

R

Gang Chen
chengang@genomics.cn

October 31, 2015

Outline

- 1 Overview
- 2 Quick Get Started
- 3 Syntax
- 4 Object-Oriented Programming
- 5 Data Visualization

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

3 Syntax

4 Object-Oriented Programming

5 Data Visualization

Next

- 1 Overview
 - Data Analysis
 - Data Analysis and R
- 2 Quick Get Started
 - Hello R!
 - Development Environment
 - References
- 3 Syntax
 - Data Types
 - Programming Structures
 - Control Statements
 - Function
 - Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output
- 4 Object-Oriented Programming

Data Analysis

Wikipedia

Analysis of data is a process of inspecting, cleaning, transforming, and modeling data with the goal of discovering useful information, suggesting conclusions, and supporting decision making.

Data Analysis

Collecting → cleaning → transforming → modeling → visualizing

Biological Data Analysis

NGS and Complex Diseases

Sequencing → QC → Alignment and Variant Calling →
GWAS, EWAS ... → Manhattan Plot, Q-Q plot ...

Biological Data Analysis

NGS and Complex Diseases

Sequencing → QC → Alignment and Variant Calling →
GWAS, EWAS ... → Manhattan Plot, Q-Q plot ...
→ paper

Next

- 1 Overview
 - Data Analysis
 - Data Analysis and R
- 2 Quick Get Started
 - Hello R!
 - Development Environment
 - References
- 3 Syntax
 - Data Types
 - Programming Structures
 - Control Statements
 - Function
 - Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output
- 4 Object-Oriented Programming

What is R?

R

R is a free software environment for statistical computing and graphics.

----R-project.org

History

- April 1st, 1997, R0.16 , 奥克兰大学的Ihaka和Gentleman 发布了第一版本的R
- 1997年4月23日 , 0.49 , CRAN网站发布 , 提供12个R的扩展包
- 1997年12月5日 , 0.60 , R成文GNU项目的一部分
- 2000年2月29日 , 1.0 , 第一个可用于生产环境的版本发布
- 2010年4月22日 , 2.11 , 支持64位Windows操作系统
- 2011年10月31日 , 2.14 , 提供全新的并行计算包
- 2013年4月 , 3.0.0
- Now, 3.1.2

R语言在中国

- 2004年，国内专业人员开始翻译R语言官方文档
- 2006年，国内开始出版R语言书籍
- 2008年，在北京中国人民大学召开第一届中国R语言会议
- 2009年-2012年，每年分别在北京和上海举办中国R语言会议，迄今已举办五届
- 2012年，国人开发的Knitr包几乎成为R语言文档自动化的新标准，同时大量R语言畅销书籍被引进到国内翻译出版。
- 2013年，《R语言实战》、《ggplot2》、《R in a nutshell》 ...

R语言的现状

- 使用领域囊括统计分析、数据挖掘、生命科学、商业智能、数据可视化、社交网络分析、电子商务、集成电路、金融、烟草、传媒、咨询等
- 赞助R语言开发工作的机构包括AT&T、默沙东、Google、新西兰电信，以及诸多大学及科研机构。
- 在商业产品中提供R语言支持的企业包括SAP、甲骨文、Teradata、IBM、Revolution、Matlab、SAS、SPSS等。
- 2012第五届中国R语言会议（上海会场）获得大量赞助，吸引了400多人注册，到会人员几乎涉及R所有应用领域的国内知名企业。
- 2013年第六届中国R语言会议（北京，5月；上海，11月）。

Pros and Cons

The best thing about R is that it was developed by statisticians. The worst thing about R is that...it was developed by statisticians.

--- Bo Cowgill

Next

- 1 Overview
- 2 Quick Get Started
 - Hello R!
 - Development Environment
 - References
- 3 Syntax
- 4 Object-Oriented Programming
- 5 Data Visualization

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

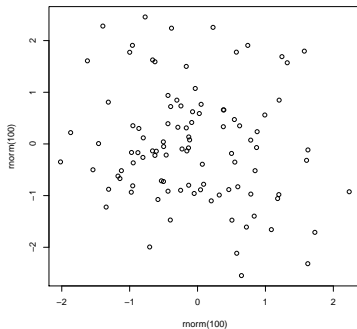
Hello R!

```
print("Hello R!")
```

```
## [1] "Hello R!"
```


Hello Plot

```
plot(rnorm(100), rnorm(100))
```



Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

Download and Installation

Download

CRAN

Installation

- R: Linux, Mac OS, Windows
- Rtools: Windows
- packages: CRAN, devtools, github, local file

Editors and IDEs

Editors

- R terminal
- Rgui
- VIM + Vim-R-plugin
- Emacs + ESS
- Notepad++ + NppToR
- ...

R Terminal and Rgui

R

- Ctrl + R: run
- Tab: auto complete
- arrow up and down: history

R and Texteditor

- copy and paste
- `source("source.R")`

source

```
sourceDir <- function(path, trace = TRUE, ...) {  
  for (nm in list.files(path, pattern = "[.] [RrSsQq]$")) {  
    if(trace) cat(nm, ":")  
    source(file.path(path, nm), ...)  
    if(trace) cat("\n")  
  }  
}
```

Quick Get Started Development Environment

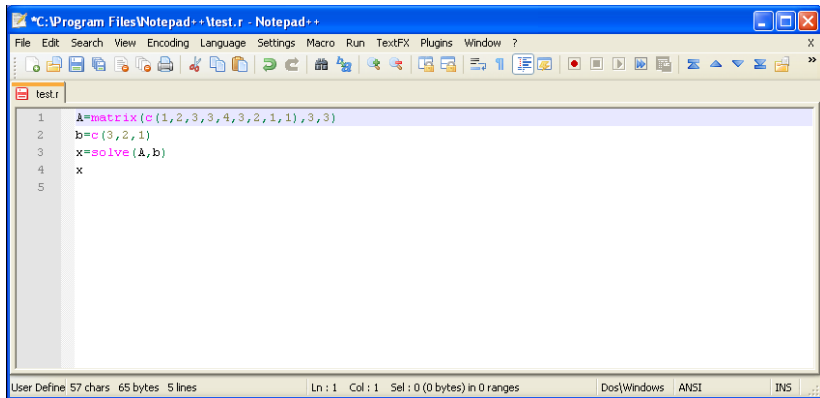
VIM + Vim-R-plugin

```

4. script2.R zzz.R RNA-Seq.R | xg9 0.1416941
| -fncsr-manual.k, Rowv = NULL, Colv = if (symm) "Rowv" else NULL, distfun, hclust | g10 0.5772262 0.3061073
| -fncsr-manual | > as.matrix(c)[1:4,1:4]
| -Rdlatex.log [Scratch] [Preview] 1,1 All | g1 1.0000000 -0.7240061 0.8050921 0.2327069
| -mypackage.Rcheatmap | # Row clustering | g2 -0.7240061 1.0000000 -0.5586679 -0.7823333
| -_o0_pkg_src/ hr <- hclust(as.dist(1-cor(t(y), method="pearson")), method="compl | g3 0.8050921 -0.5586679 1.0000000 0.2567203
| -_o0check.log # Column clustering | g4 0.2327069 -0.7823333 0.2567203 1.0000000
| -_o0install.out hc <- hclust(as.dist(1-cor(y, method="spearman")), method="comple | > y
| -mypackage.Rcheatmap | # Plot heatmap | t1 t2 t3 t4 t5
| -mypackage.Ex heatmap.2(y, Rowv=as.dendrogram(hr), Colv=as.dendrogram(hc), scale | g1 -0.2608109 -2.1287458 0.5436205 -0.1962956 0.5136432
| -mypackage.Ex # Return matrix with row/column sorting as in heatmap | g2 -2.0478162 -0.2318061 -2.1907113 -0.9185012 -1.1450074
| -mypackage/ y[rev(hr$labels),rev(hc$order)], hc$labels[hc$order]) | g3 -0.1814785 -0.5137189 1.2004188 -0.2185163 0.9562711
| -man/ heat.colors | g4 0.2493454 -0.5782053 0.7562372 -0.6143111 -1.0792957
| -colAg.Rd heat.colors function grDevices | g5 0.1082261 -1.8310231 -0.3319702 0.5535095 0.0165956
| -heatmap function stats cmle | g6 0.2596634 -0.8048402 -0.3751721 -0.6061217 -0.533725
| -R/ fmc[sdfset[[1]], sdfset[[2]], fast=T) | g7 0.4497986 -0.6475571 1.1905096 1.2794214 0.1432148
| -myFct.R result <- fmc[sdfset[[1]], sdfset[[2]]] | g8 -1.0501454 -0.3717143 0.2831488 -1.6238084 0.3429913
| -DESCRIPTION fcs <- fmc[sdfset[[1]], sdfset[[2]], au=2, bu=1, matching.mode="a | g9 -0.7831244 0.8490208 1.1253892 -0.4341535 0.6912465
| -NAMESPACE fcs | g10 -1.7273262 0.3621398 2.2920425 -0.9175735 -1.6735589
| -Read-and-del script2.R [*] 12,1 33>
| -fncsr-1.0.tar | heatmap.2 package:gplots R Documentation
| -jlitter.png n
| -matrix.xls | Enhanced Heat Map
| -myFct.R ## code chunk number 3: dist2
| -mypackage-1.0. notes.R c <- cor(t(y), method="pearson")
| -overlapper.R as.matrix(c)[1:4,1:4]
| -rangeoverlapp | Description:
| -RNA-Seq.R A heatmap is a false color image (basically 'image(t(x))') with
| -script1.R a
| -script2.R dendrogram added to the left side and/or to the top. Typically,
| -SDFstreamer.R reordering of the rows and columns according to some set of values
| -test.sdf es
| -test.svg (row or column means) within the restrictions imposed by the
| -tips_and_trick dendrogram is carried out.
| -zzz.R This heatmap provides a number of extensions to the standard R
| -zzz.Rd 'heatmap' function.
scripts/planning script1.R [*] 34,1 16>
-- Omni completion (%0-W-P) Back at original :
1:bash 2:mutt 3:col 4:screenshell * 5:rscrip 6:latex 7:bibtex 8:tasks- "Thomas-Girkes-MacBook-" 08:10 27-Jan-1

```

Notepad++ + NppToR



The screenshot shows the Notepad++ application window. The title bar reads "*C:\Program Files\Notepad++\test.r - Notepad++". The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, TextFX, Plugins, Window, and ?. The toolbar contains various icons for file operations, editing, and development. The editor area shows a file named "test.r" with the following R code:

```
1  A=matrix(c(1,2,3,3,4,3,2,1,1),3,3)
2  b=c(3,2,1)
3  x=solve(A,b)
4  x
5
```

The status bar at the bottom displays: "User Define 57 chars 65 bytes 5 lines", "Ln : 1 Col : 1 Sel : 0 (0 bytes) in 0 ranges", "Dos{Windows ANSI", and "INS".

Emacs + ESS

What is ESS?

ESS: Emacs Speak Statistics

The screenshot shows the Emacs editor interface with the ESS (Emacs Speak Statistics) package loaded. The main window displays R code and its output. The code includes loading the ESS package, loading the mtcars dataset, and performing various calculations. The output shows the first few rows of the mtcars dataset and the results of several arithmetic operations.

The side window, titled "R Graphics: D", displays a scatter plot of the mtcars dataset. The x-axis is labeled "mtcars\$wt" (weight) and the y-axis is labeled "mtcars\$wt" (weight). The plot shows a positive correlation between weight and another variable (likely mpg, though not explicitly labeled on the y-axis).

```

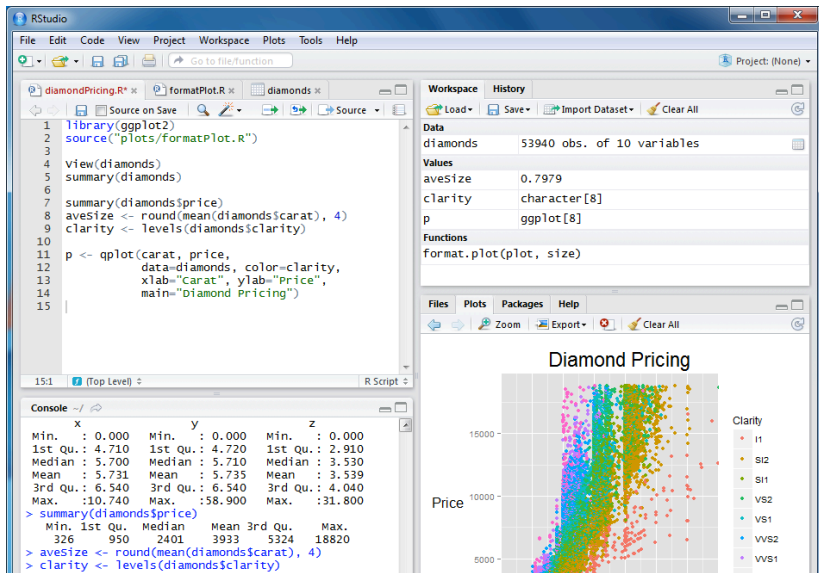
> library(ESSR)
> data(mtcars)
> colnames(mtcars)
[1] "mpg" "cyl" "d
[11] "carb"
> help(mtcars)
> 4300 * 3
[1] 12900
> 3179 * 120%
Error: unexpected input
> 3179 * 1.2
[1] 3814.8
> 3179 * 1.4
[1] 4450.6
> 3179 * 1.5
[1] 4768.5
> 3179 * 1.3
[1] 4132.7
> 3179 * 1.4
[1] 4450.6
> 4488 * 3
[1] 13464
> 3179 / 4488
[1] 0.7083333
> 5488 * 3
  
```


IDEs

IDEs

- RStudio: local and cloud-based
- TinnR
- StatET: eclipse for R
- ...

RStudio



Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

Books

- R in action (also in Chinese)
- Introduction to R (also in Chinese)
- R for beginner (also in Chinese)
- R in a Nutshell (also in Chinese)
- The art of R programming (also in Chinese)
- ggplot2. Elegant Graphics for Data Analysis (also in Chinese)

Websites

- R-project and CRAN
- COS.name (Chinese)
- Quick-R
- <http://had.co.nz/>, Hadley Wickham
- Twitter, github, RForge
- Google

Websites

- R-project and CRAN
- COS.name (Chinese)
- Quick-R
- <http://had.co.nz/>, Hadley Wickham
- Twitter, github, RForge
- Google Baidu?

Journals

- The R Journal
- Journal of Statistical Software

Next

1 Overview

2 Quick Get Started

3 **Syntax**

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

3 Syntax

- **Data Types**
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

Class, Type and Dimension

Class, Type and Dimension

Everything in R is a object, every object has class, type and dimension.

```
class(obj)  
typeof(obj)  
dim(obj)
```

Data Types

```
obj <- 1
class(obj)

## [1] "numeric"

obj <- "Gang Chen"
class(obj)

## [1] "character"

obj <- 1:3
class(obj)

## [1] "integer"

ranges <- GRanges(seqnames = c("chr1", "chr2"),
  ranges = IRanges(start = c(1013, 4351),
    end = c(2314, NA), width = c(NA, 1)),
  strand = c("+", "-"))
class(ranges)

## [1] "GRanges"
## attr(,"package")
## [1] "GenomicRanges"
```

```
class(list(a = 1, b = 2))

## [1] "list"

class(matrix(1:16, ncol=4))

## [1] "matrix"

class(array(1:64, c(4,4,4)))

## [1] "array"

obj <- as.data.frame(obj)
class(obj)

## [1] "data.frame"

obj <- as.factor(c("male", "female"))
class(obj)

## [1] "factor"
```

Types

```
obj <- 1
class(obj)
## [1] "numeric"
obj <- 1:3
class(obj)
## [1] "integer"
obj <- 1+2i
class(obj)
## [1] "complex"
```

Operations

Operators

- `+`, `-`, `*`, `/`, `==`, `=`, `<-`
- `^`
- `exp()`, `log()`, `log10()`, `log2()`
- `sqrt()`, `abs()`, `sin()`, `cos()`
- `round()`, `floor()`, `ceiling()`
- `factorial()`

Character

A character object is used to represent string values in R.

```
fname <- "Gang"  
lname <- "Chen"  
class(fname)  
  
## [1] "character"
```

```
myPI <- 3.14  
class(myPI)  
  
## [1] "numeric"  
  
myPI <- as.character(myPI)  
class(myPI)  
  
## [1] "character"
```

Character Operators

```
paste(fname, lname)
```

```
## [1] "Gang Chen"
```

```
substr("I am learning R", start=6, stop=13)
```

```
## [1] "learning"
```

```
sub("I am", "We are", "I am learning R")
```

```
## [1] "We are learning R"
```

Regular Expression

Regular Expressions == Problem

Some people,
when confronted with a problem,
think "I know, I'll use regular
expressions."
Now they have two problems.

Regular Expression in R

Regular Expression Functions

```
help(regex)  
grep(), grepl(), regexpr(), gregexpr(), sub(), gsub()
```

Example

```
grep("a.", c("Gang", "Chen", "aab", "Ag", "ga"))  
  
## [1] 1 3
```

Logical

```
u = TRUE; v = FALSE
```

```
u & v # u AND v
```

```
## [1] FALSE
```

```
u | v # u OR v
```

```
## [1] TRUE
```

```
!u # negation of u
```

```
## [1] FALSE
```

?

 $4.3 - 0.7$

[1] 3.6

 $4.3 - 0.7 == 3.6$

[1] FALSE

 $0.7 + 3.6 == 4.3$

[1] TRUE

 $4.2 / 6$

[1] 0.7

 $0.7 * 6$

[1] 4.2

 $4.2 / 6 == 0.7$

[1] FALSE

Vector

A vector is a sequence of data elements of the same basic type.

```
a = c(1,2,3)
```

```
b = c(T, F, F, T)
```

```
chars = c("Gang", "Chen", "AA", "Aa", "aB")
```

Arithmetic operations of vectors are performed memberwise.

All operators are applied to vectors

```
a^2
```

```
## [1] 1 4 9
```

```
!b
```

```
## [1] FALSE TRUE TRUE FALSE
```

```
grep("a.", chars)
```

```
## [1] 1 5
```

Vector Arithmetic

```
a = c(1,2,3,4,5)
```

```
b = c(5,4,3,2,1)
```

```
c(a, b)
```

```
## [1] 1 2 3 4 5 5 4 3 2 1
```

```
a + b
```

```
## [1] 6 6 6 6 6
```

```
a * b
```

```
## [1] 5 8 9 8 5
```

Recycling Rule:

```
d = c(1,2)
```

```
a + d
```

```
## Warning in a + d:
```

```
## [1] 2 4 4 6 6
```

Vector Index

```
a = c("one", "two", "three", "four", "five")
```

```
a[3]
```

```
## [1] "three"
```

```
a[2:4]
```

```
## [1] "two" "three" "four"
```

```
a[-3]
```

```
## [1] "one" "two" "four" "five"
```

```
a[8]
```

```
## [1] NA
```

Matrix Construction

```
mat = matrix(1:24, ncol=6, nrow=4, byrow=T)  
mat
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,]    1    2    3    4    5    6  
## [2,]    7    8    9   10   11   12  
## [3,]   13   14   15   16   17   18  
## [4,]   19   20   21   22   23   24
```


Matrix Index

```
mat[3,3]

## [1] 15

mat[2,]

## [1] 7 8 9 10 11 12

mat[,4]

## [1] 4 10 16 22
```

```
mat[2:3, 3:4]

##           [,1] [,2]
## [1,]         9  10
## [2,]        15  16

dim(mat)

## [1] 4 6

ncol(mat)

## [1] 6

nrow(mat)
```

Matrix Arithmetic

A

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

A * B

```
##      [,1] [,2] [,3] [,4]
## [1,]    1   25   81  169
## [2,]    4   36  100  196
## [3,]    9   49  121  225
## [4,]   16   64  144  256
```

B

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

A %*% B

```
##      [,1] [,2] [,3] [,4]
## [1,]   90  202  314  426
## [2,]  100  228  356  484
## [3,]  110  254  398  542
## [4,]  120  280  440  600
```

List

A list is a generic vector containing other objects.

```
n = c(2, 3, 5)
s = c("aa", "bb", "cc", "dd", "ee")
b = c(TRUE, FALSE, TRUE, FALSE, TRUE)
x = list(n, s, b, 3)
```

```
x
## [[1]]
## [1] 2 3 5
##
## [[2]]
## [1] "aa" "bb" "cc" "dd" "ee"
##
## [[3]]
## [1] TRUE FALSE TRUE FALSE TRUE
##
## [[4]]
## [1] 3
```

List Slice

```
x[1]
```

```
## [[1]]
```

```
## [1] 2 3 5
```

```
x[c(2,4)]
```

```
## [[1]]
```

```
## [1] "aa" "bb" "cc" "dd" "ee"
```

```
##
```

```
## [[2]]
```

```
## [1] 3
```

List Member

```
x[[3]]
```

```
## [1] TRUE FALSE TRUE FALSE FALSE
```

```
x[3]
```

```
## [[1]]
```

```
## [1] TRUE FALSE TRUE FALSE FALSE
```

Data Frame

A data frame is used for storing data tables. It is a list of vectors of equal length.

```
head(mtcars)
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear car
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4
## Datsun 710      22.8   4  108  93 3.85 2.320 18.61  1  1    4
## Hornet 4 Drive  21.4   6  258 110 3.08 3.215 19.44  1  0    3
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0
## Valiant         18.1   6  225 105 2.76 3.460 20.22  1  0    3
```

Data Frame

```
mtcars[1,2]
```

```
## [1] 6
```

```
mtcars["Mazda RX4", "wt"]
```

```
## [1] 2.62
```

```
ncol(mtcars)
```

```
## [1] 11
```

```
nrow(mtcars)
```

```
## [1] 32
```

Factor

```
gender <- c("male", "female")  
class(gender)
```

```
## [1] "character"
```

```
gender <- as.factor(gender)  
class(gender)
```

```
## [1] "factor"
```


Factor

```
group <- c(1, 2)
group[1] < group[2]

## [1] TRUE

class(group)

## [1] "numeric"

group <- as.factor(group)
group[1] < group[2]

## Warning in Ops.factor(group[1], group[2]): '<' not
## meaningful for factors

## [1] NA
```

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

3 Syntax

- Data Types
- **Programming Structures**
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

If else

```
if(something){  
    # do something  
}else if(something){  
    # do something  
}else{  
    # do something  
}
```

ifelse

```
ifelse(test, yes, no)
```

```
a <- c(2,3,4,2,5,6,7,12)  
ifelse(a%%2==0, a+1, 0)
```

```
## [1] 3 0 5 3 0 7 0 13
```

Loop

```
for (var in seq) expr  
while(cond) expr  
repeat  
break  
next
```

Loop

```
for(i in a){  
  if(i %% 2 == 0){  
    print(i + 1)  
  }else{  
    print(0)  
  }  
}
```

```
## [1] 3
```

```
## [1] 0
```

```
## [1] 5
```

```
## [1] 3
```

```
## [1] 0
```

```
## [1] 7
```

```
## [1] 0
```

apply functions

```
apply()  
lapply()  
sapply()  
tapply()
```

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

3 Syntax

- Data Types
- Programming Structures
 - Control Statements

• Function

- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

Function

```
add <- function(a, b){  
  a+b  
}  
add(1, 2)  
## [1] 3  
  
sapply(1:8, add, 3)  
## [1] 4 5 6 7 8 9 10 11
```

Anonymous Function

```
sapply(1:8, function(a, b){a+b}, 3)
```

```
## [1] 4 5 6 7 8 9 10 11
```

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- **Input and Output**
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

Standard I/O

```
scan()  
print()  
cat()
```

File I/O

Input

```
read.table()  
readLines()  
readChar()  
readBin()  
scan()
```

Output

```
write.table()  
write()
```

Database I/O

```
library(RMySQL) # for MySQL  
library(RPostgreSQL) # for PostgreSQL  
library(XLConnect) # for Excel
```

Next

- 1 Overview
- 2 Quick Get Started
- 3 Syntax
- 4 Object-Oriented Programming**
 - History
 - S3
 - S4
- 5 Data Visualization

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

S4 Classes and methods

History

- 1976, Rick Becker and John Chambers, S on Honeywell OS
- Ported to UNIX, S2
- Around 1986, functional programming and object self-description, S3
- 1992, concept of classes and methods, S4
- 2010, Reference Classes (RC), R 2.12

appendix in Software for Data Analysis by Chambers

S4 Classes and methods

OO Systems in R

- S3
- S4
- RC
- Base Types

Best Reference: <http://adv-r.had.co.nz/OO-essentials.html>

S3

S4 Classes and methods

S4 in R

```
library(stats4)
```

```
library(pryr)
```

```
## Error in library(pryr): there is no package called 'pryr'
```

```
y <- c(26, 17, 13, 12, 20, 5, 9, 8, 5, 4, 8)
```

```
nLL <- function(lambda) -sum(dpois(y, lambda, log = TRUE))
```

```
fit <- mle(nLL, start = list(lambda = 5), nobs = length(y))
```

```
isS4(fit)
```

```
## [1] TRUE
```

```
otype(fit)
```

```
## Error in eval(expr, envir, enclos): "otype"
```

```
isS4(nobs)
```

```
## [1] TRUE
```

S4 Classes and methods

Defining classes and creating objects

```
setClass("Person",  
  slots = list(name = "character", age = "numeric"))  
setClass("Employee",  
  slots = list(boss = "Person"),  
  contains = "Person")  
  
alice <- new("Person", name = "Alice", age = 40)  
john <- new("Employee", name = "John", age = 20, boss = alice)
```

S4 Classes and methods

access slots of an S4 object

```
alice@age  
slot(john, "boss")
```

S4 Classes and methods

Creating new methods and generics

```
setGeneric("union")
setMethod("union",
  c(x = "data.frame", y = "data.frame"),
  function(x, y) {
    unique(rbind(x, y))
  }
)
setGeneric("myGeneric", function(x) {
  standardGeneric("myGeneric")
})
```

Next

- 1 Overview
- 2 Quick Get Started
- 3 Syntax
- 4 Object-Oriented Programming
- 5 Data Visualization**
 - Scatter Plot
 - 时间序列
 - 柱状图
 - 饼图
 - 分类数据绘图

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

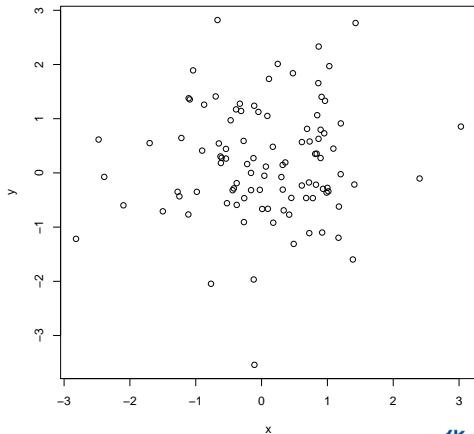
- Hello R!
- Development Environment
- References

3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

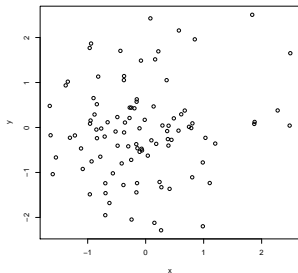
4 Object-Oriented Programming

plot



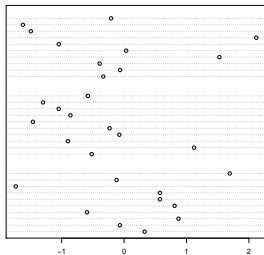
plot

```
x = rnorm(100)
y = rnorm(100)
plot(x, y)
```



dotchart

```
x = rnorm(30)
dotchart(x, groups = rep(1:3,10))
```



Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

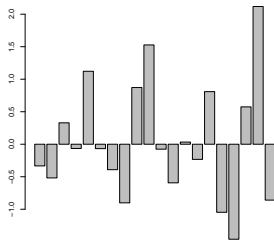
3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

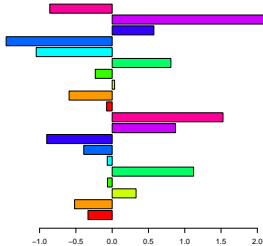
barplot

```
barplot(x[1:20])
```



barplot

```
barplot(x[1:20], width=2, horiz=T, col=rainbow(10))
```



Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

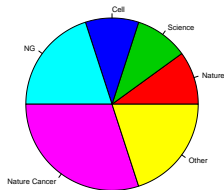
3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

pie

```
pie(c(10,10,10,20,30,20), c("Nature","Science","Cell","NG",  
Cancer","Other"),col=2:7)
```



pie

```
library(plotrix)

## Error in library(plotrix): there is no package
called 'plotrix'

pie3D(c(10,10,10,20,30,20), labels=c("Nature", "Science", "Ce
Cancer", "Other"), col=2:7)

## Error in eval(expr, envir, enclos):  "pie3D"
```

Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

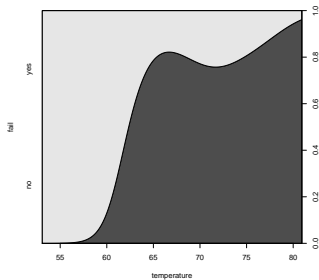
3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

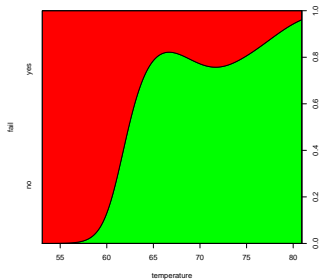
cdplot

```
cdplot(temperature, fail)
```



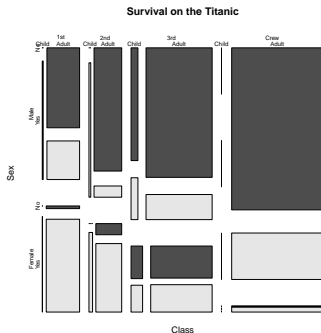
cdplot

```
cdplot(temperature, fail, col=c("green", "red"))
```



mosaicplot

```
require(stats)
mosaicplot(Titanic, main = "Survival on the Titanic",
color = TRUE)
```



Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

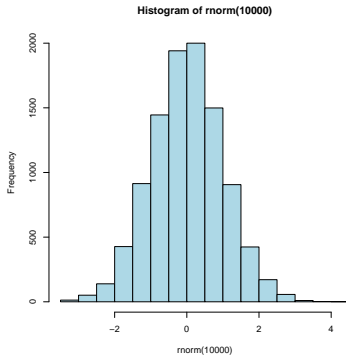
3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

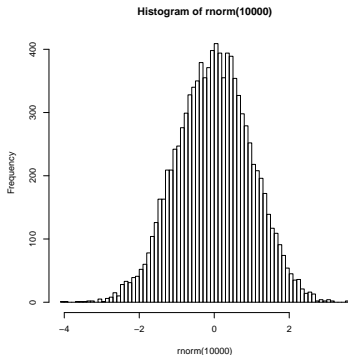
hist

```
hist(rnorm(10000), col="lightblue")
```



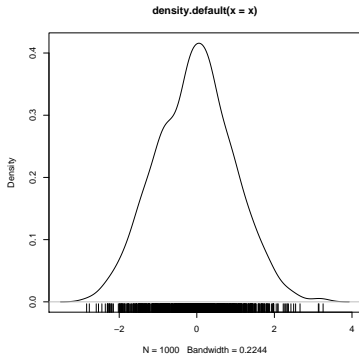
hist

```
hist(rnorm(10000), breaks=100)
```



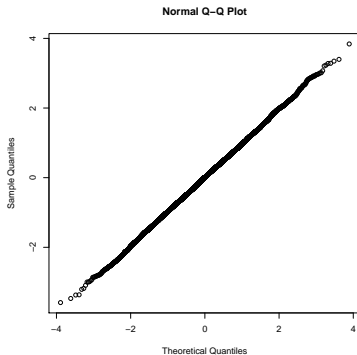
density + rug

```
x = rnorm(1000)
plot(density(x))
rug(x)
```



Q-Q plot

```
qqnorm(rnorm(10000))
```



Next

1 Overview

- Data Analysis
- Data Analysis and R

2 Quick Get Started

- Hello R!
- Development Environment
- References

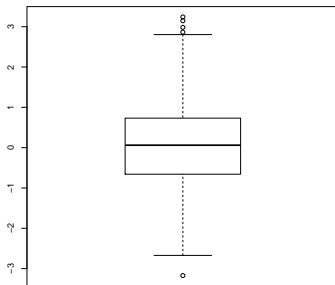
3 Syntax

- Data Types
- Programming Structures
 - Control Statements
- Function
- Input and Output
 - Standard Input and Output
 - File Input and Output
 - Database Input and Output

4 Object-Oriented Programming

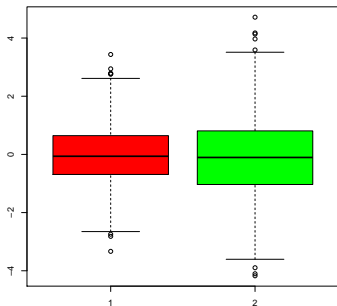
boxplot

```
boxplot(rnorm(1000))
```



boxplot

```
boxplot(cbind(rnorm(1000), rnorm(1000)+rnorm(1000)), col=c("red", "green"))
```



next

- R package
 - R package development
 - devtools
- Bioconductor
- Reproducible Research in R
- Advanced Topics
 - Machine Learning
 - Interactive Report
 - Big Data