零基础掌握 R 语言

pan junchang

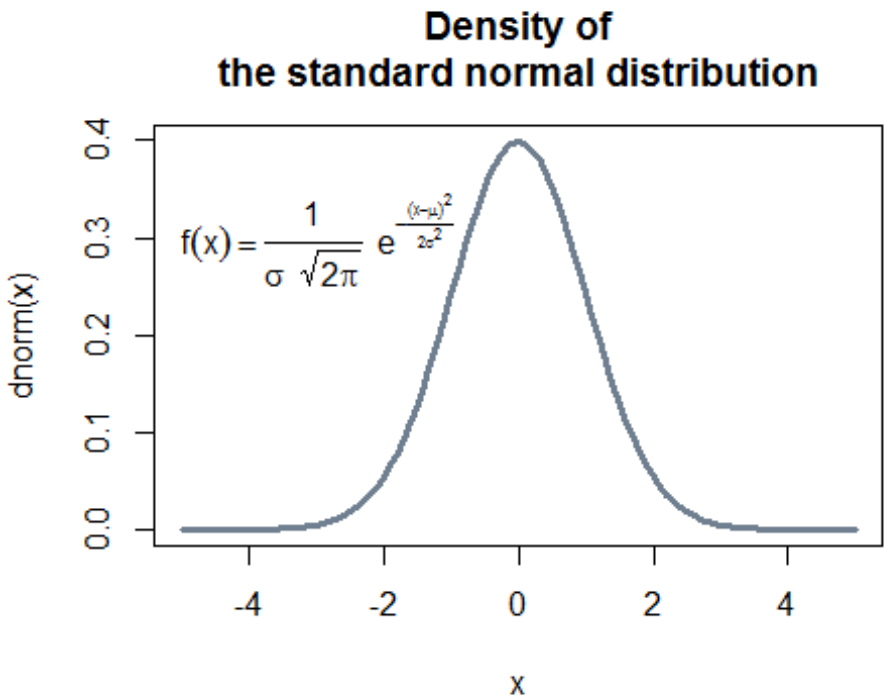
2015 年 8 月 10 日

R 语言简介

R 初步印象

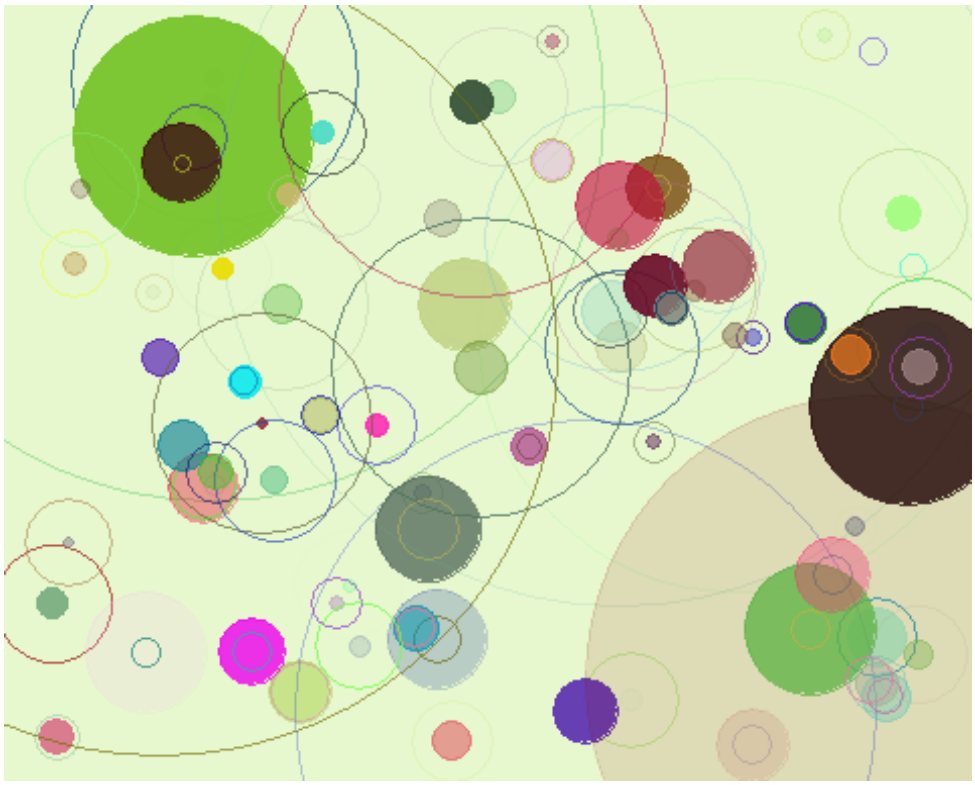
R 是由新西兰奥克兰大学的 Ross Ihaka 与 Robert Gentleman 共同开发的一个面向对象的编程语言,因为两人的名字都是以 R 开头，故命名 R 语言。R 语言的前身是 S 语言(由 AT&T Bell 实验室的 Rick Becker, John Chambers,Allan Wilks 开发)过去一直是数据分析领域里面的标准语言，但是现在正在逐步被 R 语言替代.R 语言可以按照使用者的要求完成各种计算问题,定义函数、绘制简单或复杂的图形,甚至可 以写自己的工具包。可以毫不夸张的说“它可以完成任何你能想到的计算问题”。 下面的这段代码可以画出一个漂亮的正太分布图.

```
curve(dnorm, from = -5, to = 5, col = "slategray", lwd = 3, main = "Den
sity of
the standard normal distribution")
text(-5, 0.3, expression(f(x) == frac(1,
                  sigma ~~sqrt(2*pi)) ~~ e^{-frac((x - mu)^2, 2*sigma^
2)}), adj = 0)
```



- 甚至可以绘制现代艺术图

```
par(mar = c(0, 0, 0, 0))
n = 76
set.seed(711)
plot.new()
size = c(replicate(n, 1/rbeta(2, 1.5, 4)))
center = t(replicate(n, runif(2)))
center = center[rep(1:n, each = 2), ]
color = apply(replicate(2 * n, sample(c(0:9, LETTERS[1:6]), 8,
      replace = TRUE)), 2, function(x) sprintf("#%s", paste(x,collapse
= "")))
points(center, cex = size, pch = rep(20:21, n), col = color)
```



代码来源:谢益辉 http://yihui.name/cn/2010/08/art-of-points-in-r/

再看一个例子。数据“基金业绩排名”中是 581 家基金 2012-2014 年收入统计表，现在希望 从中挑选出连续 3 年排名前 25 名的基金。

```
library(XLConnect)

## Loading required package: XLConnectJars
## XLConnect 0.2-11 by Mirai Solutions GmbH [aut],
##   Martin Studer [cre],
##   The Apache Software Foundation [ctb, cph] (Apache POI, Apache Comm
ons
##     Codec),
##   Stephen Colebourne [ctb, cph] (Joda-Time Java library)
## http://www.mirai-solutions.com ,
## http://miraisolutions.wordpress.com

library(dplyr)

##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
```

```
##
##     intersect, setdiff, setequal, union
```

```
mdt<-readWorksheetFromFile("mydata/基金业绩排名.xlsx", sheet = "Wind 资讯
",header=F,startRow=2,endCol=5)
names(mdt)<-c("code","名称","y2014","y2013","y2012")
tmp<-na.omit(mdt)
tmp<-tbl_df(tmp)
tmp%>%select(code,名称)->name
k<-25
tmp%>%select(code,y2014)%>%arrange(desc(y2014))%>%head(.,k)->tmp1
tmp%>%select(code,y2013)%>%arrange(desc(y2013))%>%head(.,k)->tmp2
tmp%>%select(code,y2012)%>%arrange(desc(y2012))%>%head(.,k)->tmp3
inner_join(name,tmp1,by="code")%>%inner_join(.,tmp2,by="code")%>%inner_
join(.,
                                                                tmp3,by="
code")
```

```
## Source: local data frame [3 x 5]
##
##        code          名称      y2014      y2013      y2012
##       (chr)        (chr)      (dbl)      (dbl)      (dbl)
## 1 340006.OF   兴全全球视野 1230838736 1455255022  459225241
## 2 510180.OF 华安上证 180ETF 6412945866  131040366 1086401656
## 3 360001.OF       光大核心 1430042630 1309488431  850849028
```

目前由 R 核心开发小组(R Development Core Team – 以后用 R DCT 表示)维 护， 他们完全自愿、 工作努力负责， 并将全球优 秀的统计应用软件打包提供给我们。

我们可以通 过 R 计划的网站(http://www.r-project.org)了解有 关 R 的最新信息和使用说明， 得到最新版本的 R 软件和基于 R 的应用统计软件包.

R 语言的特点与优势

- R 软件是一个完全免费的统计软件.
- 一个开源软件项目，得到了庞大用户社区的广泛采取.
- 站在巨人的肩膀上前进.
- R 是一个强大的脚本语言,语言简练高效.
- R 能够很好的与 LaTex 文档发布系统整合在一起，这意味着来自于 R 的统计输出和图形可 以嵌入到可出版级的文档中.

R 安装与工作环境

安装

单击下面的网址，选择适合自己电脑系统的版本进行安装

http://www.cran.r-project.org

为了提高易用性,推荐安装 Rstudio,打开下面的网址，点击 IDE,下载安装 RStudio Desktop.

(http://www.rstudio.com)

工作环境

工作空间（workspace）就是当前的 R 工作环境，存储用户定义的所有对象。函数、向量、 变量、矩阵、数据框等等都是对象。当你保存(“工作空间“)时，所有的对象都将被保存。

1. 打开 R 主程序

- 控制台(Console)

尝试输入如下代码：

```
demo(graphics)
```

- R 编辑器

点击菜单“文件--新建程序脚本”，将上述代码拷贝到新建脚本文件中，点击菜单“运行”。

2. 打开 Rstudio

- 脚本窗口
- Console 窗口,执行命令的地方
- 作图、帮助以及扩展包管理窗口

在 Consle 窗口中输入:
demo(graphics)
回车

在脚本窗口输入: demo(graphics)
点击“run”

在脚本窗口输入:
x<- 1:10
data(iris)

点击“run”

3. 工作目录

R 读取文件和设置文件的默认目录。

函数	功能

getwd()	返回当前工作目录
setwd()	设置默认工作目录
ls()	列出当前工作空间中的对象
save()	保存工作空间
load()	加载工作空间
rm()	删除内存中的对象
rm(list=ls())	删除内存中所有的对象
q()	退出

注意： *括号内路径需要用半角引号*

4. 扩展包的安装与加载

- 安装

(a) 方法一，在 Rstudio 右边中间位置点选"Package",点击 Install 就可以安装了。

(b) 方法二，直接在输入"install.packages('packagename')"来完成，在安装过程中如果需要其它扩展报支持，R 会自动安装，因此，安装扩展包时，最好连接网络。

- 加载

  load("packagename")

5. 帮助系统

help.start()

help(matrix)

?matrix

??matrix

example(sum)

练习

(1)安装并加载 quantmod 包；

(2)安装并加载 xts 和 xtsExtra 包。

R 语言数据类型与识别

R 语言 数据类型

Vectors

A vector is a sequence of data elements of the **same basic type**. Members in a vector are officially called components. Nevertheless, we will just call them members in this site.

```
v1<-c(2,3,4)#Creat a vector"v1" containing three numeric values 2, 3 an
d 5.
v2<-c("true","true","false")#Creat a vector"v2" contain character strin
gs.
v3<-c("aa", "bb", "cc", "dd", "ee") #Creat a vector"v3" contain charact
er strings.
```

Matrices

All columns in a matrix must have the **same mode** (numeric, character, etc.) and the same length. we usually creat a matices like this

```
mymatrix<-matix(vector,nrow=r,ncol=c,byrow=False,dimnames=list(
char_vector_rownames, char_vector_colnames))
```

byrow=TRUE indicates that the matrix should be filled by rows. byrow=FALSE indicates that the matrix should be filled by columns (the default). dimnames provides optional labels for the columns and rows. For example

```
cells <- c(1,26,24,68)
rnames <- c("R1", "R2")
cnames <- c("C1", "C2")
mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=TRUE,dimnames=list(rnam
es, cnames))
mymatrix

##    C1 C2
## R1  1 26
## R2 24 68
```

Data Frames

A data frame is more general than a matrix, in that **different columns can have different modes** (numeric, character, factor, etc.). One example of how to create a data frame is given below:

```
a <- c(1,2,3,4)
b <- c(2,4,6,8)
levels <- factor(c("A","B","A","B"))
```

```
mydf<-data.frame(first=a,second=b,f=levels)
mydf

##   first second f
## 1     1      2 A
## 2     2      4 B
## 3     3      6 A
## 4     4      8 B
```

Factors

The factor stores the nominal values as a vector of integers in the range [1... k] (where k is the number of unique values in the nominal variable), and an internal vector of character strings (the original values) mapped to these integers. For example

```
gender <- c(rep("male",20), rep("female", 30))
gender <- factor(gender)
```

This code generate a vector named "gender" with 20 "male" entries and 30 "female" entries

```
gender

##  [1] male   male   male   male   male   male   male   male   male
male
## [11] male   male   male   male   male   male   male   male   male
male
## [21] female female female female female female female female female
female
## [31] female female female female female female female female female
female
## [41] female female female female female female female female female
female
## Levels: female male

summary(gender)

## female   male
##     30     20
```

Numeric

数值型数据，是实数。可以写成整数(Integers)，小数(Decimal Fractions),或者科学技术(Scientific Notation)

Character

字符型数据。

Dates

日期型数据。日期型数据统称以字符串的形式输入到 R 中，然后通过 as.Date(x,format)函数进行转化为以数值形式存储的日期变量。

符号	含义
%d	数字表示的日期
%m	月份
%y	年份

example

```
dates <- c("02/27/92", "02/27/92", "01/14/92", "02/28/92", "02/01/92")
dates<-as.Date(dates, "%m/%d/%y")
weekdays(dates)

## [1] "星期四" "星期四" "星期二" "星期五" "星期六"

Sys.Date()-dates

## Time differences in days
## [1] 8637 8637 8681 8636 8663
```

list

列表变量。列表变量的元素可以是任何类型的数据。

R 语言常见数据类型的识别和转化

- 查看数据类型

mode(x) class(x)

- 数据类型识别与转换

类型	识别	转化
character	is.character()	as.character()
integer	is.integer()	as.integer()
logical	is.logical()	as.logical()
NA	is.NA()	as.NA()
numeric	is.numeric()	as.numeric()

数据输入与输出数据

R 语言读取 txt 格式数据

- 数据输入

read.table(file = "filename.txt",header = TRUE,sep="",fill=T) example

```
read.table(file="mydata/file1.txt")
read.table(file="mydata/file1.txt",header=T)
read.table(file="mydata/file2.txt",header=T,sep=",")
read.table(file="mydata/file3.txt",header=T,sep=" ")
read.table(file="mydata/file3.txt",header=T,sep=" ",fill=T)
```

同学可以思考下，同样的数据为什么第 4 句读不出来，而第 5 句能读出来？

- 数据输出

```
write.table(data,file='filename',col.names=F,sep='',quote = FALSE,
            append = FALSE, na = "NA")

write.table(file1,file="mydata/file3.txt",col.names=F,sep=" ",quote=F)
```

example

```
data(iris)
write.table(iris,file="mydata/file4.txt",row.names=F,sep=" ",quote=F)
```

R 语言 csv 格式数据

- 数据输入

read.csv("filename.csv")

example

```
read.csv("mydata/file6.csv")
x<-read.csv("mydata/file6.csv")
```

- 数据输出

```
  write.csv(x, file = "foo.csv",row.names = FALSE)
```

R 语言读取 excel 格式数据

- 数据输入

```
library(XLConnect)
readWorksheetFromFile("filename.xls", sheet = "sheet1",startRow = NULL,
endRow = NULL, startCol = NULL, endCol = NULL)
```

请同学们修改命令中的参数，练习读入“基金业绩排名.xlsx”

- 数据输出

```
writeWorksheetToFile("mydata/file5.xlsx", iris, sheet = "sheet1",header
=T)
```

R 语言读取 yahoo 网站证券交易数据

quantmod 包默认是访问 yahoo finance 的数据，其中包括上证和深证的股票数据，还有港股数据。 上证代码是 ss，深证代码是 sz，港股代码是 hk

```
library(quantmod)
```

```
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
##
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##     first, last
##
## Loading required package: TTR
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
setSymbolLookup(WK=list(name='000002.sz',src='yahoo'))
getSymbols("WK",from = "2015-06-01")
```

```
##     As of 0.4-0, 'getSymbols' uses env=parent.frame() and
##  auto.assign=TRUE by default.
##
##  This  behavior  will be  phased out in 0.5-0  when the call  will
##  default to use auto.assign=FALSE. getOption("getSymbols.env") and
##  getOptions("getSymbols.auto.assign") are now checked for alternate
defaults
##
##  This message is shown once per session and may be disabled by setti
ng
##  options("getSymbols.warning4.0"=FALSE). See ?getSymbols for more de
tails.
```

```
## Warning in download.file(paste(yahoo.URL, "s=", Symbols.name, "&a=",
## from.m, : downloaded length 5261 != reported length 200
```

```
## [1] "WK"
```

```
chartSeries(WK,theme='white')
```


WK
[2015-06-01/2015-10-20]
Last 13.55
16
15
14
13
800
600
400
200
0
Volume (millions):
40,724,600
六月 01 2015
七月 20 2015
九月 07 2015

```
addSMA(n=10)
```


WK
[2015-06-01/2015-10-20]
Last 13.55
16
15
14
13
800
600
400
200
0
Volume (millions):
40,724,600
六月 01 2015
七月 20 2015
九月 07 2015

R 语言获取 wind 数据

```
library('WindR')
w.start()
w.menu()
w.wsd("000012.SZ","sec_name,ipo_date,exch_city,open,high,low,close,volu
me,
dealnum","2015-08-25","2015-09-24","Fill=Previous;PriceAdj=B")

w.wss('000012.SZ','sec_name,ipo_date,exch_city,mkt')
```

获取多只股票 ipo 日期和地点

```
code<-read.table(file="mydata/stockID.txt",header=F,sep=" ",fill=T)
head(code)
mdt<-data.frame(id=1, numobs=1)[10,5 ]
for(i in 1:10){
tmp<-w.wss(code[i,2],'sec_name,ipo_date,exch_city,mkt')$Data
mdt<-rbind(mdt,tmp)
}
mdt
transform(mdt,IPO_DATE=w.asDateTime(mdt$IPO_DATE,asdate=T))
```

结构化网络数据抓取

```
library(XML)
url<-"http://www.sse.com.cn/market/"
tables<-readHTMLTable(url,stringsAsFactors=F,header=T)
tables[[6]]
```

数据整理

了解数据常用 R 语言函数：

函数	功能
head(x, n = 6L, ...)	产看前 6 个数据
tail(x, n = 6L, ...)	产看后 6 个数据
length(x)	产看一维向量元素的个数
dim(x)	产看二维向量维度
ncol(x)	返回二维向量 x 的列数
nrow(x)	返回二维向量 x 的行数
str(x)	返回对象 x 的结构
summary(x)	描述统计
sample()	随机抽样

- Examples

```r
data(iris)
head(iris)
```

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

```r
tail(iris)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

```r
length(iris)
```

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

```r
dim(iris)
```

```
## [1] 150   5
```

```r
ncol(iris)
```

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

```r
nrow(iris)
```

```
## [1] 150
```

```r
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1
1 1 1 1 1 1 1 ...
```

```r
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
```

```
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

```
iris[sample(nrow(iris),10),]
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 109          6.7         2.5          5.8         1.8  virginica
## 58           4.9         2.4          3.3         1.0 versicolor
## 10           4.9         3.1          1.5         0.1     setosa
## 97           5.7         2.9          4.2         1.3 versicolor
## 79           6.0         2.9          4.5         1.5 versicolor
## 113          6.8         3.0          5.5         2.1  virginica
## 12           4.8         3.4          1.6         0.2     setosa
## 3            4.7         3.2          1.3         0.2     setosa
## 135          6.1         2.6          5.6         1.4  virginica
## 111          6.5         3.2          5.1         2.0  virginica
```

sample(x,size)

```
sample(1:100,10)
```

```
##  [1] 17 50 44 70 67 77 61 10 81 41
```

R 语言选取数据子集

向量元素(vector)

- x[n] 返回向量 x 中第 n 个元素；例如：

```
  x<-c(8,7,6,5,4,3,2,1)
  x[2]
```

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

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

```
## [1] 7 5
```

- which | 返回满足条件的元素所在位置.例如

```
    x<-c(8,7,6,5,4,3,2,1)
    which(x>5)
```

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

```
    x[which(x>5)]
```

```
## [1] 8 7 6
```

矩阵元素(matrix)

```
  x.matr<-matrix(1:9,nrow=3)
  x.matr

##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9

  x.matr[2,1]

## [1] 2

  x.matr[2,2]

## [1] 5

  x.matr[2,]

## [1] 2 5 8

  x.matr[,2]

## [1] 4 5 6
```

列表元素（list）

```
a1<-"good morning!"
a2<-c(1,2,3,4,5)
a3<-matrix(1:9,nrow=3)
mylist<-list(a1,a2,a3)
mylist

## [[1]]
## [1] "good morning!"
##
## [[2]]
## [1] 1 2 3 4 5
##
## [[3]]
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9

mylist[[1]]

## [1] "good morning!"

mylist[[2]]
```

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

mylist[[3]]

##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9

mylist[[3]][3,2]

## [1] 6
```

数据框元素(data frame)

```
x1<-seq(1,10)
x2<-letters[1:10]
y<-seq(2,20,by=2)
mdt<-data.frame(x1,x2,y)
mdt

##    x1 x2  y
## 1   1  a  2
## 2   2  b  4
## 3   3  c  6
## 4   4  d  8
## 5   5  e 10
## 6   6  f 12
## 7   7  g 14
## 8   8  h 16
## 9   9  i 18
## 10 10  j 20

mdt[,1]

##  [1]  1  2  3  4  5  6  7  8  9 10

mdt[1,]

##   x1 x2 y
## 1  1  a 2

mdt['x1']

##    x1
## 1   1
## 2   2
## 3   3
## 4   4
## 5   5
## 6   6
## 7   7
## 8   8
```

```
## 9   9
## 10 10
```

```
mdt$x1
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
subset(mdt,select = c(2,3))
```

```
##    x2  y
## 1   a  2
## 2   b  4
## 3   c  6
## 4   d  8
## 5   e 10
## 6   f 12
## 7   g 14
## 8   h 16
## 9   i 18
## 10  j 20
```

```
subset(mdt,select=c(2,3),x1>8)
```

```
##    x2  y
## 9   i 18
## 10  j 20
```

R 语言数据合并

函数	功能
c()	向量合并
cbind()	列拼接
rbind()	行拼接
merge(x,y,by=...,by.x=...,by.y=...)	横向合并两个数据框.

- Example

```
a<-1:5
a
```

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

```
b<-6:10
b
```

```
## [1]  6  7  8  9 10
```

```
c<-11:15
c
```

```
## [1] 11 12 13 14 15
```

```
ab<-c(a,b)
ab

##  [1]  1  2  3  4  5  6  7  8  9 10

ab<-data.frame(a,b)
ab

##   a  b
## 1 1  6
## 2 2  7
## 3 3  8
## 4 4  9
## 5 5 10

ac<-data.frame(a,c)
ac

##   a  c
## 1 1 11
## 2 2 12
## 3 3 13
## 4 4 14
## 5 5 15

abc<-cbind(ab,ac)
abc

##   a  b a  c
## 1 1  6 1 11
## 2 2  7 2 12
## 3 3  8 3 13
## 4 4  9 4 14
## 5 5 10 5 15

abc<-merge(ab,ac,by='a')
abc

##   a  b  c
## 1 1  6 11
## 2 2  7 12
## 3 3  8 13
## 4 4  9 14
## 5 5 10 15
```

R 语言数据排序

- mdt[order(mdt$varable,decreasing=T),]
- Example

```
data("iris")
head(iris)
```

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

```
iris[order(iris$Sepal.Length,decreasing = F)[1:10],]
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 14          4.3         3.0          1.1         0.1  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 39          4.4         3.0          1.3         0.2  setosa
## 43          4.4         3.2          1.3         0.2  setosa
## 42          4.5         2.3          1.3         0.3  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 23          4.6         3.6          1.0         0.2  setosa
## 48          4.6         3.2          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
```

R 语言生成数据常用函数

seq(from,to,by/length) 生成等差序列向量

```
seq(from=1,to=10,by=2)
```

```
## [1] 1 3 5 7 9
```

```
seq(from=1,to=10,length=5)
```

```
## [1]  1.00  3.25  5.50  7.75 10.00
```

rep(x,times/each)重复 x 中的元素 time 次或者每个元素 each 次

```
rep(1:3,times=2)
```

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

```
rep(1:3,each=2)
```

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

生成零个观察值的数据框

```
mdt<-data.frame(age=numeric(0),gender=character(0),weight=numeric(0))
mdt
```

```
## [1] age    gender weight
## <0 rows> (or 0-length row.names)
```

R 语言数据编辑

手动编辑

- 用 edit()函数可以对数据进行手动编辑
- Example

```
a<-head(iris)
edit(a)
```

数值变换

- transform()函数用于变化数据中的对象，在数据框的数据结构中经常用到此函数。 Example

```
data("iris")
a<-head(iris)
a

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

a<-transform(a,Sepal.Length=-Sepal.Length,sum=Sepal.Length+Sepal.Width)
a

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species sum
## 1         -5.1         3.5          1.4         0.2  setosa 8.6
## 2         -4.9         3.0          1.4         0.2  setosa 7.9
## 3         -4.7         3.2          1.3         0.2  setosa 7.9
## 4         -4.6         3.1          1.5         0.2  setosa 7.7
## 5         -5.0         3.6          1.4         0.2  setosa 8.6
## 6         -5.4         3.9          1.7         0.4  setosa 9.3
```

修改变量名称

- names()函数返回变量名称
- Example

```
names(a)

## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
## [5] "Species"      "sum"

names(a)<-LETTERS[1:6]
a
```

```
##      A   B   C   D      E   F
## 1 -5.1 3.5 1.4 0.2 setosa 8.6
## 2 -4.9 3.0 1.4 0.2 setosa 7.9
## 3 -4.7 3.2 1.3 0.2 setosa 7.9
## 4 -4.6 3.1 1.5 0.2 setosa 7.7
## 5 -5.0 3.6 1.4 0.2 setosa 8.6
## 6 -5.4 3.9 1.7 0.4 setosa 9.3
```

```
names(a)<-paste("x",1:6,sep="")
a
```

```
##     x1  x2  x3  x4     x5  x6
## 1 -5.1 3.5 1.4 0.2 setosa 8.6
## 2 -4.9 3.0 1.4 0.2 setosa 7.9
## 3 -4.7 3.2 1.3 0.2 setosa 7.9
## 4 -4.6 3.1 1.5 0.2 setosa 7.7
## 5 -5.0 3.6 1.4 0.2 setosa 8.6
## 6 -5.4 3.9 1.7 0.4 setosa 9.3
```

删除含有缺失值的行

`na.omit()` 将会删除数据中有缺失值的观测样本（行）

- Example

```
mt<-read.table(file="mydata/file3.txt",header=T,sep=" ",fill=T)
mt
```

```
##   X1  name age height
## 1  2  John  10    150
## 2  3  Jack  27    180
## 3  4  Mary  29    167
## 4  5   DDD  NA     NA
## 5  6 Smith  23    164
```

```
mt<-na.omit(mt)
mt
```

```
##   X1  name age height
## 1  2  John  10    150
## 2  3  Jack  27    180
## 3  4  Mary  29    167
## 5  6 Smith  23    164
```

将数值型变量根据大小转化成因子变量

- cut()函数将数值变换成因子，是生产因子的一种常用方法。函数形式为：

```
cut(x,breaks,include.lowest=F)
```

```r
data(iris)
a<-iris[1:50,]
cut(a$Sepal.Length,breaks=4.2,to=5.9,length=5)
```

```
##  [1] (5.05,5.42] (4.67,5.05] (4.67,5.05] (4.3,4.67]  (4.67,5.05]
##  [6] (5.05,5.42] (4.3,4.67]  (4.67,5.05] (4.3,4.67]  (4.67,5.05]
## [11] (5.05,5.42] (4.67,5.05] (4.67,5.05] (4.3,4.67]  (5.42,5.8]
## [16] (5.42,5.8]  (5.05,5.42] (5.05,5.42] (5.42,5.8]  (5.05,5.42]
## [21] (5.05,5.42] (5.05,5.42] (4.3,4.67]  (5.05,5.42] (4.67,5.05]
## [26] (4.67,5.05] (4.67,5.05] (5.05,5.42] (5.05,5.42] (4.67,5.05]
## [31] (4.67,5.05] (5.05,5.42] (5.05,5.42] (5.42,5.8]  (4.67,5.05]
## [36] (4.67,5.05] (5.42,5.8]  (4.67,5.05] (4.3,4.67]  (5.05,5.42]
## [41] (4.67,5.05] (4.3,4.67]  (4.3,4.67]  (4.67,5.05] (5.05,5.42]
## [46] (4.67,5.05] (5.05,5.42] (4.3,4.67]  (5.05,5.42] (4.67,5.05]
## Levels: (4.3,4.67] (4.67,5.05] (5.05,5.42] (5.42,5.8]
```

```r
b<-cut(a$Sepal.Length,breaks=4.2,to=5.9,length=5)
b
```

```
##  [1] (5.05,5.42] (4.67,5.05] (4.67,5.05] (4.3,4.67]  (4.67,5.05]
##  [6] (5.05,5.42] (4.3,4.67]  (4.67,5.05] (4.3,4.67]  (4.67,5.05]
## [11] (5.05,5.42] (4.67,5.05] (4.67,5.05] (4.3,4.67]  (5.42,5.8]
## [16] (5.42,5.8]  (5.05,5.42] (5.05,5.42] (5.42,5.8]  (5.05,5.42]
## [21] (5.05,5.42] (5.05,5.42] (4.3,4.67]  (5.05,5.42] (4.67,5.05]
## [26] (4.67,5.05] (4.67,5.05] (5.05,5.42] (5.05,5.42] (4.67,5.05]
## [31] (4.67,5.05] (5.05,5.42] (5.05,5.42] (5.42,5.8]  (4.67,5.05]
## [36] (4.67,5.05] (5.42,5.8]  (4.67,5.05] (4.3,4.67]  (5.05,5.42]
## [41] (4.67,5.05] (4.3,4.67]  (4.3,4.67]  (4.67,5.05] (5.05,5.42]
## [46] (4.67,5.05] (5.05,5.42] (4.3,4.67]  (5.05,5.42] (4.67,5.05]
## Levels: (4.3,4.67] (4.67,5.05] (5.05,5.42] (5.42,5.8]
```

```r
table(b)
```

```
## b
##  (4.3,4.67] (4.67,5.05] (5.05,5.42]  (5.42,5.8]
##           9          19          17           5
```

```r
b<-factor(b,labels=c("小","中","大","特大"))
table(b)
```

```
## b
##   小   中   大 特大
##    9   19   17    5
```

R 语言常用数学运算

R 语言数值运算

功能	运算符号
加法	+

减法	-
乘法	*
除法	/
相除取整	%/%
乘幂	^

R 语言逻辑运算

功能	运算符号
判断是否相等	==
判断是否小于等于	<=
判断是否大于等于	>=
逻辑“与”	&
逻辑“非”	!=

R 语言常用数学运算

功能	运算符号
sum(x)	向量 x 之和
cumsum(x)	向量 x 累积和
prod(x)	向量的积
cumprod(x)	向量的累积积
log(x)	计算向量 x 的自然对数
exp(x)	计算自然数的 x 次方

- Example1

计算万科股票的收益率算术平均值和几何平均值

```
library(quantmod)
 setSymbolLookup(WK=list(name='000002.sz',src='yahoo'))
 getSymbols("WK",from = "2015-06-01")

## Warning in download.file(paste(yahoo.URL, "s=", Symbols.name, "&a=",
## from.m, : downloaded length 5261 != reported length 200

## [1] "WK"

close<-WK[,6]
ret<-dailyReturn(WK)
n<-length(ret)
arithemetic.average<-sum(ret)/n
arithemetic.average

## [1] -0.0001654687
```

```
gemetric.avgerage<-prod(ret+1)^(1/n)-1
gemetric.avgerage

## [1] -0.000582691
```

- Example2

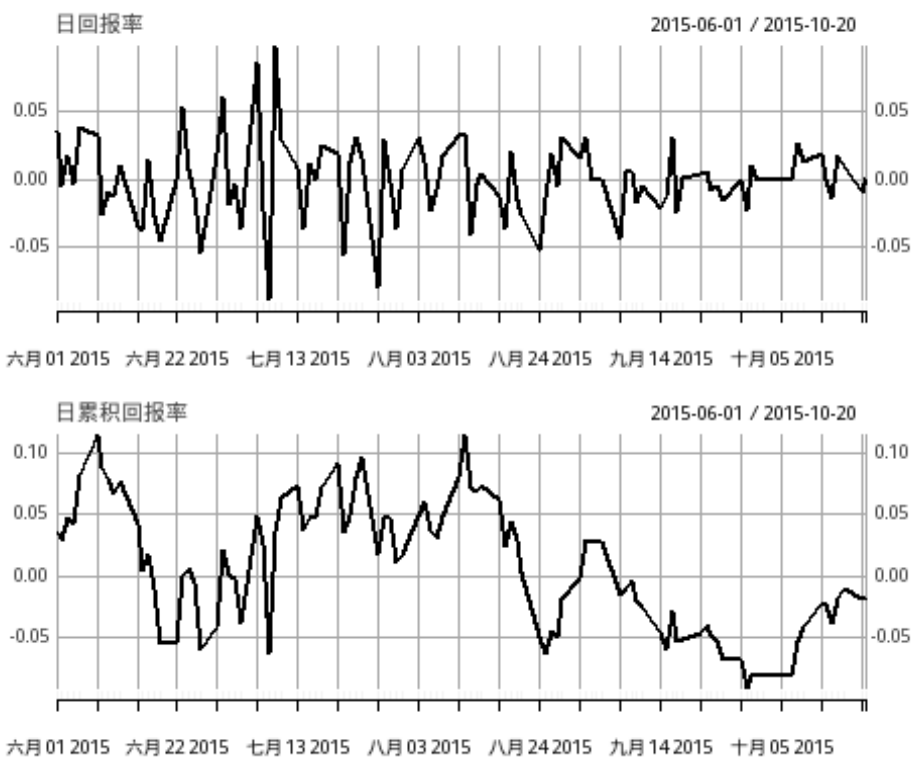
计算累积日回报率

```
library(quantmod)
 setSymbolLookup(WK=list(name='000002.sz',src='yahoo'))
 getSymbols("WK",from = "2015-06-01")

## Warning in download.file(paste(yahoo.URL, "s=", Symbols.name, "&a=",
## from.m, : downloaded length 5261 != reported length 200

## [1] "WK"

close<-WK[,6]
ret<-dailyReturn(WK)
cumret<-cumsum(ret)
devi<-par(mfrow = c(2,1), mar = c(3, 3, 2, 1))
plot(ret,main = "日回报率")
plot(cumret,main="日累积回报率")
```




```
par(devi)
```

R 语言常用统计函数

函数	描述
mean(x)	平均数
median(x)	中位数
sd(x)	标准差
var(x)	方差
quantile(x)	分位数
range(x)	求取值范围
max(x)	求最大值
min(x)	求最小值

R 语言高级技巧

R 语言的自编函数

R 语言中可以自己写函数，来完成特定的任务。编写函数的一般语法：

```
function.name<-function(argument1,argument2,...){
statements
return(object)
}
```

各部分内容含义如下：

项目	含义
function.name	所建函数的名称
argument	函数参数，运算时参数数值将传递到函数内部参与运算
statements	函数体内部的运算过程
return(object)	最终输出的对象

- Example1

编写一个计算圆面积的函数

```
myfunc1<-function(x){
res<-pi*x^2
return(res)
}
myfunc1(2)

## [1] 12.56637
```

- Example2

计算资金流现值

```
mypvf<-function(cf,r){
n<-length(cf)
tmp1<-1:n
cf.pv<-cf/(1+r)^tmp1
pv<-sum(cf.pv)
return(pv)
}

cf<-c(rep(10,4),110)
r<-0.1
mypvf(cf,r)

## [1] 100
```

条件结构

if/else 语句

```
if (cond) expression
```

如果条件成立(condition 为 True)则执行表达式，否则跳过；

```
if (cond) expression1 else expression2
```

如果条件成立则执行表达式 1，否则执行表达式 2

ifelse()函数

```
ifelse(condition,expression1,expression2)
```

如果条件 condition 成立，则执行表达式 1，否则执行表达式 2

switch 语句

```
switch(expression,list)
```

如果 switch 等于列表分量名，则返回分量对应的值

```
js1<-function(x,y,type){
switch(type,
jia=x+y,
jian=x-y,
chu=x/y,
cheng=x*y
)
}
js1(4,2,"jia")

## [1] 6
```

```
js1(4,2,"jian")

## [1] 2

js1(4,2,"cheng")

## [1] 8

js1(4,2,"chu")

## [1] 2
```

如果 switch 在 1 到 length(list)之间，最大返回列表相应位置的值。

```
js2<-function(x,y,list){
switch(list,
x+y,
x-y,
x*y,
x/y
)
}
js2(4,2,1)

## [1] 6

js2(4,2,2)

## [1] 2

js2(4,2,3)

## [1] 8

js2(4,2,4)

## [1] 2
```

循环结构

- for 语句

```
for(循环变量 in expression1){
expression2
}
```

expression1 是一个向量表达式，比如 1:5,2:15,expression2 通常是一组表达式。表示循环重复执行 expression2,直到循环变量的值不再包含在 expression1(序列)里为止。

Example

```
n<-0
for(i in 1:100){
```

```
  n<-n+i
}
n

## [1] 5050
```

- while 语句

```
while (condition) expression
```

当条件(condition)成立时，执行 expression，条件不满足时不执行。

```
x<-0
while(x < 5) {
x <- x+1
print(x)
}

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

x<-0
while(x < 5) {x <- x+1; if (x == 3) next; print(x);}

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

repeat 语句

```
repeat{
  expression...
  if(condition){
    break
  }
}
```

重复 expression 直到 break 语句

```
x<-0
repeat{
x<-x+1
print(x)
  if(x>5){
    break
  }
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
```

dplyr 工具包

dplyr 工具包是一个用 C++编写的工具包，主要用于数据运算。其特点是速度快，语法简洁。

观测值筛选(筛选行)

语法：

```
filter(mdt,condition1,condition2)
```

- Example

```
library(dplyr)
data("iris")
mdt<-tbl_df(iris)
mdt

## Source: local data frame [150 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## ..          ...         ...          ...         ...     ...

filter(mdt,Species=="setosa")

## Source: local data frame [50 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
```

```
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## ..          ...         ...          ...         ...     ...
```

```
filter(mdt,Species=="setosa",Petal.Width==0.2)
```

```
## Source: local data frame [29 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.0         3.4          1.5         0.2  setosa
## 7           4.4         2.9          1.4         0.2  setosa
## 8           5.4         3.7          1.5         0.2  setosa
## 9           4.8         3.4          1.6         0.2  setosa
## 10          5.8         4.0          1.2         0.2  setosa
## ..          ...         ...          ...         ...     ...
```

变量筛选(筛选列)

语法：

```
select(data,ColumnName1，ColumnName2)
```

- Example

```
library(dplyr)
data("iris")
mdt<-tbl_df(iris)
mdt
```

```
## Source: local data frame [150 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
```

```
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## ..          ...         ...          ...         ...     ...
```

```
select(mdt,Species,Sepal.Length)
```

```
## Source: local data frame [150 x 2]
## 
##    Species Sepal.Length
##     (fctr)        (dbl)
## 1   setosa          5.1
## 2   setosa          4.9
## 3   setosa          4.7
## 4   setosa          4.6
## 5   setosa          5.0
## 6   setosa          5.4
## 7   setosa          4.6
## 8   setosa          5.0
## 9   setosa          4.4
## 10  setosa          4.9
## ..     ...          ...
```

```
select(mdt,-Sepal.Length)
```

```
## Source: local data frame [150 x 4]
## 
##    Sepal.Width Petal.Length Petal.Width Species
##          (dbl)        (dbl)       (dbl)  (fctr)
## 1          3.5          1.4         0.2  setosa
## 2          3.0          1.4         0.2  setosa
## 3          3.2          1.3         0.2  setosa
## 4          3.1          1.5         0.2  setosa
## 5          3.6          1.4         0.2  setosa
## 6          3.9          1.7         0.4  setosa
## 7          3.4          1.4         0.3  setosa
## 8          3.4          1.5         0.2  setosa
## 9          2.9          1.4         0.2  setosa
## 10         3.1          1.5         0.1  setosa
## ..         ...          ...         ...     ...
```

```
select(mdt,-Sepal.Length,-Sepal.Width)
```

```
## Source: local data frame [150 x 3]
## 
##    Petal.Length Petal.Width Species
##           (dbl)       (dbl)  (fctr)
## 1           1.4         0.2  setosa
## 2           1.4         0.2  setosa
## 3           1.3         0.2  setosa
## 4           1.5         0.2  setosa
## 5           1.4         0.2  setosa
```

```
## 6           1.7         0.4  setosa
## 7           1.4         0.3  setosa
## 8           1.5         0.2  setosa
## 9           1.4         0.2  setosa
## 10          1.5         0.1  setosa
## ..          ...         ...     ...
```

```
select(mdt,length=Sepal.Length,width=Sepal.Width)
```

```
## Source: local data frame [150 x 2]
##
##    length width
##     (dbl) (dbl)
## 1     5.1   3.5
## 2     4.9   3.0
## 3     4.7   3.2
## 4     4.6   3.1
## 5     5.0   3.6
## 6     5.4   3.9
## 7     4.6   3.4
## 8     5.0   3.4
## 9     4.4   2.9
## 10    4.9   3.1
## ..    ...   ...
```

数据排序

语法：

```
arrange(data,ColunmName1,ColunmName2)
```

- Example

```
arrange(mdt,Sepal.Length)
```

```
## Source: local data frame [150 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           4.3         3.0          1.1         0.1  setosa
## 2           4.4         2.9          1.4         0.2  setosa
## 3           4.4         3.0          1.3         0.2  setosa
## 4           4.4         3.2          1.3         0.2  setosa
## 5           4.5         2.3          1.3         0.3  setosa
## 6           4.6         3.1          1.5         0.2  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           4.6         3.6          1.0         0.2  setosa
## 9           4.6         3.2          1.4         0.2  setosa
## 10          4.7         3.2          1.3         0.2  setosa
## ..          ...         ...          ...         ...     ...
```

```
arrange(mdt,desc(Sepal.Length),desc(Sepal.Width ))
```

```
## Source: local data frame [150 x 5]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
##           (dbl)       (dbl)        (dbl)       (dbl)    (fctr)
## 1           7.9         3.8          6.4         2.0 virginica
## 2           7.7         3.8          6.7         2.2 virginica
## 3           7.7         3.0          6.1         2.3 virginica
## 4           7.7         2.8          6.7         2.0 virginica
## 5           7.7         2.6          6.9         2.3 virginica
## 6           7.6         3.0          6.6         2.1 virginica
## 7           7.4         2.8          6.1         1.9 virginica
## 8           7.3         2.9          6.3         1.8 virginica
## 9           7.2         3.6          6.1         2.5 virginica
## 10          7.2         3.2          6.0         1.8 virginica
## ..          ...         ...          ...         ...       ...
```

数据拓展

语法：

```
mutate(data,NewVariable=expression)
```

- Example

```
mutate(mdt,ration=Sepal.Length/Sepal.Width)
```

```
## Source: local data frame [150 x 6]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species   ratio
n
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)    (dbl)
## 1           5.1         3.5          1.4         0.2  setosa 1.45714
3
## 2           4.9         3.0          1.4         0.2  setosa 1.63333
3
## 3           4.7         3.2          1.3         0.2  setosa 1.46875
0
## 4           4.6         3.1          1.5         0.2  setosa 1.48387
1
## 5           5.0         3.6          1.4         0.2  setosa 1.38888
9
## 6           5.4         3.9          1.7         0.4  setosa 1.38461
5
## 7           4.6         3.4          1.4         0.3  setosa 1.35294
1
## 8           5.0         3.4          1.5         0.2  setosa 1.47058
8
## 9           4.4         2.9          1.4         0.2  setosa 1.51724
1
## 10          4.9         3.1          1.5         0.1  setosa 1.58064
```

```
5
## ..          ...         ...          ...         ...     ...      ...
```

生成新数据

语法:

```
transmute(data,expression)
```

- Example

```
transmute(mdt,x1=Sepal.Length*Sepal.Width,x2=Petal.Length*Petal.Width)

## Source: local data frame [150 x 2]
##
##       x1    x2
##    (dbl) (dbl)
## 1  17.85  0.28
## 2  14.70  0.28
## 3  15.04  0.26
## 4  14.26  0.30
## 5  18.00  0.28
## 6  21.06  0.68
## 7  15.64  0.42
## 8  17.00  0.30
## 9  12.76  0.28
## 10 15.19  0.15
## ..   ...   ...
```

数据分组

语法:

```
group_by(data,FactorNames)
```

- Example

```
group_by(mdt,Species)

## Source: local data frame [150 x 5]
## Groups: Species [3]
##
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           (dbl)       (dbl)        (dbl)       (dbl)  (fctr)
## 1           5.1         3.5          1.4         0.2  setosa
## 2           4.9         3.0          1.4         0.2  setosa
## 3           4.7         3.2          1.3         0.2  setosa
## 4           4.6         3.1          1.5         0.2  setosa
## 5           5.0         3.6          1.4         0.2  setosa
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           5.0         3.4          1.5         0.2  setosa
```

```
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## ..          ...         ...          ...         ...     ...
```

数据汇总计算

语法:

```
summarise(data,expression)
```

- Example

```
summarise(mdt,total=sum(Sepal.Length,Petal.Length ))

## Source: local data frame [1 x 1]
##
##    total
##    (dbl)
## 1 1440.2

groupdt<-group_by(mdt,Species)
summarise(groupdt,total=sum(Sepal.Length,Petal.Length ),mean(Sepal.Leng
th),mean(Petal.Length))

## Source: local data frame [3 x 4]
##
##      Species total mean(Sepal.Length) mean(Petal.Length)
##       (fctr) (dbl)              (dbl)              (dbl)
## 1     setosa 323.4              5.006              1.462
## 2 versicolor 509.8              5.936              4.260
## 3  virginica 607.0              6.588              5.552
```

数据连接

语法:

```
left_join(x,y,by="")
inner_join(x,y,by="")
semi_join(x,y,by="")#相当于根据 y 保留 x 中的数据
anti_join(x,y,by="")#相当于根据 y 剔除 x 中的数据
```

- Example

```
mdt1<-read.table("mydata/Inkfish1.txt",header = T)
mdt2<-read.table("mydata/Inkfish2.txt",header = T)
dim(mdt1);dim(mdt2)

## [1] 2644    2

## [1] 2643    5

head(mdt1);head(mdt2)
```

```
##   Sample     GSI
## 1      1 10.4432
## 2      2  9.8331
## 3      3  9.7356
## 4      4  9.3107
## 5      5  8.9926
## 6      6  8.7707
```

```
##   Sample YEAR MONTH Location Sex
## 1      1    1     1        1   2
## 2      2    1     1        3   2
## 3      3    1     1        1   2
## 4      5    1     1        1   2
## 5      6    1     1        1   2
## 6      7    1     1        1   2
```

```
tail(mdt1);tail(mdt2)
```

```
##      Sample    GSI
## 2639   2639 0.0257
## 2640   2640 0.0200
## 2641   2641 0.0191
## 2642   2642 0.0165
## 2643   2643 0.0132
## 2644   2644 0.0070
```

```
##      Sample YEAR MONTH Location Sex
## 2638   2639    4    12        1   2
## 2639   2640    4    10        1   1
## 2640   2641    4    10        1   2
## 2641   2642    4    12        1   1
## 2642   2643    4    10        1   1
## 2643   2644    4    12        1   1
```

```
mdt_l<-left_join(mdt1,mdt2,by="Sample")
dim(mdt_l)
```

```
## [1] 2644    6
```

```
head(mdt_l)
```

```
##   Sample     GSI YEAR MONTH Location Sex
## 1      1 10.4432    1     1        1   2
## 2      2  9.8331    1     1        3   2
## 3      3  9.7356    1     1        1   2
## 4      4  9.3107   NA    NA       NA  NA
## 5      5  8.9926    1     1        1   2
## 6      6  8.7707    1     1        1   2
```

```
mdt_i<-inner_join(mdt1,mdt2,by="Sample")
dim(mdt_i)
```

```
## [1] 2643    6
```

```
head(mdt_i)

##   Sample     GSI YEAR MONTH Location Sex
## 1      1 10.4432    1     1        1   2
## 2      2  9.8331    1     1        3   2
## 3      3  9.7356    1     1        1   2
## 4      5  8.9926    1     1        1   2
## 5      6  8.7707    1     1        1   2
## 6      7  8.2576    1     1        1   2

mdt_s<-semi_join(mdt1,mdt2,by="Sample")
dim(mdt_s)

## [1] 2643    2

head(mdt_s)

##   Sample     GSI
## 1      1 10.4432
## 2      2  9.8331
## 3      3  9.7356
## 4      5  8.9926
## 5      6  8.7707
## 6      7  8.2576

mdt_a<-anti_join(mdt1,mdt2,by="Sample")
dim(mdt_a)

## [1] 1 2

head(mdt_a)

##   Sample    GSI
## 1      4 9.3107
```

管道函数

%>%

- Example

```
data(iris)
iris %>% group_by(Species) %>% summarise(sum(Sepal.Length),mean(Sepal.L
ength))

## Source: local data frame [3 x 3]
##
##      Species sum(Sepal.Length) mean(Sepal.Length)
##       (fctr)             (dbl)              (dbl)
## 1     setosa             250.3              5.006
## 2 versicolor             296.8              5.936
## 3  virginica             329.4              6.588
```

R 语言绘图

plot()函数

plot()函数是 R 中创建图形的最基本的函数，是泛函数。使用 plot()时，其绘图效果依赖于对象所属的类别。基本语法如下：

```
plot(x, y = NULL, type = ,  xlim = , ylim = ,log = "", main = , sub = ,
 xlab = , ylab = ,
      axes = TRUE, frame.plot = , asp = NA, ...)
```

- x,y:Vector of coordinates
- type:绘图类型

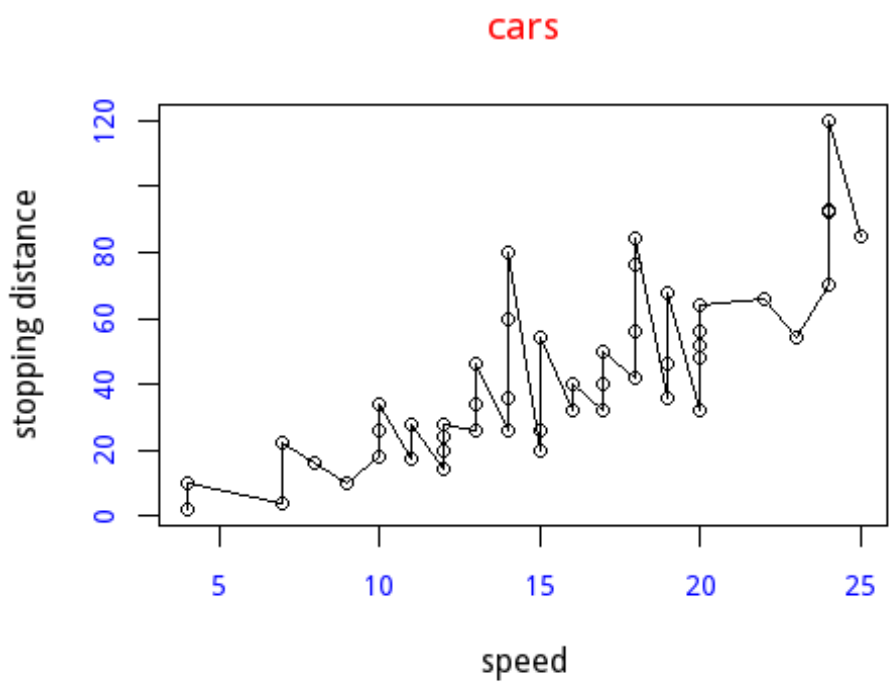
参数	含义
p	Points, default
l	lines
b	Points with line connection
c	Line connections without points
o	Both overplotted
h	Histogram like vertical lines
s	Stair steps
S	Stair steps, another style
n	No plotting

参数	功能
pch	绘制点图时所使用的符号类型
cex	绘图符号相对于默认值大小的倍数
col	绘图颜色
main	图形上方图形标题
sub	图形下方标题
col.main	标题颜色
cex.main	标题字体大小
xlab	x 轴名称
ylab	y 轴名称
xlim	x 轴坐标范围
ylim	y 轴坐标范围
asp	坐标轴 y/x 比例
axes=T/F	绘制/不绘制坐标轴

plot.frame=T/F	是否画框
col.axis	坐标刻度文字颜色
col.lab	坐标轴标签颜色
col.main	标题颜色
cex.axis	坐标轴刻度文字大小

- Example1

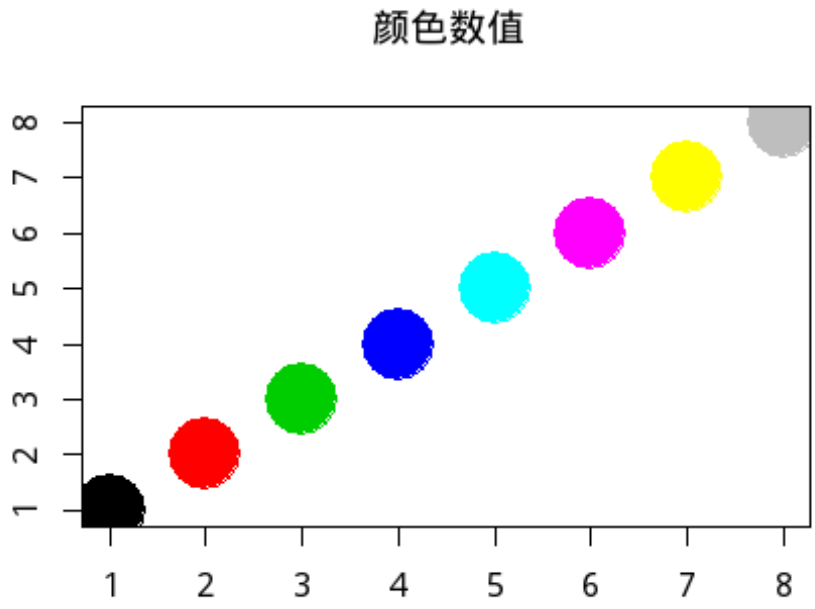
```
data(cars)
de<-par(col.axis="blue",col.main="red",cex.axis=0.9,tck=-0.05)
plot(cars,type="o",main="cars",xlab="speed",ylab="stopping distance")
```




```
par(de)
```

- Example2

```
x<-1:8
plot(x,col=x,pch=19,cex=5,main="颜色数值",xlab=" ",ylab=" ")
```



abline()函数

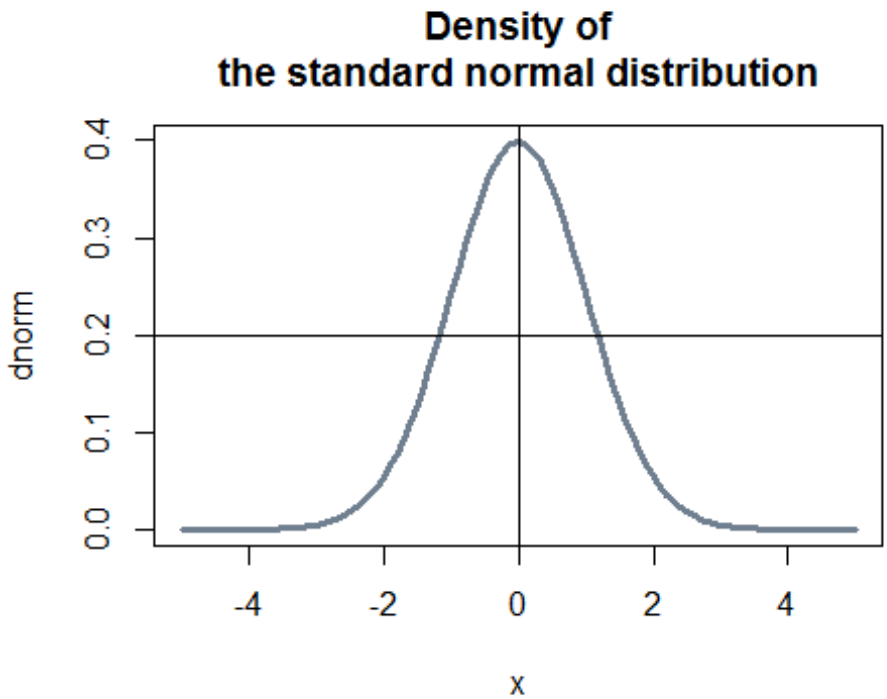
为图形添加线条语法如下：

```
abline(a = NULL, b = NULL, h = NULL, v = NULL, reg = NULL,
       coef = NULL, untf = FALSE, ...)
```

参数	功能
a, b	the intercept and slope, single values.
h	the y-value(s) for horizontal line(s).
v	the x-value(s) for vertical line(s).

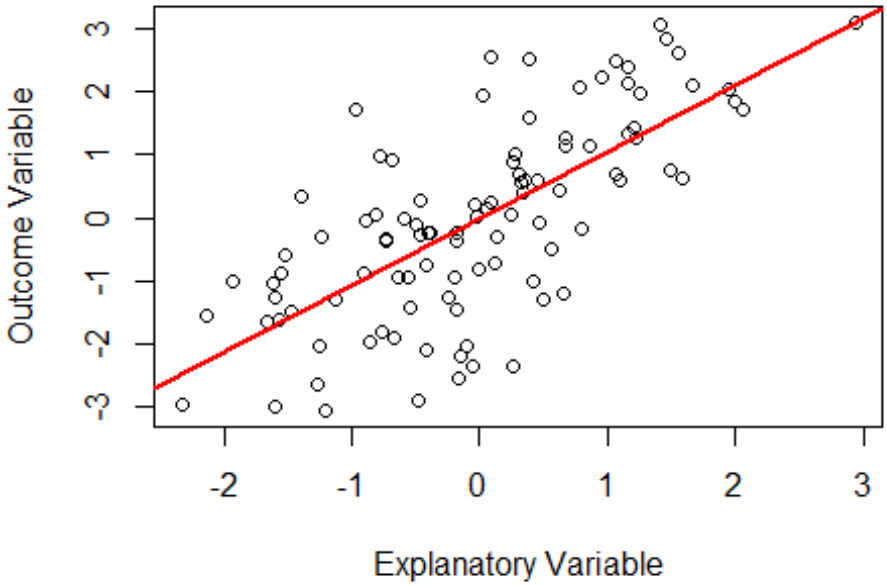
- Example1

```
plot(dnorm, from = -5, to = 5, col = "slategray", lwd = 3, main = "Dens
ity of
the standard normal distribution")
abline(h=0.2)
abline(v=0)
```



- Example2

```
n<-100
x<-rnorm(n)
y<-rnorm(n,x)
plot(x,y,
  xlab="Explanatory Variable",
  ylab="Outcome Variable")
abline(lm(y~x),col="red",lwd=2)
```



legend()函数

在图形中包含多条线条时，可以用图例加以区分

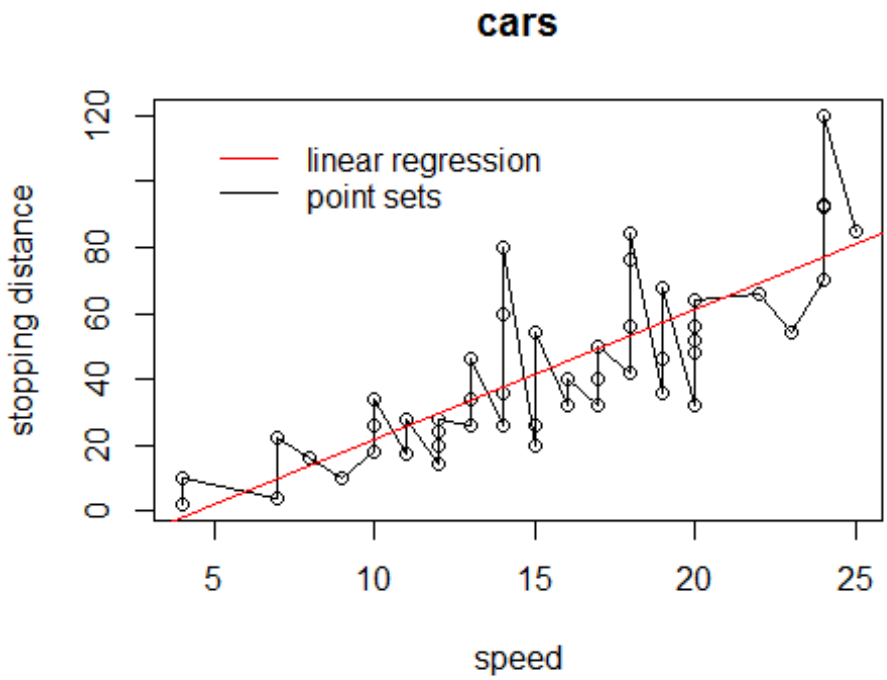
```
legend(location,title,...)
```

参数	功能
location	参数制定图例的位置，可以直接给定图例坐标(x,y)，或者用 locator(1)指定位置,或者使用关键字“bottom”,"bottomleft","left","top","topright","right","center",如果使用关键字，可以同时使用参数 inset=指定图例向图形内侧移动的大小(以绘图区域大小的分数表示)。
title	图例字符串
x.intersp	图例和文字之间的距离
y.intersp	文字之间的行距
pt.cex	图例字体大小
lty,lwd	图例线条形状和宽度
bty	图例是否家边框

- Example

```
data(iris)
plot(cars,type="o",main="cars",xlab="speed",ylab="stopping distance")
```

```
abline(lm(cars$dist~cars$speed),lwd=1.5,lty=1,col="red")
legend("topleft",inset=0.05,c("linear regression","point sets"),lty=c(1,
1),col=c("red","black"),bty="n",pt.cex=0.8,y.intersp =0.9)
```



条形图

条形图用垂直或水平的条形来展现类别变量的频数 语法：

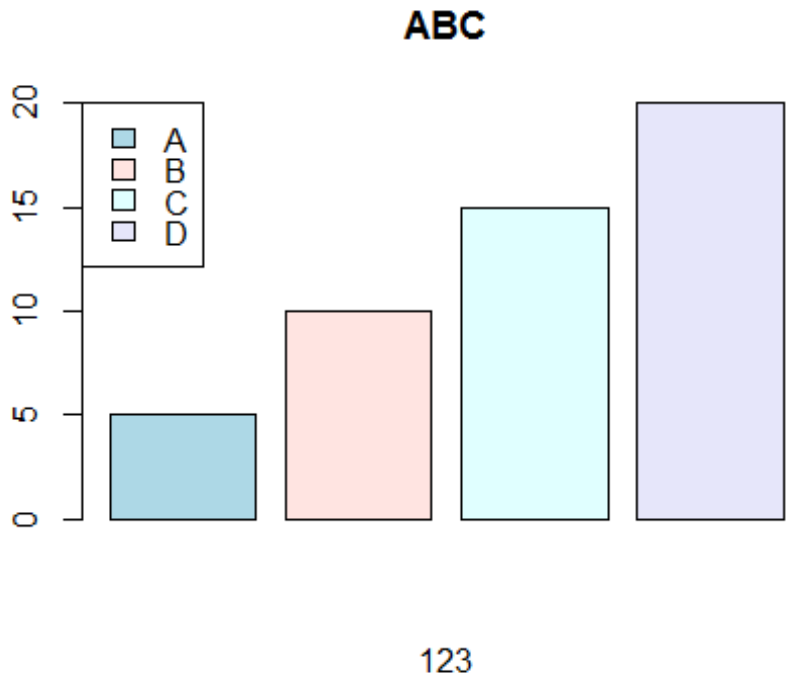
```
barplot(height,horiz=,legend.text=, col=,main=,xlab=,ylab=,...)
```

参数	功能
height	向量或者矩阵
horiz	条形图是垂直还是水平
legend.text	图例标签
col	图形颜色
main	标题
xlab	x 坐标名称
ylab	y 坐标名称

- Example1

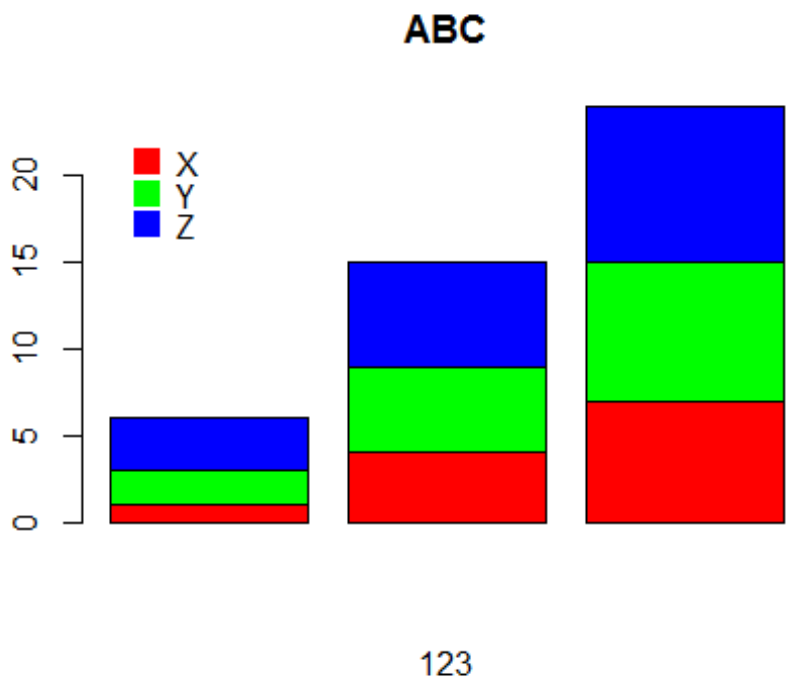
```
a<-seq(5,20,by=5)
barplot(a,col=c("lightblue", "mistyrose", "lightcyan",
```

```
                  "lavender"),main="ABC",xlab="123",legend.text = c("A",
 "B","C","D"), args.legend = list(x = "topleft",y.intersp=0.8))
```



- Example2

```
b<-matrix(1:9,3)
barplot(b,main="ABC",xlab="123",col=rainbow(3))
legend("topleft",inset=0.05,c("X","Y","Z"),pch=c(15,15,15),col=rainbow
(3),bty="n",pt.cex=1.8,y.intersp =0.8)
```



饼图

饼图是应用非常广泛的统计图形之一。

语法：

```
pie(x, labels = names(x), edges = 200, radius = 0.8,clockwise = FALSE,
init.angle = NULL,
     col = NULL, main = NULL)
```

参数	功能
参数 x	为一个数值向 量
labels	为标签
clockwise	画图的方向
init.angle	第一块饼图的起始位置，可写 0 或 90
col	颜色
main	标题

- Example

```
pie.sales <- c(0.12, 0.3, 0.26, 0.16, 0.04, 0.12)
pie.col<- c("purple", "violetred1", "green3", "cornsilk", "cyan", "whit
e")
```

```r
pi.names<-c("A","B","C","D","E","F")
pie(pie.sales, col = pie.col,labels=pi.names,main="饼图示例")
```