate the columns makhar and diwali = 0.8356
```

Variables and values

```
Data
data        An xts object on 2004-01-01/2014-02-01 co
data1       An xts object on 2004-01-01/2014-02-01 co
diwali      122 obs. of 102 variables
Values
dates       2004-01-01
dts         chr [1:122, 1:4] "-0.387" "-0.386" ...
```

Console with Command Prompt to type Commands to R

```
Console ~/stats-r/
> library(TTR)
>
> diwali <- read.csv(file.choose(), header=T, skip=11,
fill=T)
>
> # Convert date from string to numeric.
> dates <- as.Date(diwali$Date, format="%Y-%m-%d")
>
> # Merge dates with the data.
> data <- xts( diwali, order.by=dates)
>
> # Pick a few columns to analyze.
> data1 <- subset(data, select = c(2:5) )
>
> # Convert the date into a time series.
> dts <- ts(data1, frequency=12, start=c(2004,1))
> plot.ts(dts)
>
> |
```

```
> 2 + 2 + 100
104
```

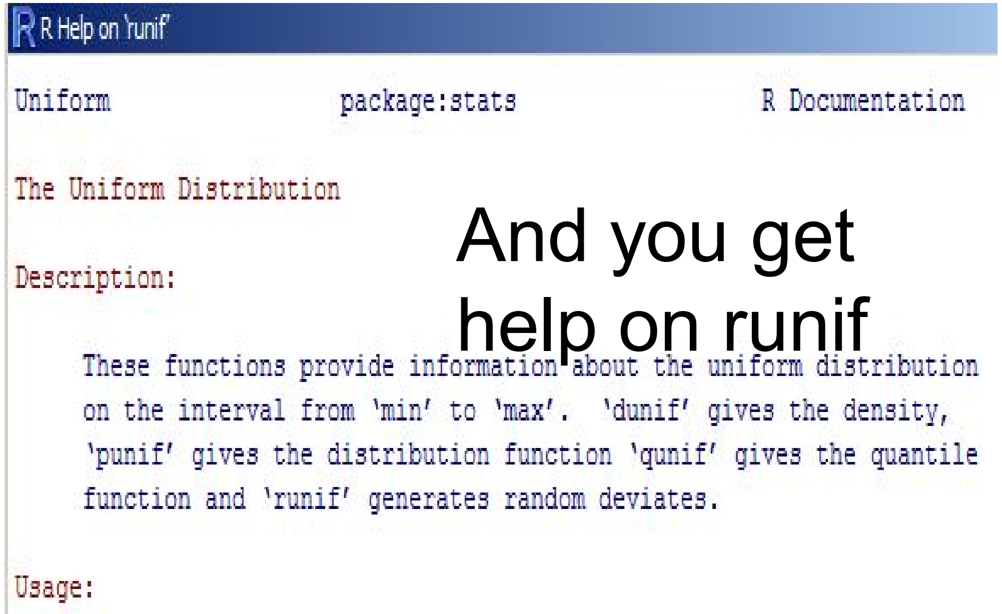
Graphs plotted by R

Workflow in R

1. Read Data into R
2. Analyze Data
3. Visualize Data
4. Make Conclusions from Data

Help in R Studio

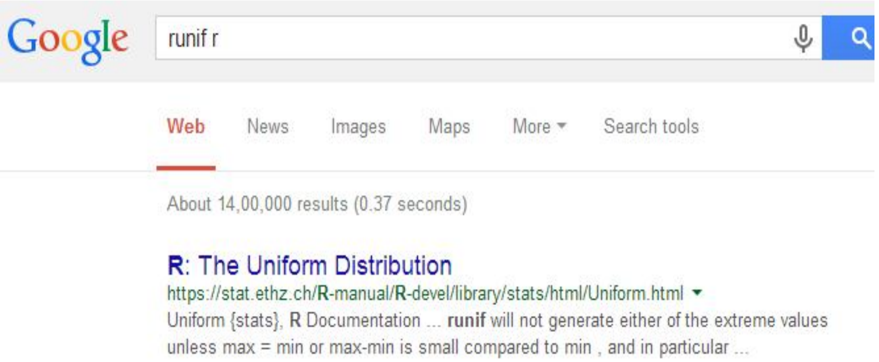
```
> ?runif
```



```
Uniform                  package:stats                  R Documentation

The Uniform Distribution

Description:

     These functions provide information about the uniform distribution
     on the interval from 'min' to 'max'.  'dunif' gives the density,
     'punif' gives the distribution function 'qunif' gives the quantile
     function and 'runif' generates random deviates.

Usage:
```

And you get help on runif

Use Google search for “Runif R statistics”



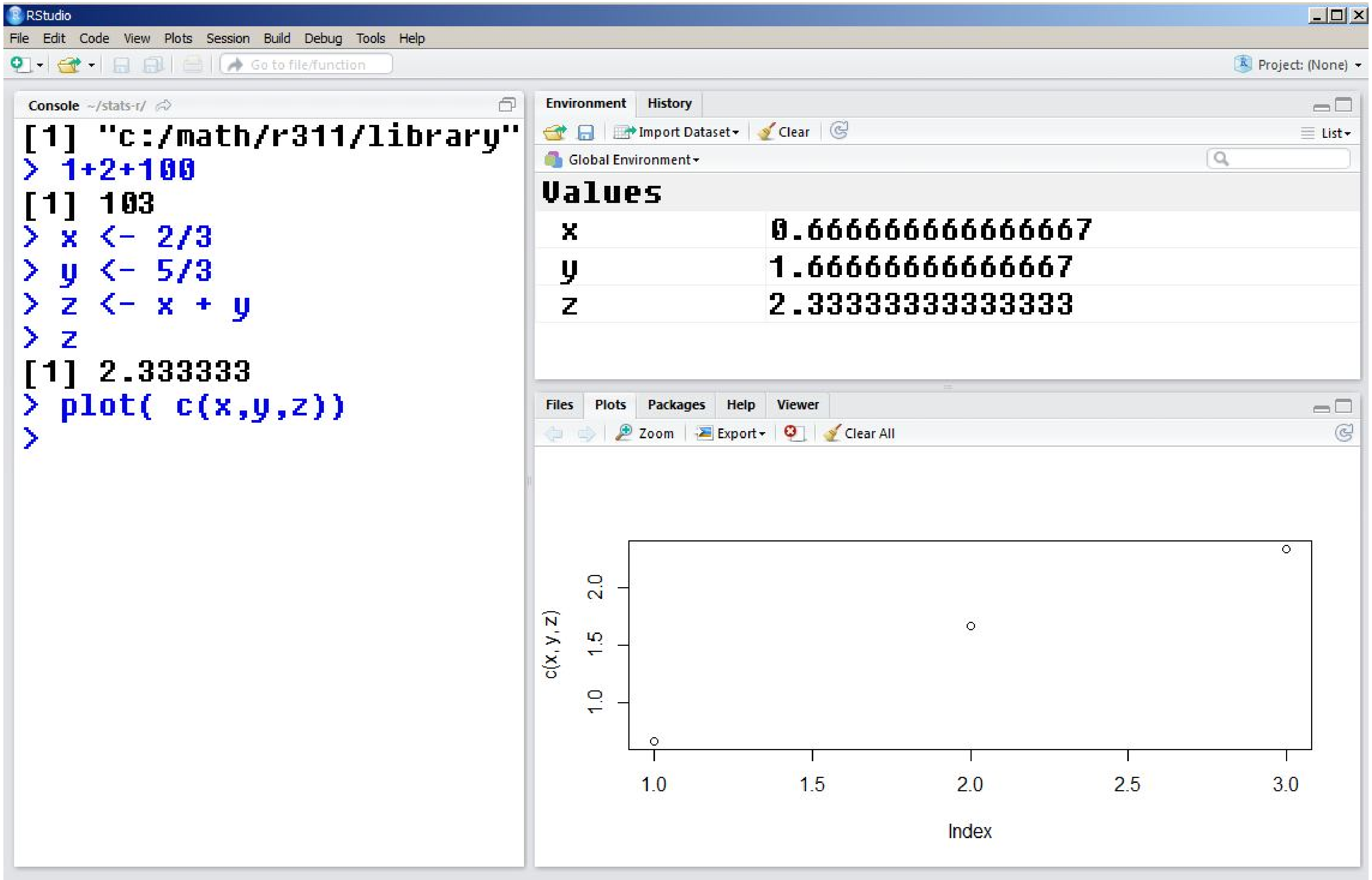
About 14,00,000 results (0.37 seconds)

R: The Uniform Distribution
https://stat.ethz.ch/R-manual/R-devel/library/stats/html/Uniform.html
Uniform {stats}, R Documentation ... **runif** will not generate either of the extreme values unless max = min or max-min is small compared to min , and in particular ...

R Command Prompt

```
# Start R by clicking on its icon

>   # Comment lines are  ignored by R


> 2+2     # You type commands at the R prompt.
  4       # Result ‘4’ printed by R


> 1+1/2+1/3+2/3
    2.5
```

R as a calculator

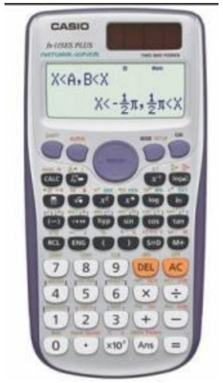


```
[1] "c:/math/r311/library"
> 1+2+100
[1] 103
> x <- 2/3
> y <- 5/3
> z <- x + y
> z
[1] 2.333333
> plot( c(x,y,z))
>
```

Values

```
x                0.666666666666667
y                1.66666666666667
z                2.33333333333333
```

R as a Scientific calculator



```
> 1+1/2+1/3+2/3
  2.5


> sqrt( 2i )         # Complex numbers
  1+1i


> 1/0               # Divide by zero
   Inf              # Infinite


> 0/0
  NaN              # NaN = Not-A-Number, undefined
```

Data: Vector of numbers

```
# Sequence of numbers from 1 to  5
> 1:5
   1  2  3  4  5

# Create 4 numbers and
# save it in a vector named u.
> u   <-   c(1, 4, 0, -2)
```

Sequences and vector

```
> 1:5                    #  1 2 3 4 5
> c(1,2,3,4,5)           #  1 2 3 4 5
> seq( 0, 4, len=3)      #  0 2 4
> seq( 0, 4, by=2)       #  0 2 4

> c(a=1, b=5, c=10)  # Named vector
  a   b   c
  1   5  10
```

summary(data)

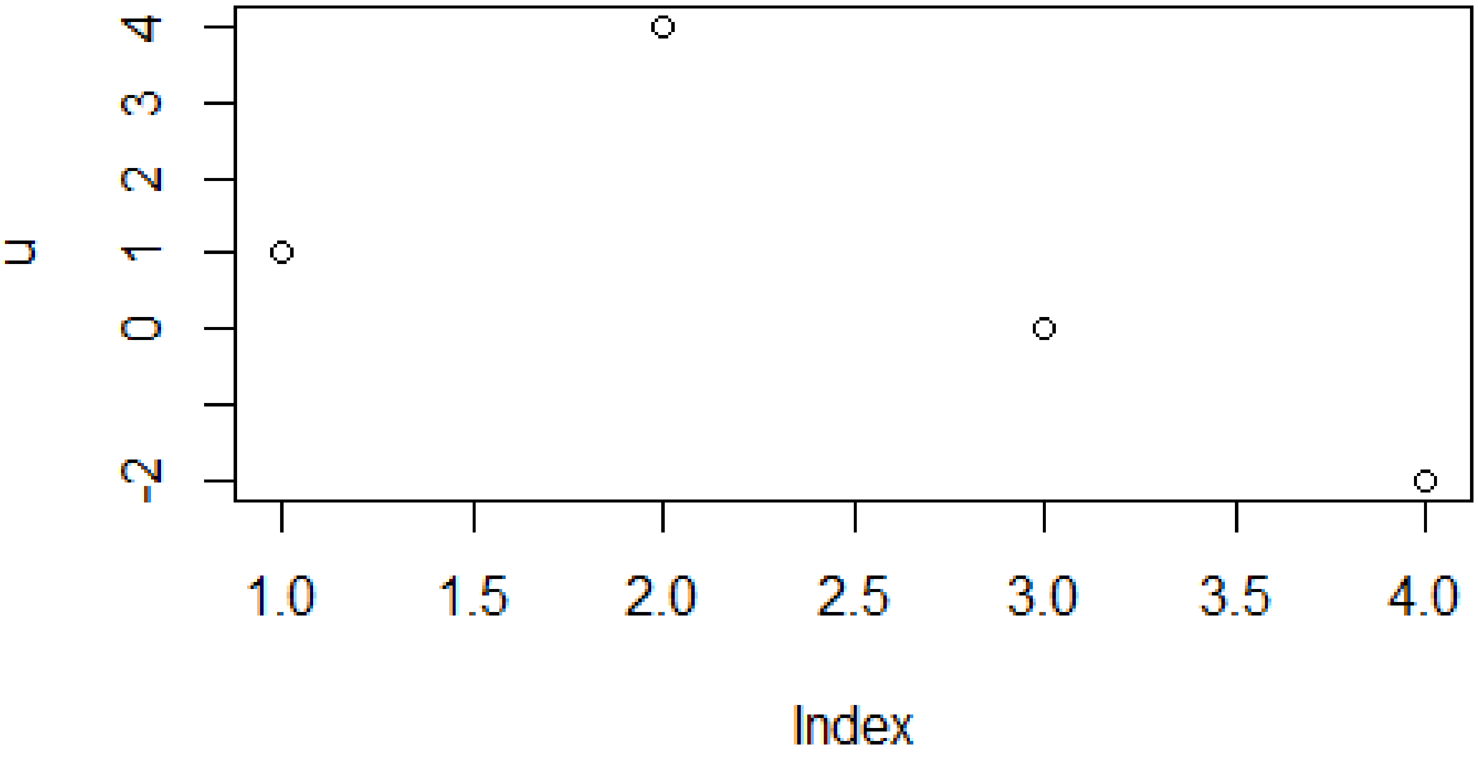
```
> u   <-   c(1, 4, 0, -2)

> summary(u)
Min.    1Q.    Median  Mean    3Q.    Max.
-2      -0.5     0.5        0.75     1.75    4
```

quantile(data)

```
> u   <-   c(1, 4, 0, -2)
> quantile(u)
  0%      25%    50%   75% 100%
 -2      -0.5      0.5    1.75  4

> quantile(u, c(0, 0.33, 0.66, 1) )
   0%     33%  66% 100%
  -2     -0.02   0.98   4
```

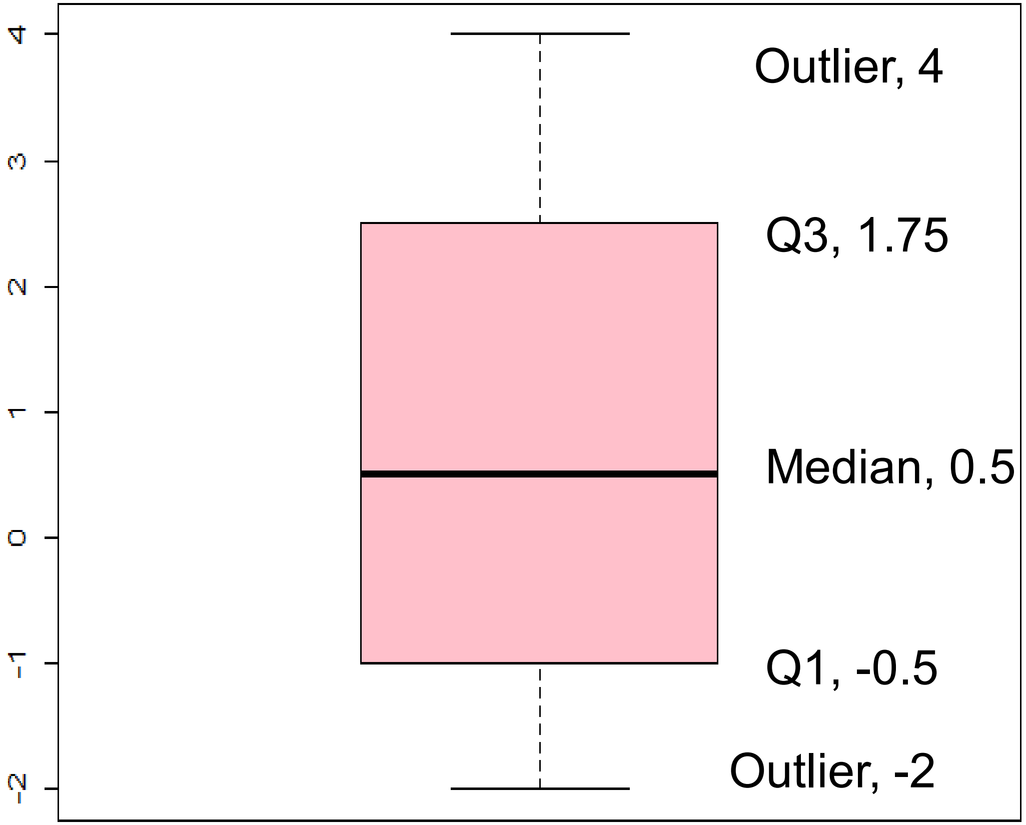
plot(data)

```
> u   <-   c(1, 4, 0, -2)
> plot(u)
```



boxplot(data)

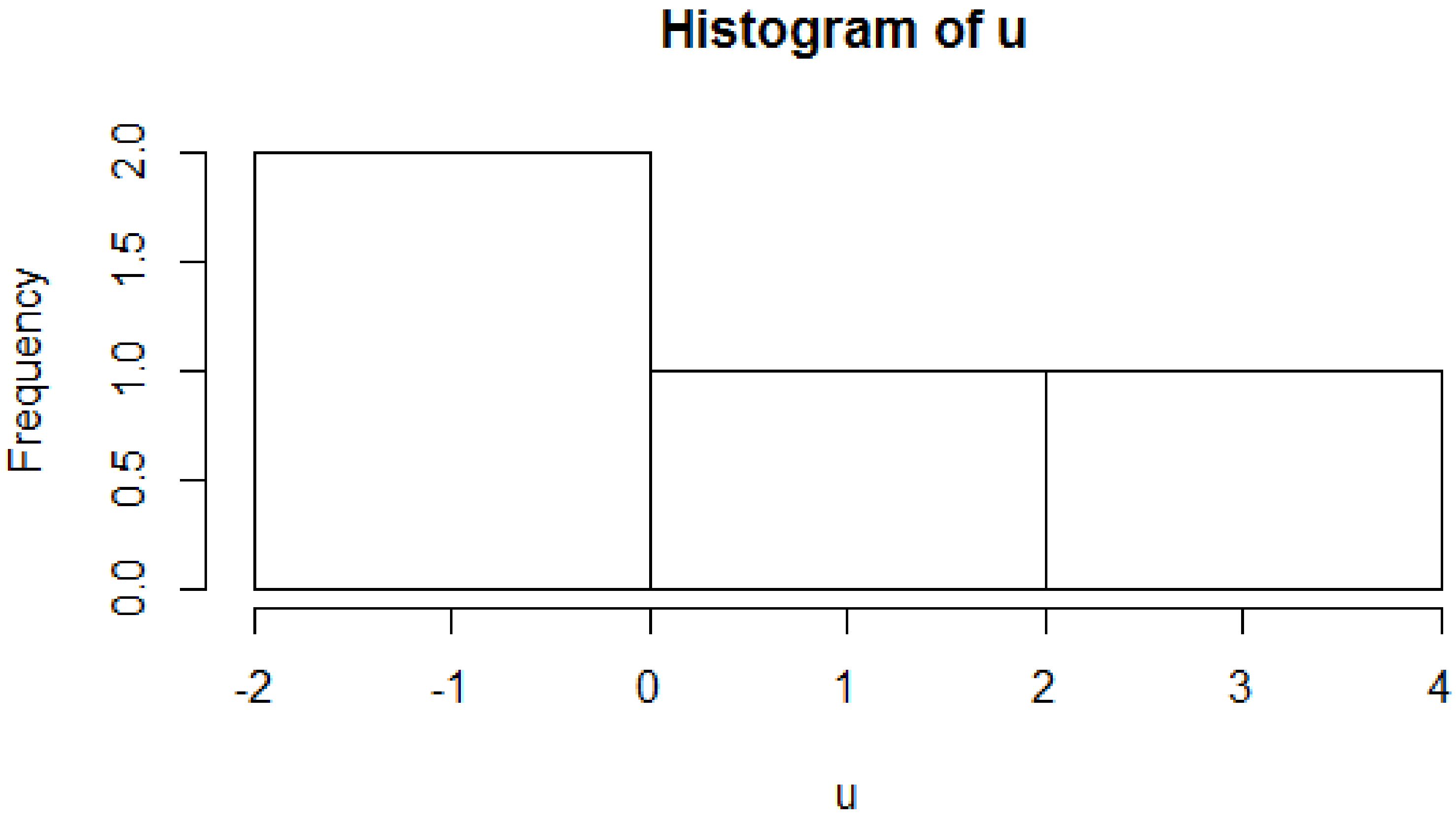
```
> u   <-   c(1, 4, 0, -2)
> boxplot(u)
```



```
> median(u)
0.5
> summary(u)
   Min.   1Q.    Median Mean  3Q.    Max.
   -2     -0.5    0.5      0.75   1.75   4
```

histogram(data)

```
> u   <-   c(1, 4, 0, -2)
> hist(u)
```



Statistical functions

```
> u   <-   c(1, 4, 0, -2)
> mean(u)
   0.75
> sd(u); max(u); min(u); median(u); var(u)
    2.5,      4,         -2,          0.5,             6.25
> sum(u) ; length(u)
   3,            4,
```

Not Available: NA and NaN

```
> str( log ( c(-1,   0,    1,   2,     NA ) ) )
               NaN  -Inf   0  0.693   NA
Warning ..  NaNs produced

> is.finite(  log(c(-1, 0, 1, 2, NA))  )
                    F F  T  T  F
```

Dealing with Missing Values

```
> x <- c(1,5,9,NA,2)
> mean(x)
 NA     # NA = Not Available

# Find mean after removing NA
> mean( x, na.rm=T )
 4.25

# Find NA in x
> is.na(x)
F F F T F
```

Make some random numbers

```
# Make 3 random uniform numbers
> runif(3)
   0.428   0.142    0.877

# Make 3 numbers between 5 to 10
> runif(3, 5,10)
    6.749    8.611    8.108
```

Random numbers

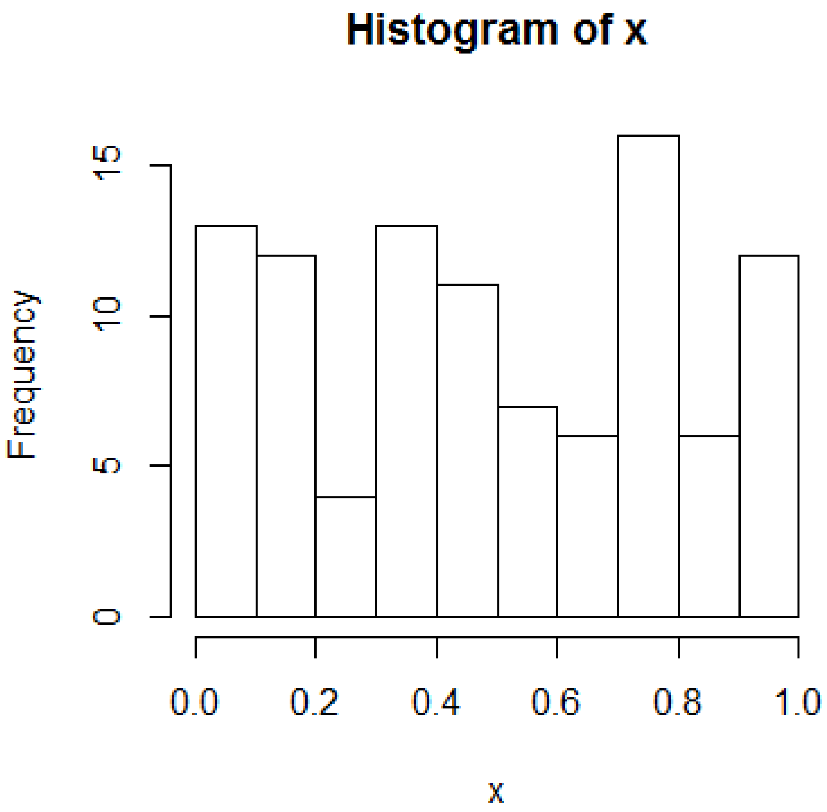
```
# Generate 3 random numbers in
# the range 5 to10,
# round them to 1 decimal digit.

> round( runif(3, 5, 10), digits=1)
[1]   5.5   9.7    9.5
```

Save the numbers in variable y

```
# Save 3 numbers in a variable named y
>    y <- runif(3)
# See what’s in y
>     y
[1]  0.179   0.384    0.176
```

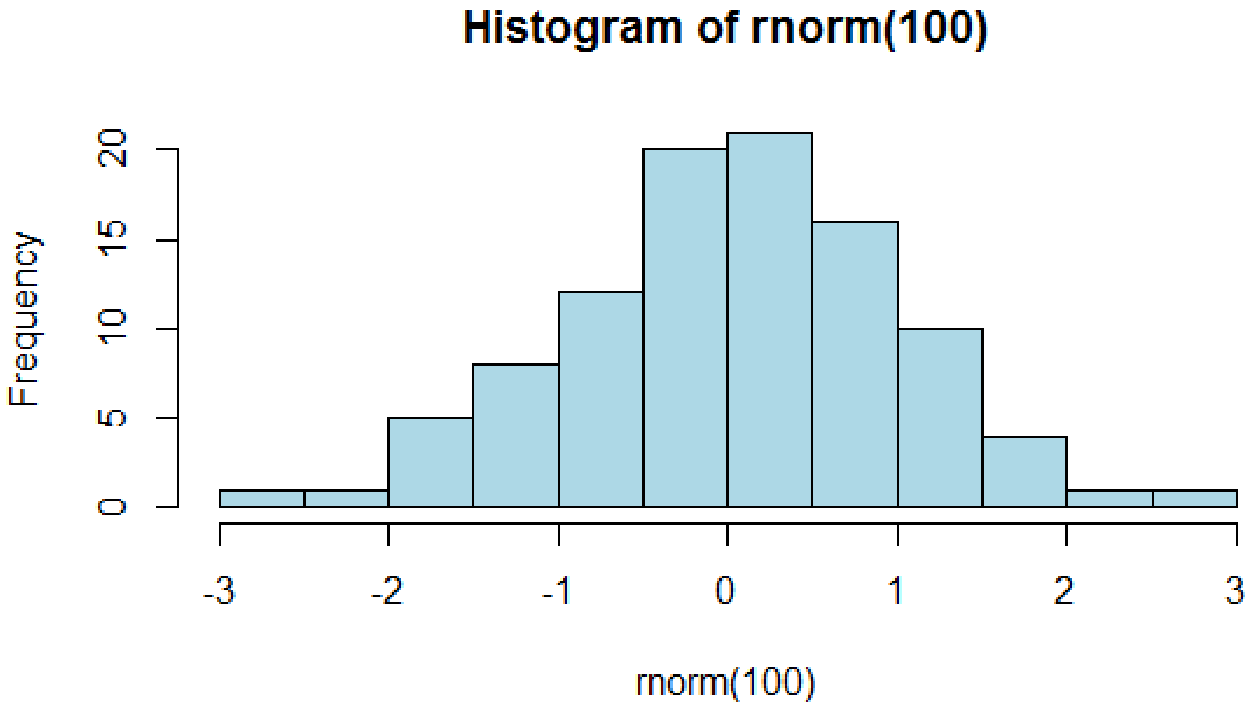
Histogram

```
# Save100 numbers in a
# variable named x
>    x <- runif(100)

# Plot the histogram
>   hist( x )
```



Histogram of normally distributed random numbers

```
#  100 normal distributed random numbers
> hist(  rnorm(100),   col="light blue“ )
```



Data sharing with Excel, SPSS

```
# Reading excel csv data files
sales <- read.csv( file.choose() )

# Import the spss data file
read.spss("newData.sav")
```

Data sharing with Excel, SPSS

```
# Reading excel csv data files
sales <- read.csv( file.choose() )
prices <- read.csv(“prices-2012.csv”)

# Load the foreign package
library(foreign)
# Import the spss data file
read.spss("newData.sav")
```

Data frame (excel sheet)

```
# 3 columns: a, b c
> x <- data.frame(a=1:3, b=5:7,
  c=11:13 )
> x
    a b c
1   1 5 11
2   2 6 12
3   3 7 13
> x$a      # Get column ‘a’ of x, same as x[[‘a’]]
  1 2 3
```

Columns of a data frame

```
> x$c <- NULL     # delete column c.
> x$d <- 21:23    # add new column d.
> x
     a  b  d
1    1  5  21
2    2  6  22
3    3  7  23
```

Combine two sheets with cbind

```
> y <- 31:33
> cbind( x, y)
  a b d  y
1 1 5 21 31
2 2 6 22 32
3 3 7 23 33
```

Omit rows with missing data

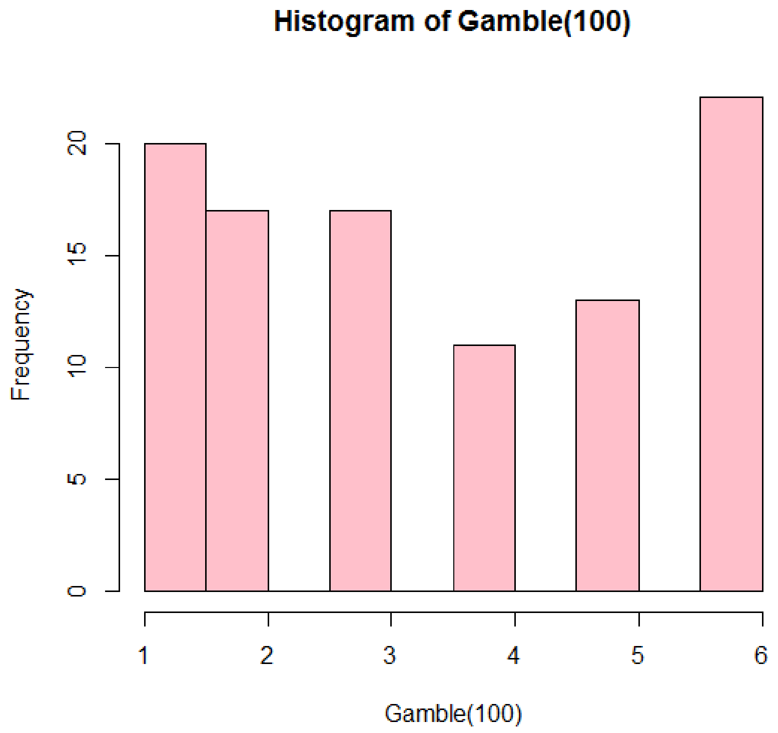
```
> x <- c(1,2,NA,4)
> d <- data.frame(x, y=rev(x))
> d
     x    y
1    1    4
2    2    NA
3    NA   2
4    4    1
> na.omit(d)        # Remove rows with NA
1      1   4
2      4   1
```

Matrix

```
> m <- matrix( c(1,2,3,4), nrow=2)
> m
      [,1]  [,2]
[1,]    1     3          # Row 1
[2,]    2     4          # Row 2

# Determinant of m = 1x4-2x3 = 4 - 6 = -2
> det(m)
  -2
```

Define your own function

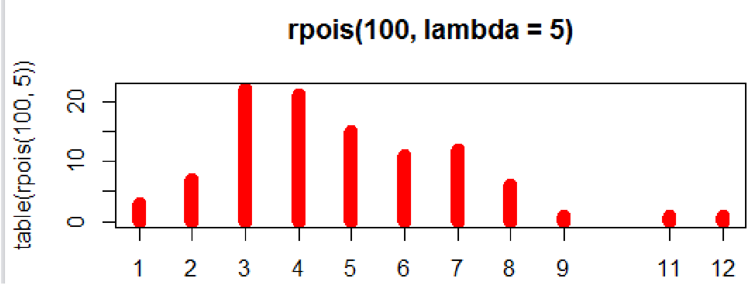
```
# Create a function.
> Gamble = function(n)
sample(1:6, n, replace=T)

# Call Gamble with n=4
> Gamble(4)
[1] 3 4 3 6

# Plot Gamble
> hist(  Gamble(100),
         col="pink")
```



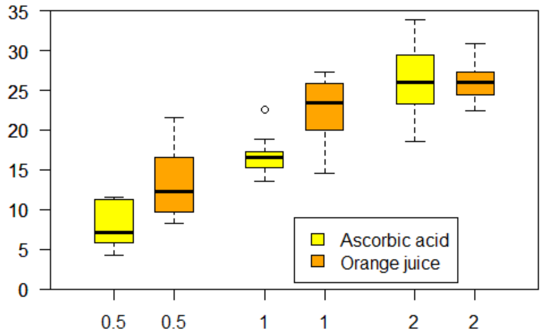
Builtin Data and Examples in R

Try these examples in R

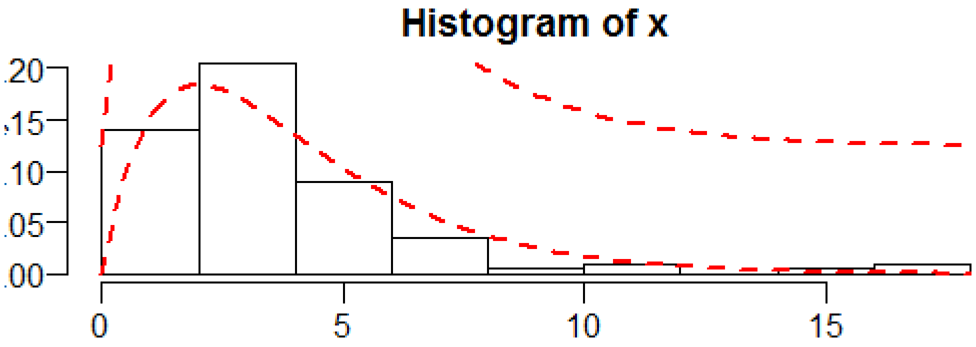
> example(plot)



> example(boxplot)



> example(hist)



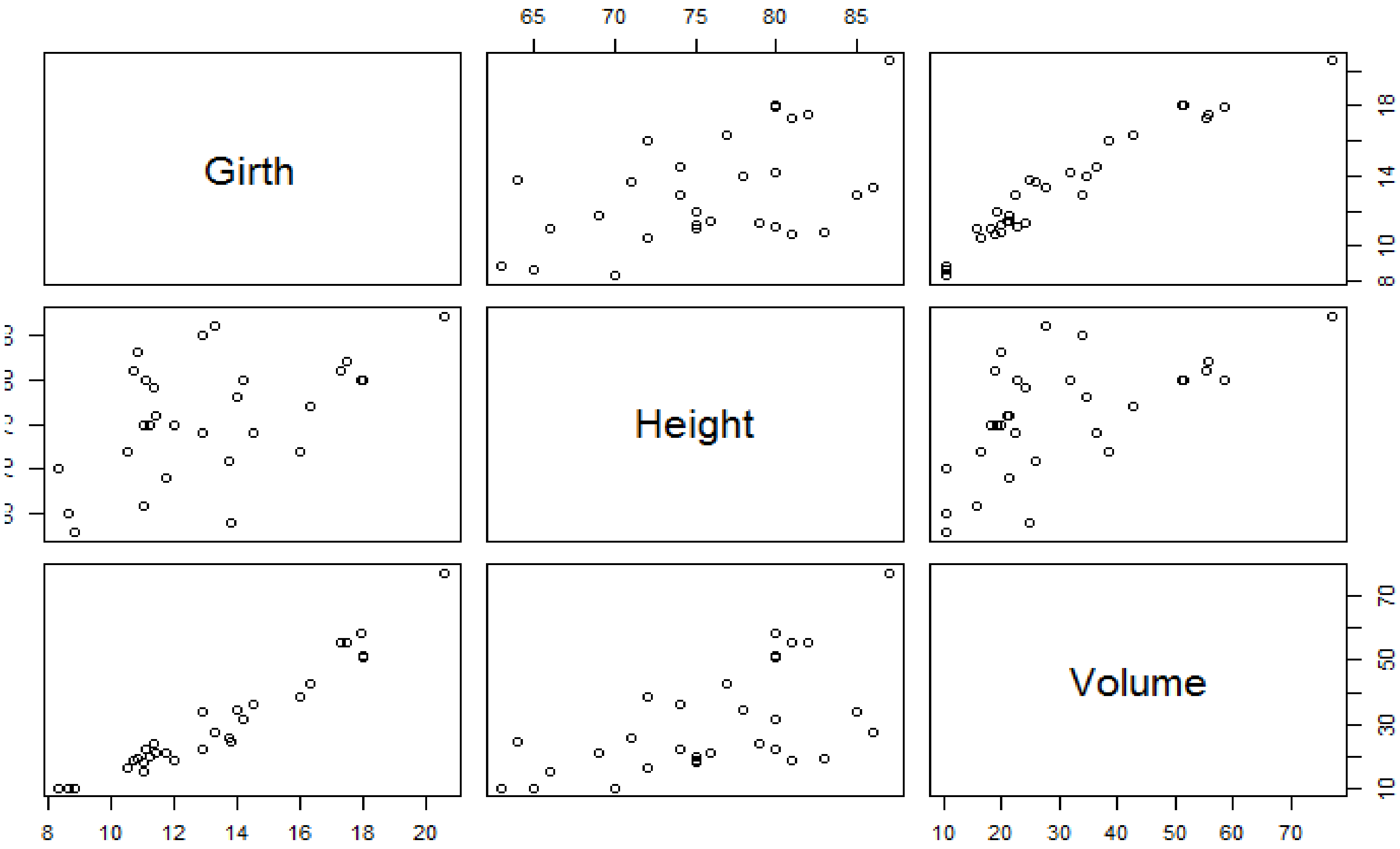
Playing with Builtin Data

```
> data()        # see the list of builtin datasets
> data( trees )
> ? trees       # see info about trees data
   This data set provides measurements of the
   girth, height and volume of timber ...
```

Correlation in tree data

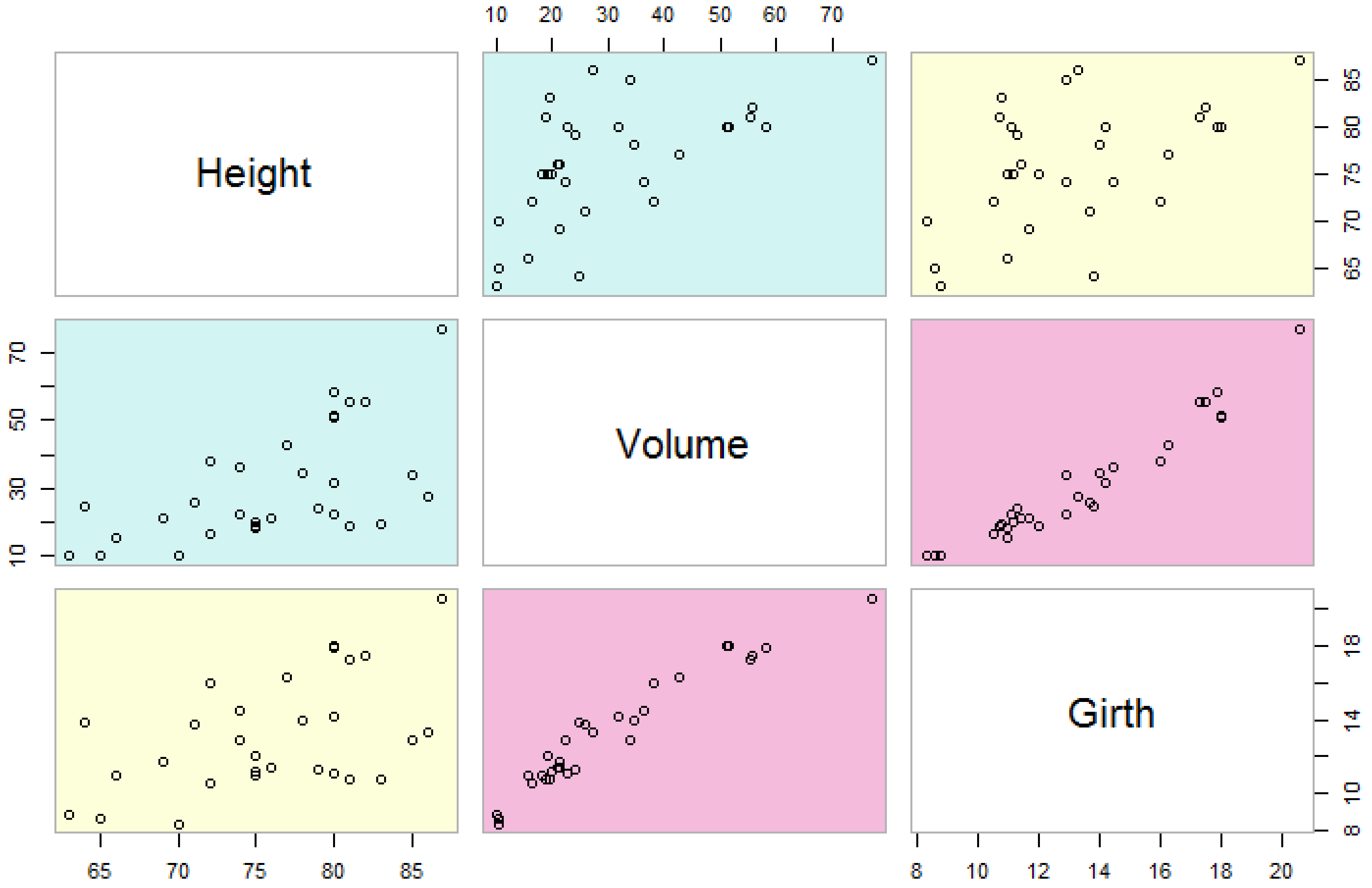
# plots correlation of girth, height volume of trees.

```
> pairs( trees )
```



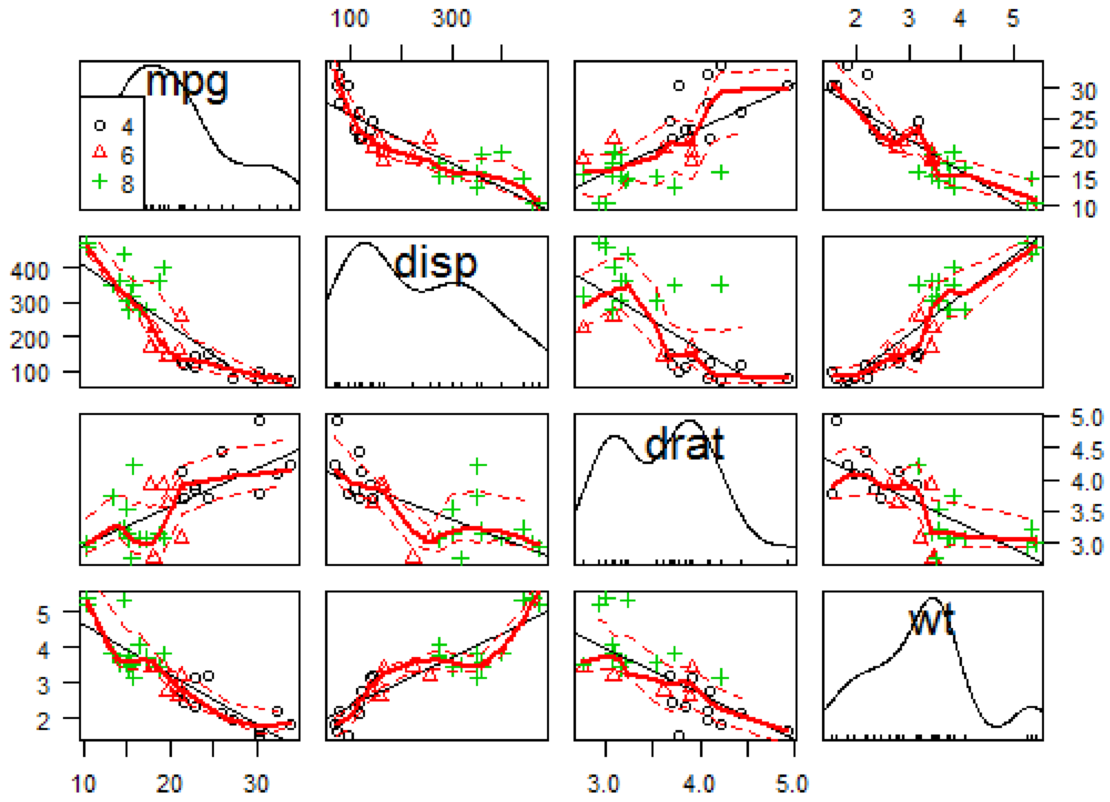
# Tree Variables Ordered and Colored by Correlation

```
> library(gclus)
> cpairs(trees,  order.single(cor(trees)),
                 dmat.color(cor(trees)) )
```



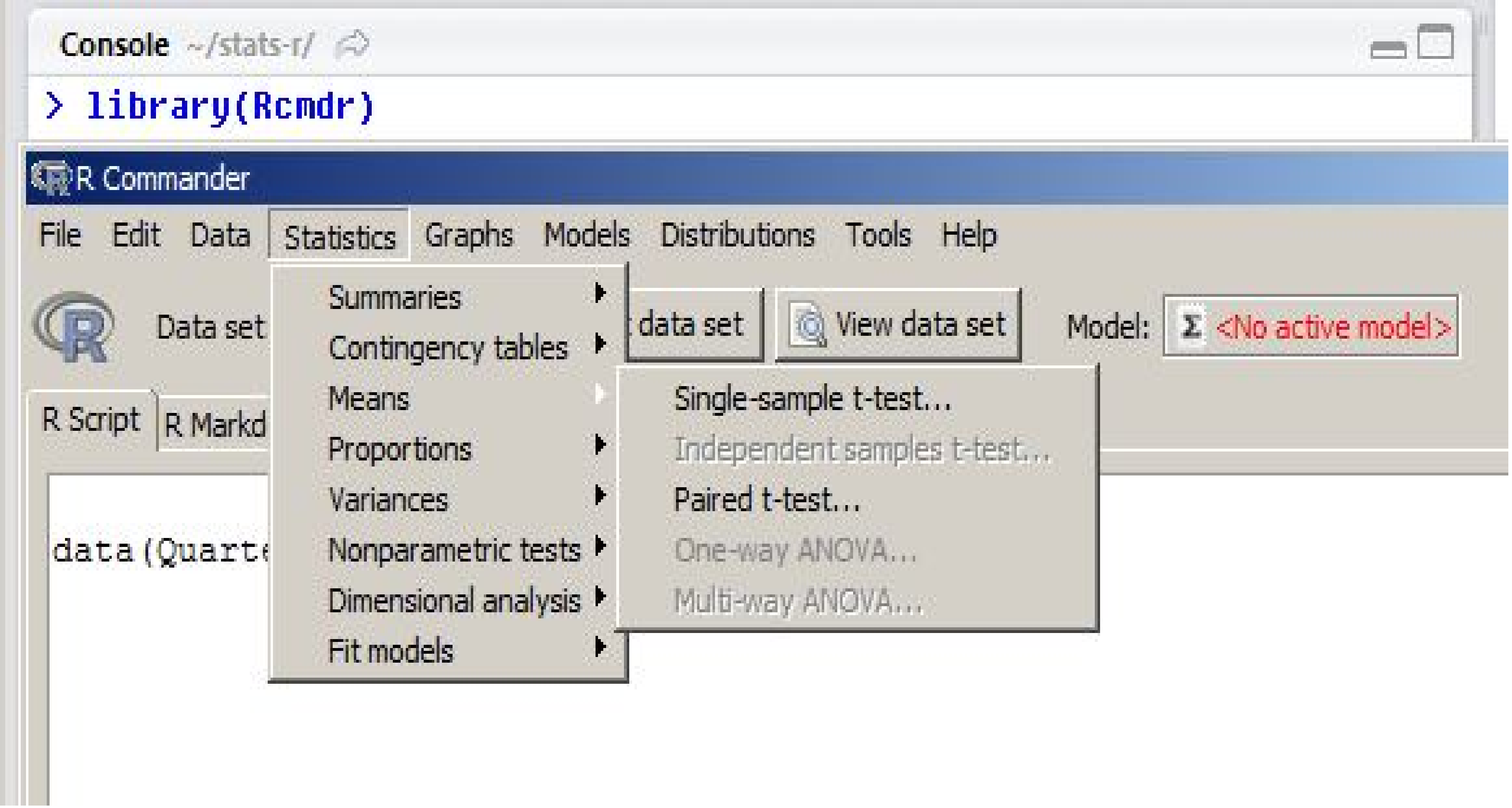
Scatterplot Matrices from the car Package

```
> library(car)
> scatterplotMatrix( ~mpg +disp + drat +wt | cyl,
    data=mtcars)
```



Use R Commander

> library(Rcmdr)

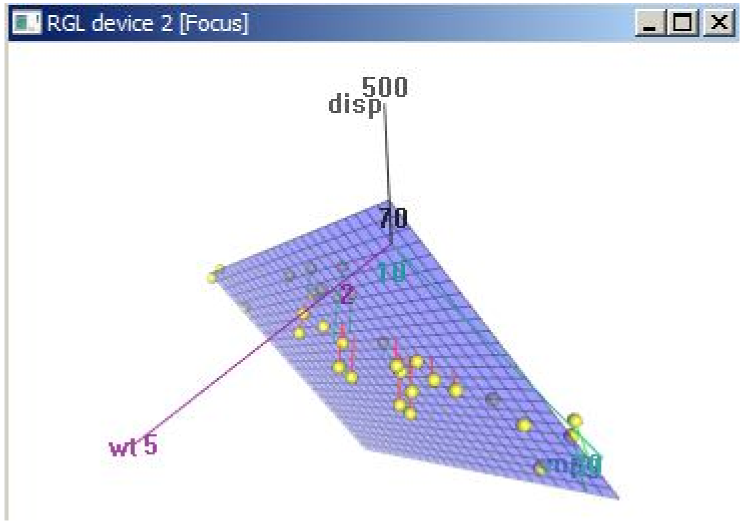


```
> library(Rcmdr)
```

```
data(Quarte
```

3D graphs

```
> library(Rcmdr)
> attach(mtcars)
> ?mtcars     # help on car data
> scatter3d(wt, disp, mpg)
```



Statistical Tests and Regression in R

Part 3. by Laxmi Nayak

To roll a Dice (Die) 10 times.

```
> sample(1:6, 1)  # one throw
   2
```

```
# Throw dice 10 times
> sample( 1:6, 10, replace=T)
      5 6 3 2 5 5 3 4 1 6
```

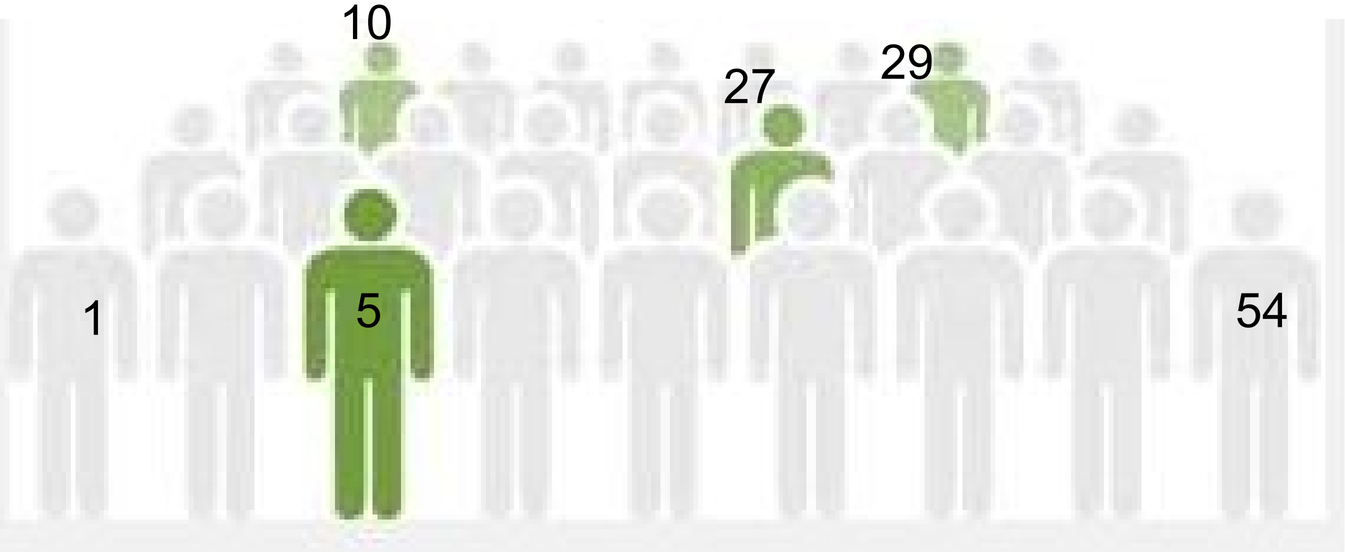
```
# Replace=T means, the same number can repeat.
# Replace=F means, each number can appear only once.
```

Toss a coin 10 times.

```
> sample(c("H","T"),  10,  replace=TRUE)
    T   H   H   H   T   H   T   H   T   T
```

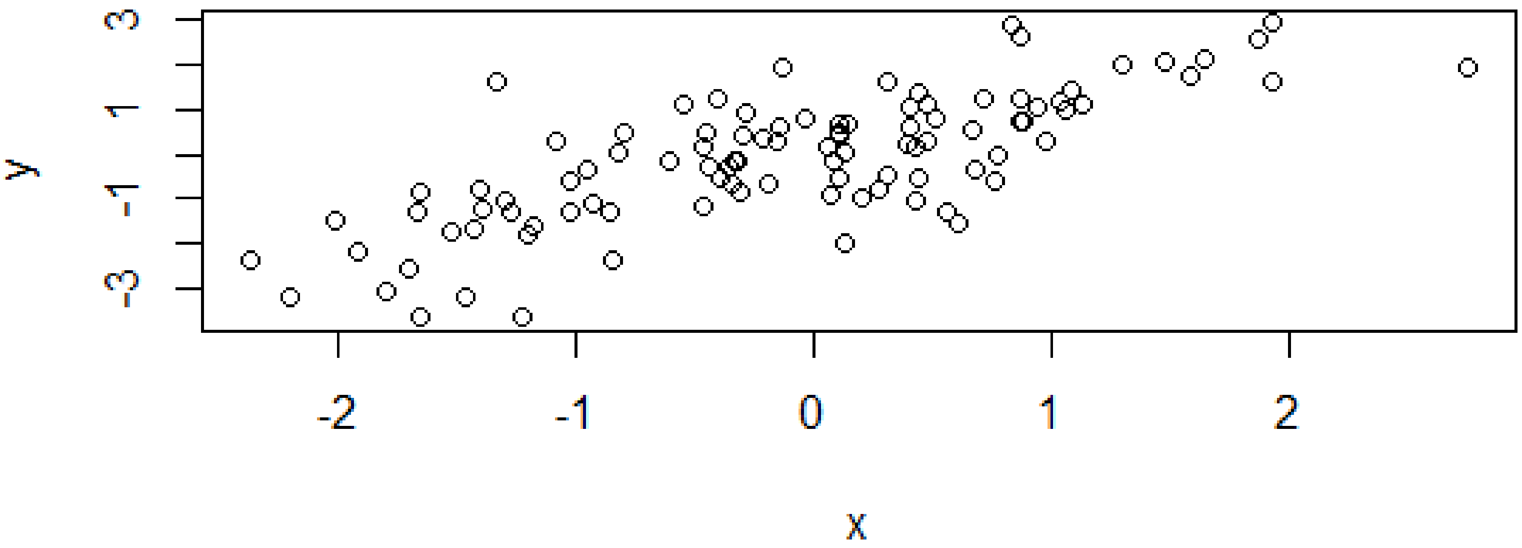
Select 4 different students from a class of 54 students

```
> sample(1:54, 4)  # default is no replacement
   27  5 10 29
```



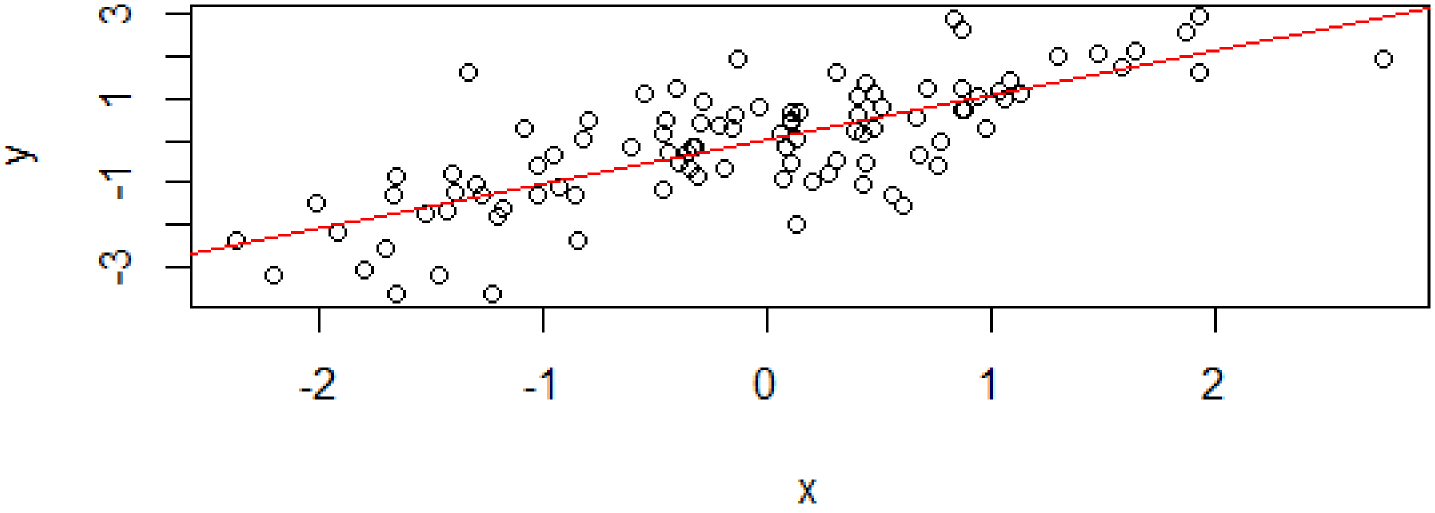
Scatter plot of two variables

```
# Generate 100 (x,y) pairs of random data
> x <- rnorm(100)
> y <- x + rnorm(100)
> plot( x,  y)
```



Add a Regression Line

```
> x <- rnorm(100)
> y <- x + rnorm(100)
> plot( x, y)
> abline( lm(y ~ x), col = "red" )
```



Statistical tests

```
> apropos(‘test’)  # See all the tests
> help(‘t.test’)      # details on t test
```

test if means of two groups are equal

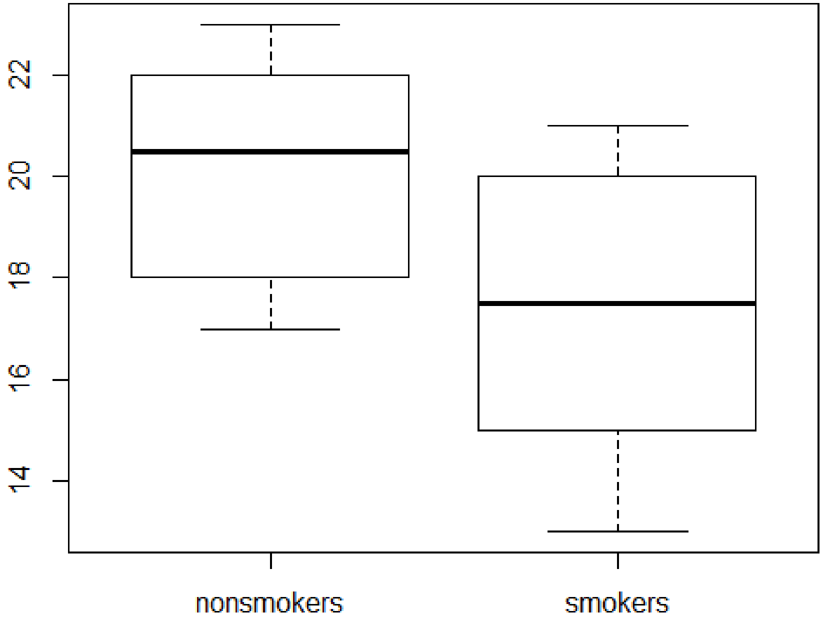
assume groups are normal with same var

null hypo: m1 = m2

alt hypo: m1 != m2 (2 tailed)

Example: Effect of smoking

```
# Performance before and after smoking.
> nonsmokers = c(18,22,21,17,20,17,23,20,22,21)
> smokers    = c(16,20,14,21,20,18,13,15,17,21)
> boxplot( nonsmokers, smokers)
```



t-test on data

```
> t.test( nonsmokers, smokers )
   Welch Two Sample t-test
data:  nonsmokers and smokers
   mean: 20.1 and 17.5
t=2.25, df=16.3, p-value=0.038
alt hypo: diff in means is not 0
95% confidence interval:  0.16 ..  5.03
```

Power of a test

```
> power.t.test(n = 20, delta = 1)
    Two-sample t test power calculation
    n = 20
    delta = 1
    sd = 1
    sig.level = 0.05
    power = 0.86
    alternative = two.sided
```

Other tests

```
> t.test( nonsmokers, smokers,
   alternative="greater",
   var.equal=T,
   paired=T        )
> wilcox.test(...)
> chisq.test( ... )  # Chi Square test
> aov( ... )  # Anova (Analysis of variance)
```

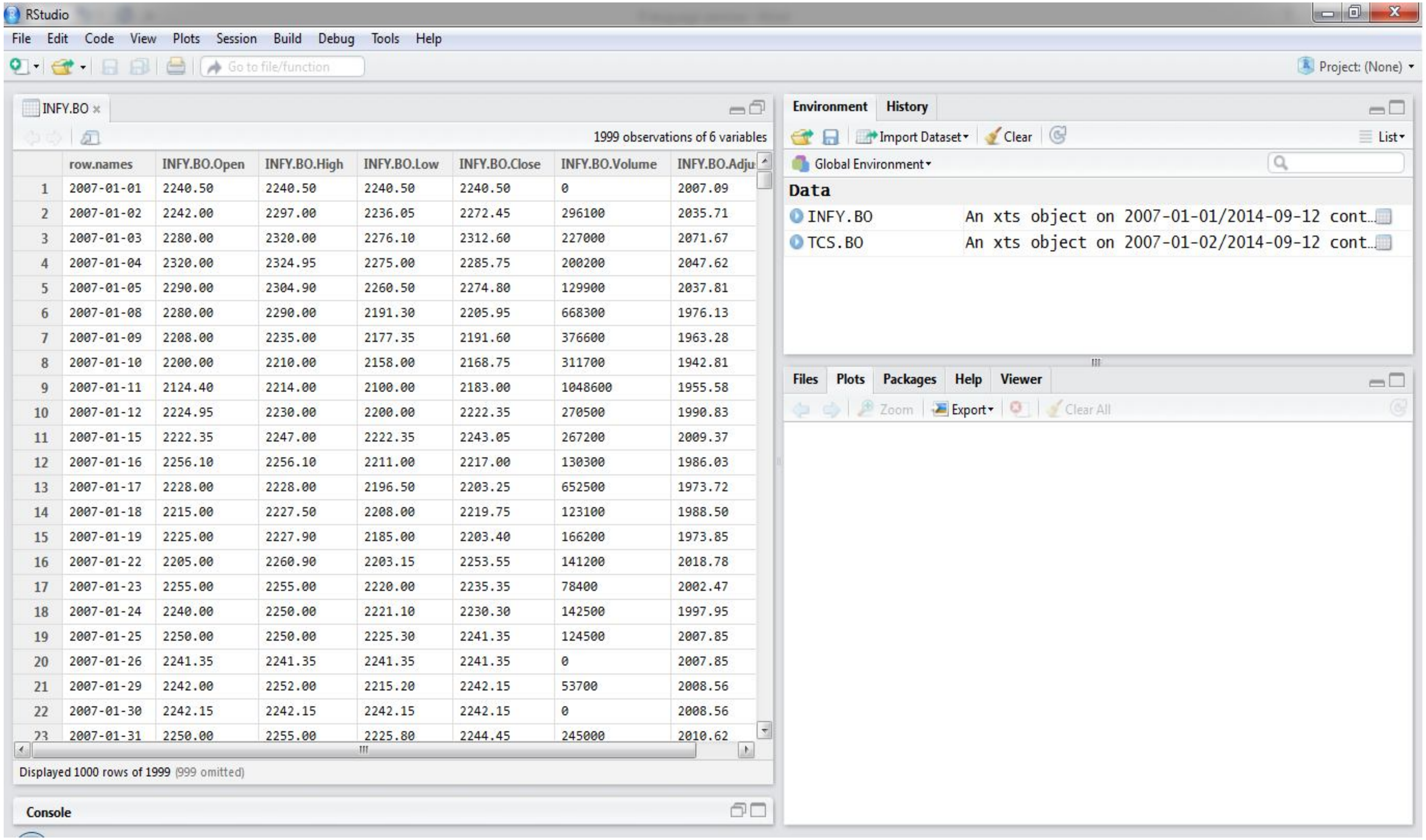
R for Finance

Part 4. By Amrutia Meet

To get the share quotes

- Get historical stock prices easily
- Opening prices, closing prices, day high, day low
- Get numerous stocks at a time
- In 2 line of R code
- Useful for automated daily computation

Stock prices in R studio



1999 observations of 6 variables

	row.names	INFY.BO.Open	INFY.BO.High	INFY.BO.Low	INFY.BO.Close	INFY.BO.Volume	INFY.BO.Adju
1	2007-01-01	2240.50	2240.50	2240.50	2240.50	0	2007.09
2	2007-01-02	2242.00	2297.00	2236.05	2272.45	296100	2035.71
3	2007-01-03	2280.00	2320.00	2276.10	2312.60	227000	2071.67
4	2007-01-04	2320.00	2324.95	2275.00	2285.75	200200	2047.62
5	2007-01-05	2290.00	2304.90	2260.50	2274.80	129900	2037.81
6	2007-01-08	2280.00	2290.00	2191.30	2205.95	668300	1976.13
7	2007-01-09	2208.00	2235.00	2177.35	2191.60	376600	1963.28
8	2007-01-10	2200.00	2210.00	2158.00	2168.75	311700	1942.81
9	2007-01-11	2124.40	2214.00	2100.00	2183.00	1048600	1955.58
10	2007-01-12	2224.95	2230.00	2200.00	2222.35	270500	1990.83
11	2007-01-15	2222.35	2247.00	2222.35	2243.05	267200	2009.37
12	2007-01-16	2256.10	2256.10	2211.00	2217.00	130300	1986.03
13	2007-01-17	2228.00	2228.00	2196.50	2203.25	652500	1973.72
14	2007-01-18	2215.00	2227.50	2208.00	2219.75	123100	1988.50
15	2007-01-19	2225.00	2227.90	2185.00	2203.40	166200	1973.85
16	2007-01-22	2205.00	2260.90	2203.15	2253.55	141200	2018.78
17	2007-01-23	2255.00	2255.00	2220.00	2235.35	78400	2002.47
18	2007-01-24	2240.00	2250.00	2221.10	2230.30	142500	1997.95
19	2007-01-25	2250.00	2250.00	2225.30	2241.35	124500	2007.85
20	2007-01-26	2241.35	2241.35	2241.35	2241.35	0	2007.85
21	2007-01-29	2242.00	2252.00	2215.20	2242.15	53700	2008.56
22	2007-01-30	2242.15	2242.15	2242.15	2242.15	0	2008.56
23	2007-01-31	2250.00	2255.00	2225.80	2244.45	245000	2010.62

Displayed 1000 rows of 1999 (999 omitted)

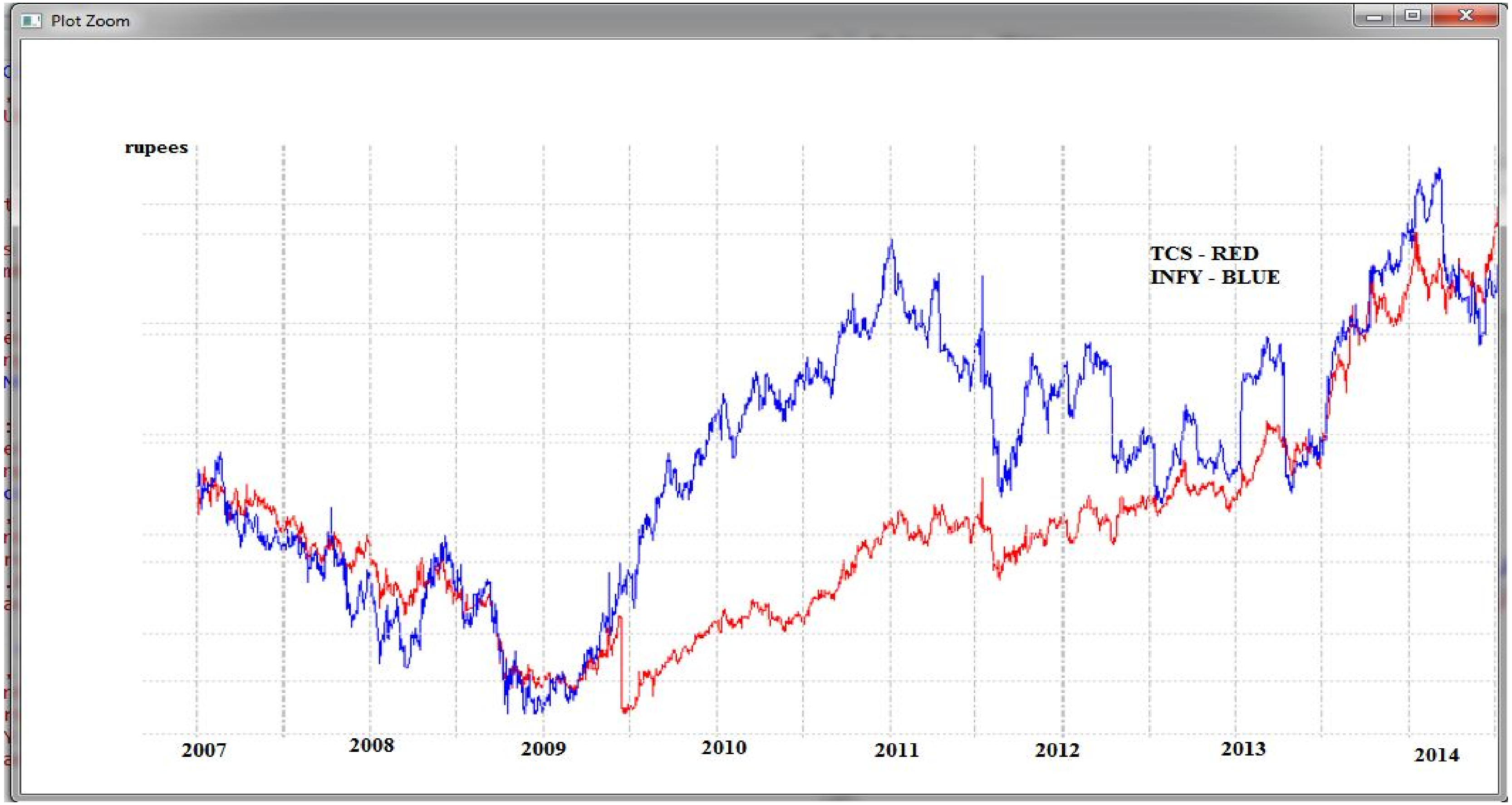
Data

INFY.BO	An xts object on 2007-01-01/2014-09-12 cont…
TCS.BO	An xts object on 2007-01-02/2014-09-12 cont…

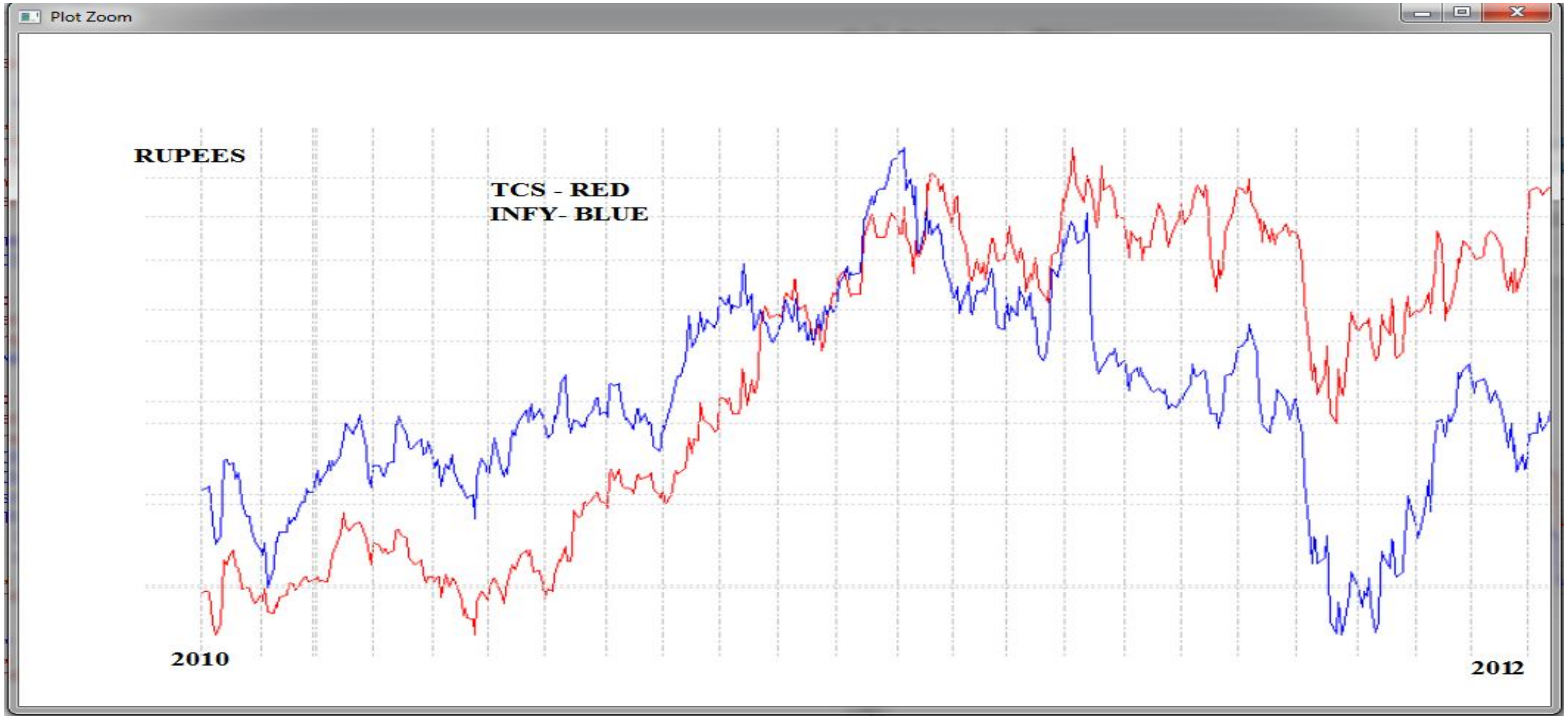
Merged Stock graphs

- Comparing two or more stock (script)
- Helps in getting the growth of the stock
- The return can be obtained from the graph

Infosys/TCS stock 2007 to 2014



Infosys/TCS stock 2010 to 2014



Computing stock CAGR (Compound Annual Return)

- To obtain Compound Annual Return for any stock (share) is easy in R.

Publications using R and Statistics

1. *“**Psychic Ads: Identifying Students for Higher Education by Analysing Google Trends Time Series for Targeted Advertising on Google Adwords**”, by* Kavya MN, Rai N, Mohsin A.
2. *“**Correlating Gender Sensitivity and Learning Traits in Higher Education**”, by* Kavya MN, Rao S, Joseph V, Mohsin A,
3. *“**A Statistical Approach to Modernize the Indian Higher Education System for Rural and Vernacular**”,*
   by Kavya MN, Jain S, K Vaishali, R. Krishnakumar, Mohsin.
4. *“**Exploratory Factor Analysis in R for MBA Students**”,*
   by J Monteiro, N Fernandes, Mohsin A .
5. “***Learning Financial Analysis in MBA with R.”***
   by Nayak L, Meet A, Mohsin A.

In *Nitte University, Fourth International Conference on Higher Education: Special Emphasis on Management Education, 2014*

MBA Projects using Statistics and R

1. **Market Research**: Factor Analysis in R.
2. **Marketing**: Time series analysis in R for predicting Ads.
3. **Finance**: Analysing Stock market with R

Questions for Student Presenters?

- Nikhita
- Jovita
- Kavya
- Nishita
- Laxmi
- Meet

Thank you.

References

1. Intro to R, Venables and Smith, http://cran.r-project.org/doc/manuals/R-intro.pdf http://cran.r-project.org/manuals.html
2. Basic Statistics tests in R, http://www.statmethods.net/stats/index.html
3. Advanced Probability/Statistics in R, http://zoonek2.free.fr/UNIX/48_R/all.html
4. More Statistics tests in R, http://www.ats.ucla.edu/stat/r/whatstat/whatstat.htm
5. 7 lectures on Financial Trading with R, http://www.rfortraders.com/