



语言入门

张金龙

jinlongzhang01@gmail.com

中国科学院植物研究所

2009.12.9

报告内容

- 一 R简介
 - 二 函数与对象
 - 三 编写脚本
 - 四 R绘图
 - 五 编写函数
 - 六 数据保存
-

一 R 简介

R语言的由来

- R语言是从S语言演变而来的。
 - S语言是二十世纪70年代诞生于贝尔实验室，由Rick Becker, John Chambers, Allan Wilks开发。
 - 基于S语言开发的商业软件Splus，可以方便的编写函数、建立模型，具有良好的扩展性，取得了巨大成功。
 - 1995年由新西兰Auckland大学统计系的Robert Gentleman和Ross Ihaka，编写了一种能执行S语言的软件，并将该软件的源代码全部公开，这就是R软件，其命令统称为R语言。
-

R软件简介

R是开源软件，代码全部公开，对所有人免费。

R可在多种操作系统下运行，如Windows、MacOS、多种Linux和UNIX等。

R需要输入命令，可以编写函数和脚本进行批处理运算，语法简单灵活。

目前在R网站上约有2100个程序包，涵盖了基础统计学、社会学、经济学、生态学、地理学、医学统计学、生物信息学等诸多方面。

下载和安装R

The **C**omprehensive **R** Archive **N**etwork

简称**CRAN**，提供下载安装程序和相应软件包。

R主页 <http://www.r-project.org/>

Windows版本下载地址之一：

[http://ftp.ctex.org/mirrors/CRAN/bin/windows/base/
R-2.9.2-win32.exe](http://ftp.ctex.org/mirrors/CRAN/bin/windows/base/R-2.9.2-win32.exe)

下载完成后，双击[R-2.9.2-win32.exe](http://ftp.ctex.org/mirrors/CRAN/bin/windows/base/R-2.9.2-win32.exe) 开始安装。

一直点击下一步，各选项默认。



The R Project for Statistical Computing

About R
[What is R?](#)
[Contributors](#)
[Screenshots](#)
[What's new?](#)

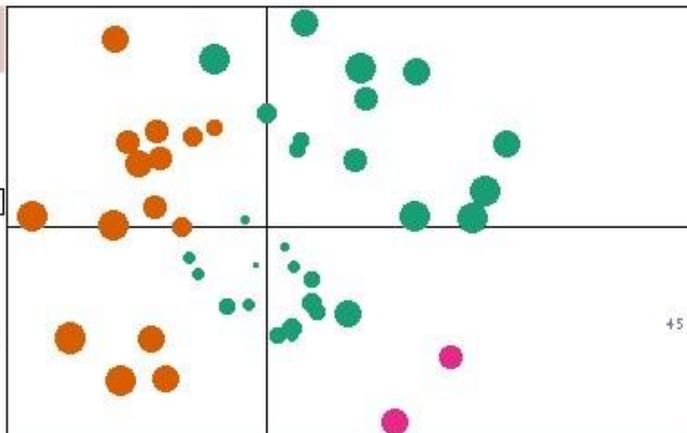
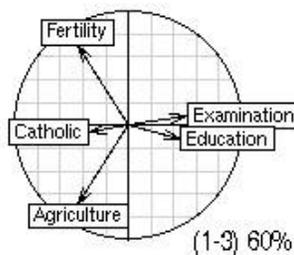
Download, Packages
[CRAN](#)

R Project
Foundation
[Members & Donors](#)
[Mailing Lists](#)
[Bug Tracking](#)
[Developer Page](#)
[Conferences](#)
[Search](#)

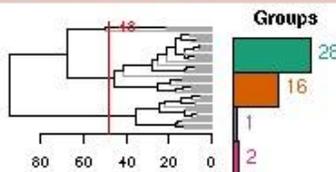
Documentation
[Manuals](#)
[FAQs](#)
[The R Journal](#)
[Wiki](#)
[Books](#)
[Certification](#)
[Other](#)

Misc
[Bioconductor](#)
[Related Projects](#)

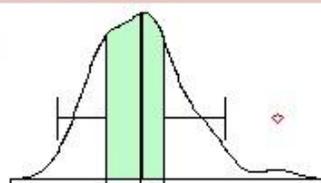
PCA 5 vars
`princomp(x = data, cor = cor)`



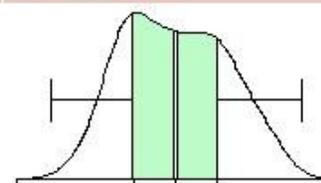
Clustering 4 groups



Factor 1 [41%]



Factor 3 [19%]



Getting Started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred [CRAN mirror](#).
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

图1 R软件首页 <http://www.r-project.org/>

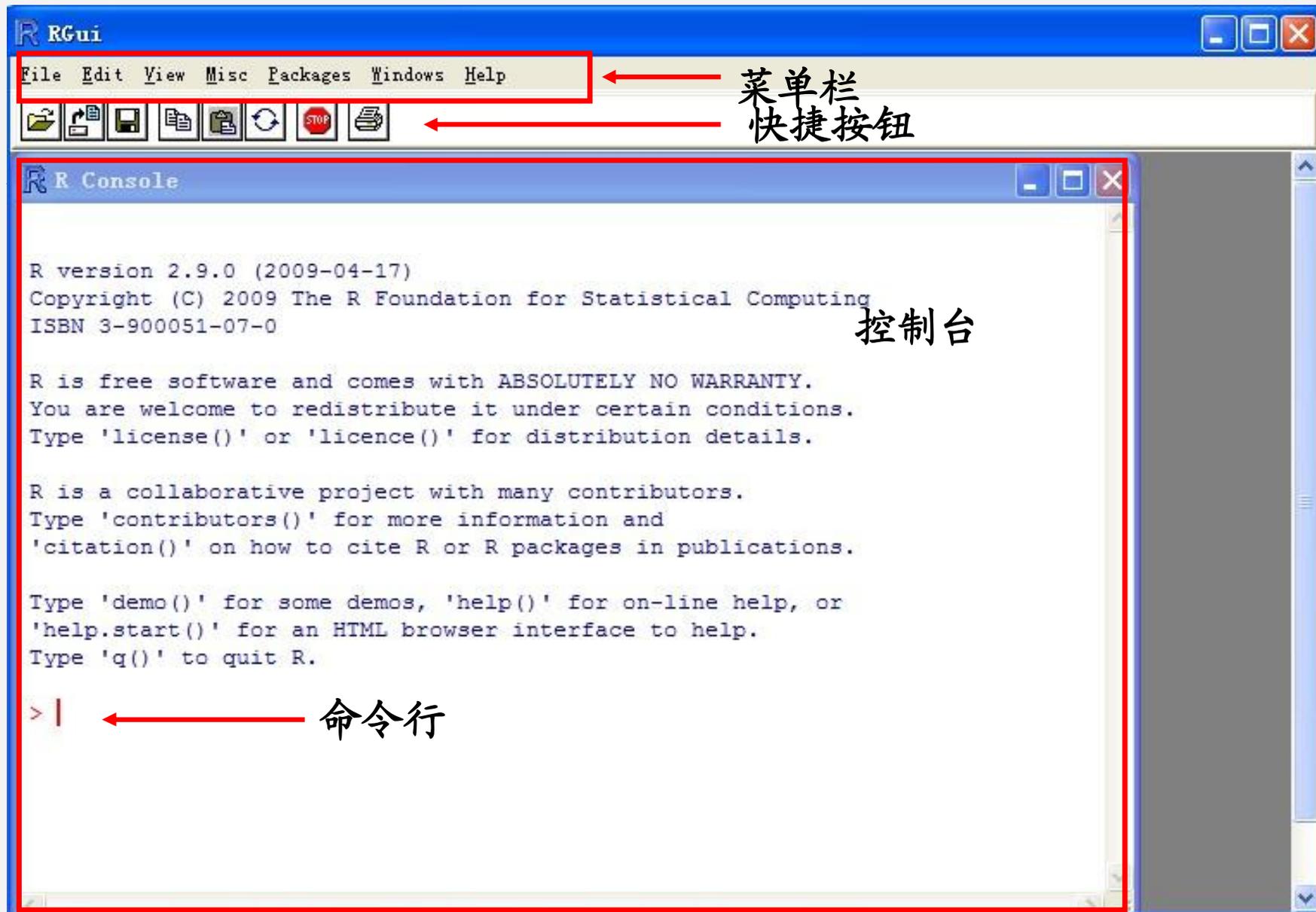


图2 R登陆界面(Windows版)

路径: 开始>所有程序>R 2.9.2



图3 R Gui 的菜单介绍

练习一：下载和安装R

下载并安装R软件

了解R的菜单

R程序包

为什么要安装程序包？

特定的分析功能，需要用相应的程序包实现。

例如：系统发育分析，往往要用到ape程序包，群落生态学vegan包等等。

程序包是什么？

R程序包是多个函数的集合，具有详细的说明和示例。

Window下的R程序包是已经编译好的zip包。

每个程序包包含R函数、数据、帮助文件、描述文件等。

常用R程序包

base-	R 基础功能包
stats-	R统计学包
nlme-	线性及非线性混合效应模型
Graphics-	绘图
lattice-	栅格图
ape-	系统发育与进化分析
apTreeshape-	进化树分析
seqinr-	DNA序列分析
ade4-	利用欧几里得方法进行生态学数据分析

常用R程序包

- cluster- 聚类分析
 - ecodist- 生态学数据相异性分析
 - mefa- 生态学和生物地理学多元数据处理
 - mgcv- 广义加性模型相关
 - mvpart- 多变量分解
 - nlme- 线性及非线性混合效应模型
 - ouch- 系统发育比较
 - BiodiversityR - 基于Rcmdr的生物多样性数据分析
 - vegan- 植物与植物群落的排序，生物多样性计算
-

常用R程序包

- maptools- 空间对象的读取和处理
 - sp- 空间数据处理
 - spatstat- 空间点格局分析，模型拟合与检验
 - splancs- 空间与时空点格局分析
 - picante- 群落系统发育多样性分析
-

CRAN Task Views

Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
Cluster	Cluster Analysis & Finite Mixture Models
Distributions	Probability Distributions
Econometrics	Computational Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
Finance	Empirical Finance
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visual
gR	gRaphical Models in R
HighPerformanceComputing	High Performance and Parallel Computing
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Psychometrics	Psychometric Models and Methods
Robust	Robust Statistical Methods
SocialSciences	Statistics for the Social Sciences
Spatial	Analysis of Spatial Data
Survival	Survival Analysis

图4 CRAN Task Views: 对程序包的分类介绍

vegan: Community Ecology Package

Ordination methods, diversity analysis and other functions for community and vegetation ecologists.

Version: 1.15-3
Suggests: [MASS](#), [mgcv](#), [lattice](#), [cluster](#), [scatterplot3d](#), [rgl](#), [ellipse](#), [tcltk](#)
Published: 2009-06-17
Author: Jari Oksanen, Roeland Kindt, Pierre Legendre, Bob O'Hara, Gavin L. Simpson, Peter Solymos, Stevens, Helene Wagner
Maintainer: Jari Oksanen <jari.oksanen at oulu.fi>
License: [GPL-2](#)
URL: <http://cran.r-project.org/>, <http://vegan.r-forge.r-project.org/>
In views: [Environmetrics](#), [Multivariate](#), [Psychometrics](#), [Spatial](#)
CRAN checks: [vegan results](#)

Downloads:

Package source: [vegan 1.15-3.tar.gz](#)
MacOS X binary: [vegan 1.15-3.tgz](#)
Windows binary: [vegan 1.15-3.zip](#)
Reference manual: [vegan.pdf](#)
Vignettes: [Design decisions and implementation](#)
[Diversity analysis in vegan](#)
[Introduction to ordination in vegan](#)
News/ChangeLog: [NEWS](#) [ChangeLog](#)

图5 vegan包页面

R程序包

在CRAN 提供了每个包的源代码和编译好的MacOS、
Window下的程序包

以vegan包为例，CRAN提供了：

Package source: [vegan_1.15-3.tar.gz](#)

MacOS X binary: [vegan_1.15-3.tgz](#)

Windows binary: [vegan_1.15-3.zip](#)

Reference manual: [vegan.pdf](#) 等

Window平台下程序包为zip文件，安装时**不要解压缩**。

R程序包安装

1 连网时，用函数**install.packages()**，
选择镜像后，程序将自动下载并安装程序包。

例如： 打开**RGui**，在控制台中输入

```
install.packages("ape")
```

2 安装本地zip包

路径：**Packages>install packages from
local files**

选择光盘或者本地磁盘上存储zip包的文件夹。

程序包使用

在控制台中输入如下命令:

```
library(vegan)
```

```
library(ade4)
```

调用程序包内的函数与R内置的函数调用方法一样

```
library(vegan)
```

```
This is vegan 1.15-3
```

```
Warning message:
```

```
package 'vegan' was built under R
```

```
version 2.9.1
```

练习二 安装并导入程序包

安装程序包

程序包>从本地**zip**文件安装程序包

调用程序包

```
library(vegan)
```

```
library(ape)
```

查看帮助文件

如何知道ape程序包内部都有哪些函数？

最常用的方法：

1 菜单 帮助>Html帮助

2 查看pdf帮助文档（从程序包下载页面下载）

查看帮助文件

1 `help("t.test")`

2 `?t.test`

3 `help.search("t.test")`

4 `apropos("t.test")`

5 RGui>Help>Html help

6 查看R包pdf手册

帮助文件的内容

以lm函数为例:

lm(stats) #函数名及所在包

Fitting Linear Models # 标题

Description #函数描述

Usage # 默认选项

Arguments # 参数

Details # 详情

Author(s) # 作者

References # 参考文献

Examples # 举例

练习三 查看帮助文件

打开ape软件包的帮助文件

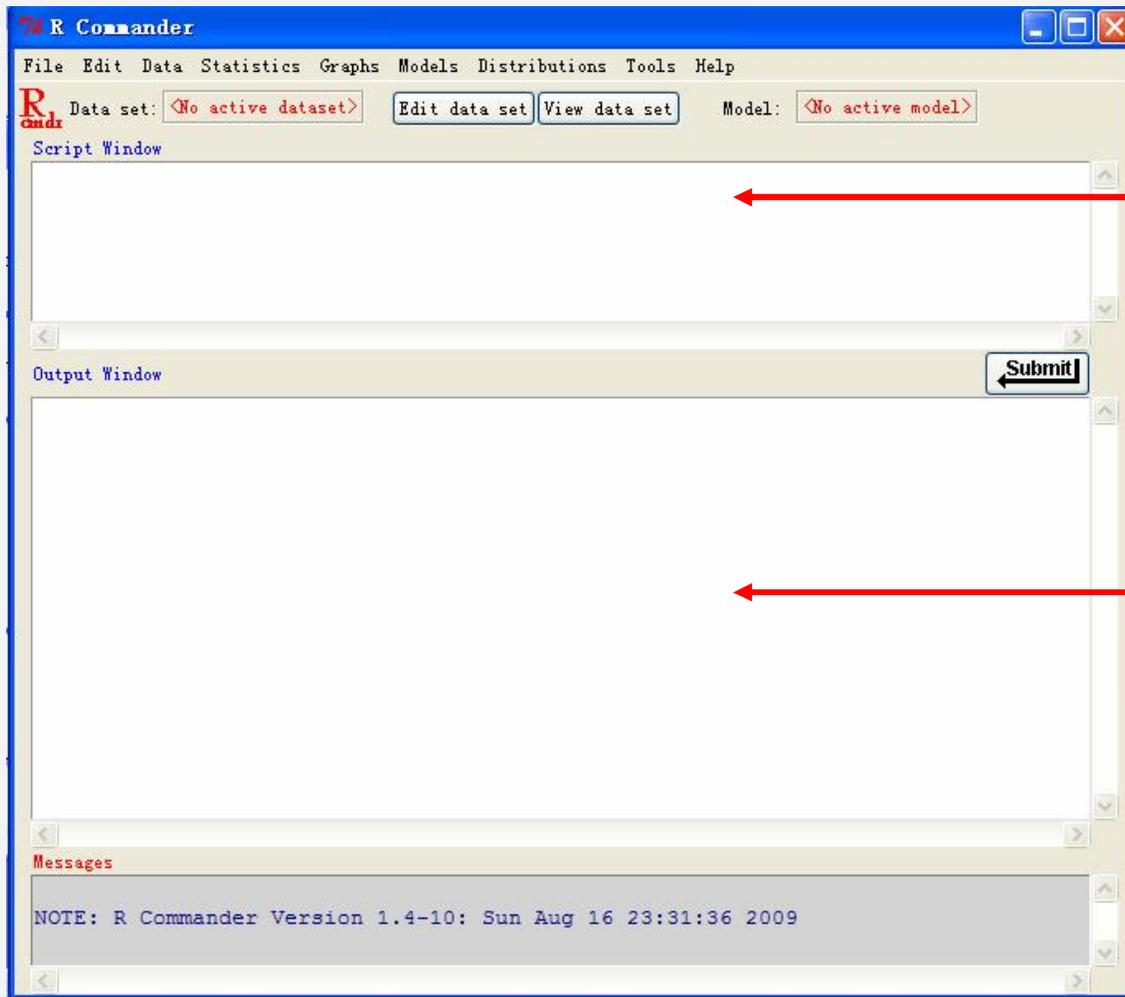
```
library(ape)
```

查找ape包中plot.phylo函数的帮助

```
输入 ?plot.phylo
```

将其中的Example文件粘贴到Console中，查看运行的结果。

图形界面之一：R Commander



界面操作代码

结果输出

图6 R commander 界面 **library(Rcmdr)**

为什么要学习编程？

界面操作直观易学，但也存在一些不足：

操作的过程难以保存，数据处理不够灵活，在进行步骤繁多的数据处理工作时十分费时费力；在建立模型或自己编写函数时也会遇到困难。而这些困难可以通过编程解决。

学习R编程，首先要了解R的函数、对象及其操作。

二 R函数与对象

数据表的行与列

表1 数据表、数据框与向量

	物种数	科数	属数	海拔	坡度	类型
样方1	40	15	22	600	25	山顶
样方2	51	12	26	350	30	山坡
样方3	46	11	20	390	45	山坡
样方4	38	12	24	260	20	低地
样方5	49	10	25	220	33	低地

列名
Column
names

每行
作为一个
Entry

字符串、因素

行名 Row names

每列可看做带名

字符串

称的向量

R的函数

R是一种解释性语言，不需要先编译成 **.exe** 文件，输入后可直接运行。

函数形式

function(对象, 选项=)

平均值 **mean**()

线性回归 **lm**(y~x, data=test)

R处理的所有数据、变量、函数和结果都以**对象**的形式保存。

R的函数

每一个函数执行特定的功能，后面紧跟括号，例如：

平均值 `mean()`

求和 `sum()`

绘图 `plot()`

排序 `sort()`

除了基本的运算之外，R的函数又分为高级和低级函数，高级函数内部嵌套了复杂的低级函数，例如 `plot()` 是高级绘图函数，函数本身会根据数据的类型，经过程序内部的函数判别之后，绘制相应类型的图形，并有大量的参数可选择。

部分函数

计算

`log(x)`
`log10(x)`
`exp(x)`
`sin(x)`
`cos(x)`
`tan(x)`
`asin(x)`
`acos(x)`
`min(x)`
`max(x)`
`range(x)`
`length(x)`

统计检验

`mean(x)`
`sd(x)`
`var(x)`
`median(x)`
`quantile(x,p)`
`cor(x,y)`
`t.test()`
`lm(y ~ x)`
`wilcox.test()`
`kruskal.test()`

统计检验

`lm(y ~ f+x)`
`lm(y ~ x1+x2+x3)`
`bartlett.test`
`binom.test`
`fisher.test`
`chisq.test`
`glm(y ~ x1+x2+x3,
 binomial)`
`friedman.test`
`...`

R函数调用及其选项

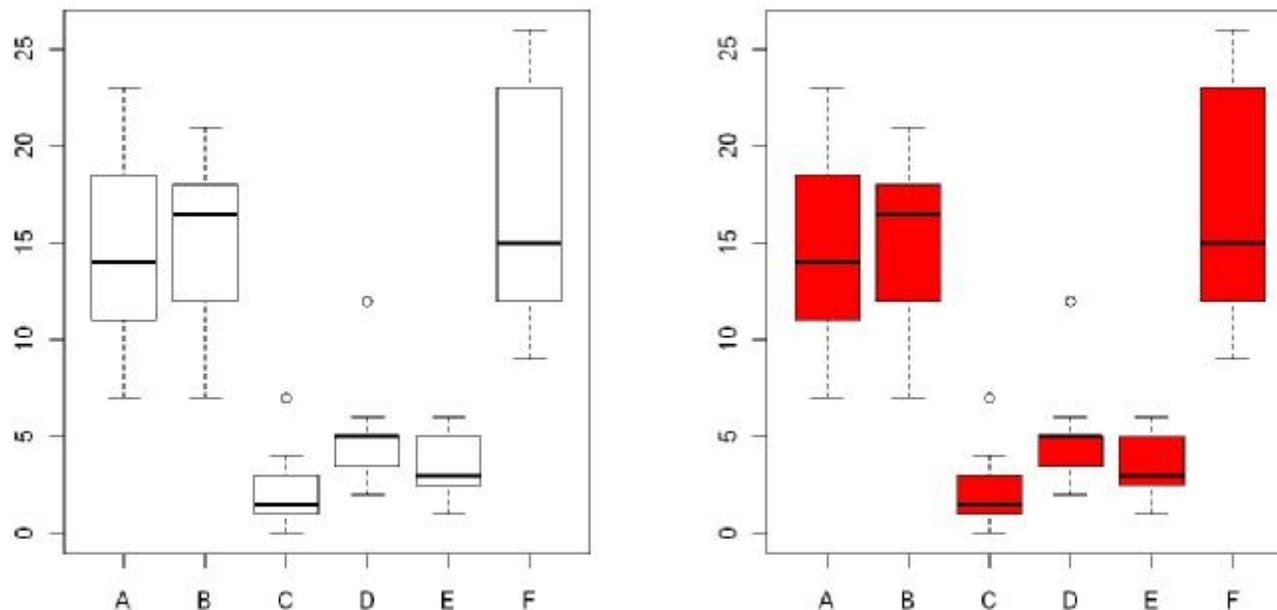


图7 箱线图修饰前后(左: 默认值, 右: 修改属性后)

```
boxplot(count ~ spray, data = InsectSprays)
```

```
boxplot(count ~ spray, data = InsectSprays, col = "red")
```

R函数调用及其选项

函数的调用方法，**函数名+()** 如 **plot()**，**lm()**，并将对象放入括号中，“=”表示设定参数。例如：

```
boxplot(day~type, data=bac, col="red",  
xlab="Virus", ylab="days")
```

day~type，以**type**为横轴，**day**为纵轴绘制箱线图。

data=bac 数据来源**bac**

col="red" 箱线图为红色

xlab="Virus" 横轴名称为**Virus**

ylab="days" 纵轴名称为**days**

练习四：查询函数帮助

查看**boxplot**的帮助文件

?boxplot

查看最后的**examples**

将帮助文件中的内容粘贴到控制台中，运行并观看运行结果。

```
boxplot(count ~ spray, data = InsectSprays,  
         col = "lightgray")
```

选项更改

```
boxplot(count ~ spray, data = InsectSprays,  
         col = "red", xlab="spray", ylab="counts")
```

赋值与注释

在控制台 中键入如下命令

```
2+2
```

```
a<-2
```



赋值符号

<-也可用=, ->代替

```
b<-2
```

```
c<-a+b
```

```
c
```

```
#注释
```

为对象起名

R是区分大小写的，**A**与**a**是不同的。

对象名不能用数字开头，但是数字可以放在中间或结尾。推荐用“.”作为间隔，例如 `anova.result1`。

举例：一个向量**a**，包含了四个元素，

`a<-c(10,15,21,18)`该向量为数值型，长度为4。

保留名称(命名对象时不要与保留名称冲突)：

`NA, NaN, pi, LETTERS, letters, month.abb, month.name`

对象的类型

数值型	Numeric	如 100, 0, -4.335
字符型	Character	如 "China"
逻辑型	Logical	如 TRUE, FALSE
因子型	Factor	表示不同类别
复数型	Complex	如: $2 + 3i$

对象的类别

向量 (vector) 一系列元素的组合。

因子 (factor) 因子是一个分类变量，如

"a", "a", "a", "a", "b", "b", "b", "c", "c"

数组 (array) 数组是k维的数据表。

矩阵 (matrix) 矩阵是数组的一个特例，维数 $k = 2$ 。

数据框 (dataframe) 是由一个或几个向量和（或）因子构成，它们必须是等长的，但可以是不同的数据类型。

列表 (list) 列表可以包含任何类型的对象。

(据Paradis, 2005)

运算符

数学运算 运算后给出数值结果

+, -, *, /, ^

比较运算 运算后给出判别结果 (**TRUE FALSE**)

>, <, <=, >=, ==, !=

逻辑运算 与、或、非

!, &, &&, |, ||

外部数据读取

最为常用的数据读取方式是用 `read.table()` 函数或 `read.csv()` 函数读取外部 `txt` 或 `csv` 格式的文件。

txt文件，制表符间隔

csv文件，逗号间隔

一些R程序包（如 `foreign`）也提供了直接读取 `Excel`, `SAS`, `dbf`, `Matlab`, `spss`, `systat`, `Minitab` 文件的函数。

read.table()的使用

例: test.data<-

```
read.table("D:/R/test2.txt",header=T)
```

header=T表示将数据的第一行作为标题。

```
read.table(file=file.choose(),header=T)
```

可以弹出对话框，选择文件。

实例：从数据输入到t检验

现有6名患者的身高和体重，检验体重除以身高的平方是否等于22.5。

表2 六名患者的身高和体重

编号	1	2	3	4	5	6
身高 m	1.75	1.80	1.65	1.90	1.74	1.91
体重kg	60	72	57	90	95	72

第一种方式：从控制台输入数据

数据量较少时可以从控制台直接输入：

```
height<-c(1.75, 1.80, 1.65, 1.90, 1.74,  
          1.91)
```

```
weight<-c(60, 72, 57, 90, 95, 72)
```

```
sq.height<-height^2
```

```
ratio<-weight/sq.height
```

```
t.test(ratio, mu=22.5)
```

第二种方式 从外部读取数据

数据量较大时用 `read.table` 函数从外部 `txt` 文件
读取

第1步 将 `Excel` 中的数据另存为 `.txt` 格式（制表符间隔）或 `.csv` 格式。

第2步 用 `read.table()` 或 `read.csv()` 函数将数据读入 `R` 工作空间，并赋值给一个对象。

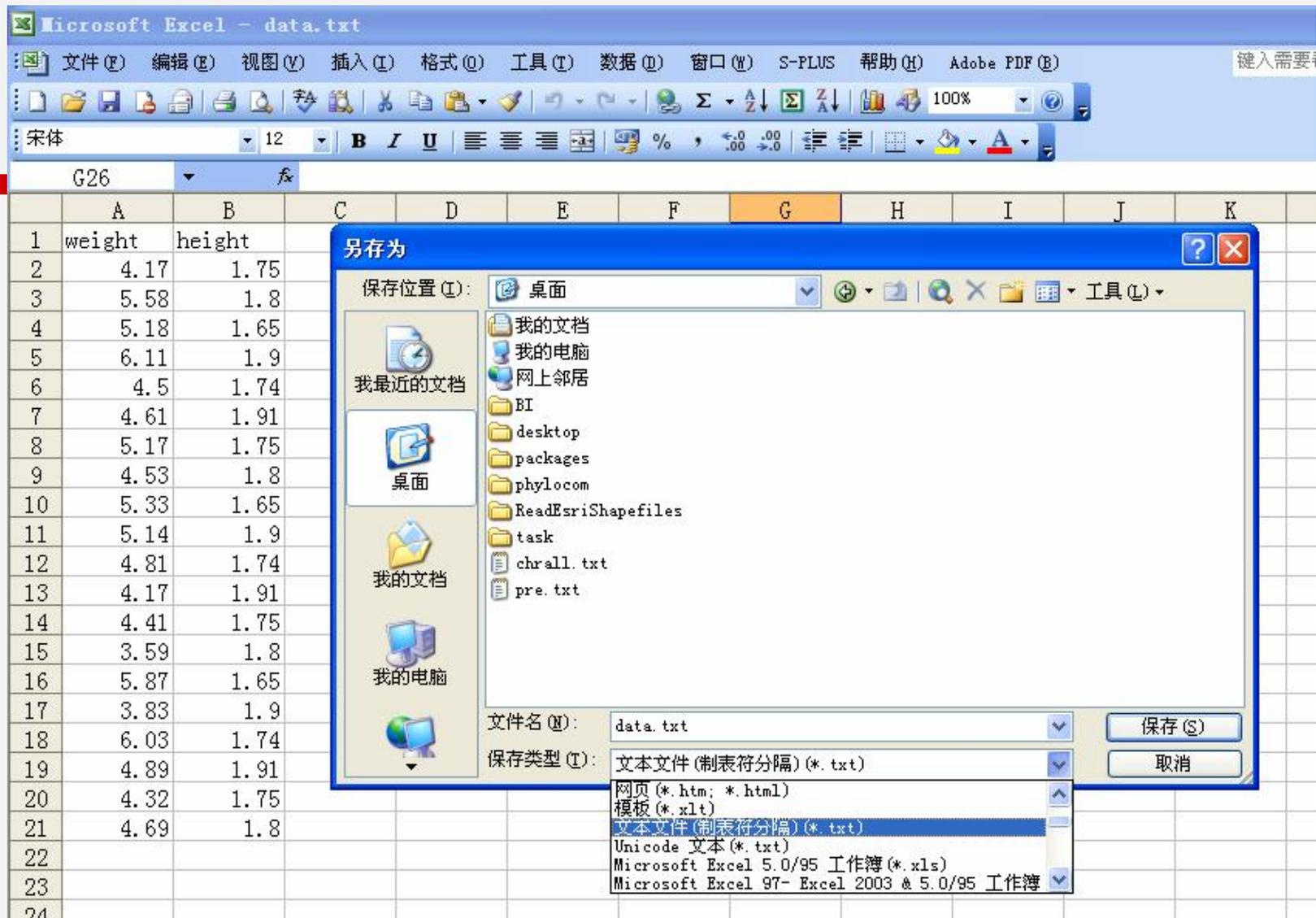


图8 在Excel中将数据存为txt文件

实例：t检验(续)

一般从txt文档读取数据。每一行作为一个观测值。每一行的变量用制表符，空格或逗号间隔开。

```
read.table("位置", header=T)
```

```
read.csv("位置", header=T)
```

```
#从外部读取数据
```

```
data1<-read.table("d:/t.test.data.txt", header=T)
```

```
bmi<- data1$weight/data1$height2
```

```
t.test(bmi, mu=22.5) #t检验
```

练习五：数据读取和t检验

将表2中的数据录入Excel中，另存为t.test.txt文件。

用read.table函数读取该文件。

```
t.test.data<-read.table("X:/t.test.txt",  
header=T)
```

对变量t.test.data中的

```
attach(t.test.data)
```

```
ratio<-weight/height^2
```

```
t.test(ratio)
```

实例：从数据输入到单因素方差分析

将三种不同菌型的伤寒病毒**a, b, c**分别接种于**10, 9, 和11**只小白鼠上，观察其存活天数，问三种菌型下小白鼠的平均存活天数是否有显著差异。

a菌株： 2, 4, 3, 2, 4, 7, 7, 2, 5, 4

b菌株： 5, 6, 8, 5, 10, 7, 12, 6, 6

c菌株： 7, 11, 6, 6, 7, 9, 5, 10, 6, 3, 10

准备数据表

	A	B	C	D	E	F
1	day	type				
2	2	a				
3	4	a				
4	3	a				
5	2	a				
6	4	a				
7	7	a				
8	7	a				
9	2	a				
10	5	a				
11	4	a				
12	5	b				
13	6	b				
14	8	b				
15	5	b				
16	10	b				
17	7	b				
18	12	b				

图9 数据表的准备
day和type 各为一列

实例：方差分析(续)

#数据读取，将**test1.txt**中的内容保存到**bac**中，
header=T表示保留标题行。

```
bac<-read.table("d:/anova.data.txt",header=T)
```

#将**ba**数据框中的**type**转换为因子(**factor**)

```
bac$type<-as.factor(bac$type)
```

```
ba.an<-aov(lm(day~type, data=bac))
```

```
summary(ba.an)
```

```
boxplot(day~type,data=bac,col="red")
```

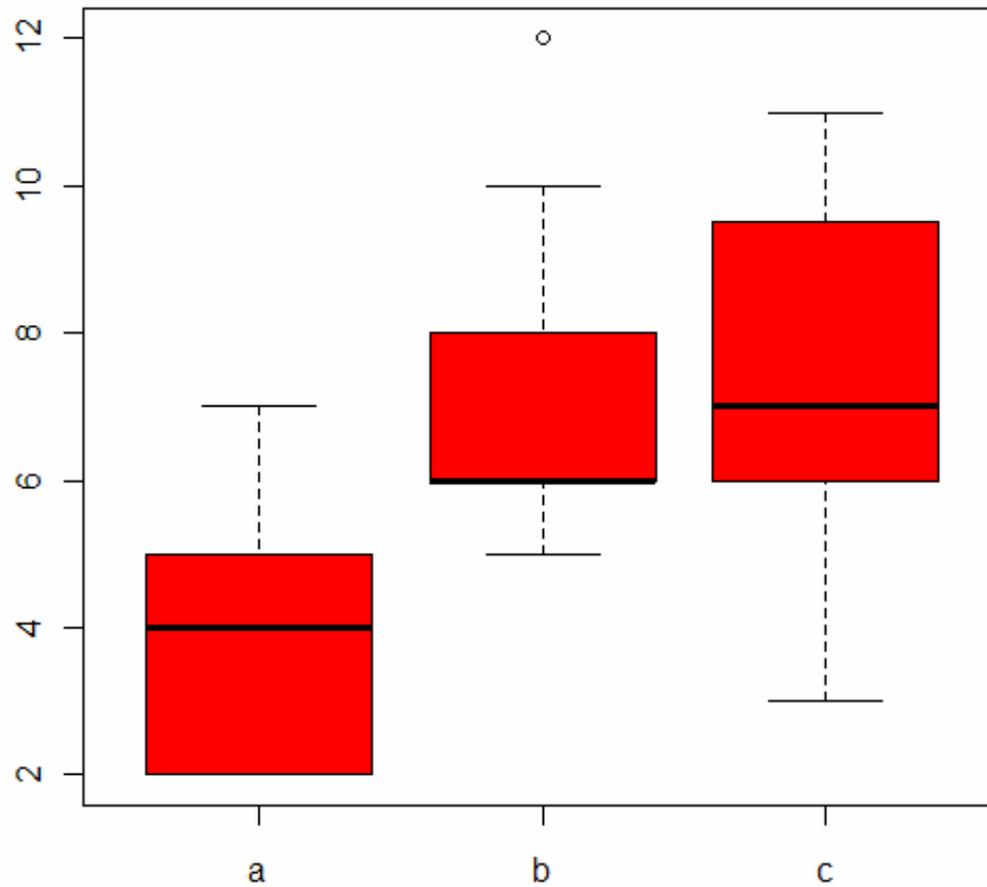


图10 三种菌型对小白鼠影响的箱线图

练习六： 方差分析和箱线图绘制

- 1 在**excel**中准备数据表
- 2 用**R**读取数据表
- 3 输入如下命令进行方差分析、绘制箱线图

```
boxplot ( day~type , data=bac , col="red" )
```

```
ba . an <- aov ( lm ( day~type , data=bac ) )
```

```
summary ( ba . an )
```

向量、矩阵和数据框的生成与条件筛选

有时需要对读入的数据进行操作，将某一向量转换成矩阵，如条件筛选，此时将遇到向量、矩阵和数据框的生成、条件筛选等。

例如：提取前面群落数据表中，物种数 >30 的行，提取其中的某一系列，进行分析等。

向量的创建

四种类型的向量

字符型

```
character<-c("China", "Korea", "Japan",  
"UK", "USA", "France", "India", "Russia")
```

数值型

```
numeric<-c(1, 3, 6, 7, 3, 8, 6, 4)
```

逻辑型

```
logical<-c(T, F, T, F, T, F, F, T)
```

复数型 略

向量的创建

```
c(2,5,6,9)
```

```
rep(2,times=4)
```

```
seq(from=3, to=21, by=3 )
```

```
[1] 3 6 9 12 15 18 21
```

```
" : "
```

```
1:15
```

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

通过与向量的组合，产生更为复杂的向量。

```
rep(1:2, c(10,15))
```

向量: 随机数的生成

```
runif(10, min = 0, max= 1)
```

```
[1] 0.32227168 0.12759789 0.33849635  
    0.84843855 0.67293416 0.14646444  
[7] 0.60117150 0.39023874 0.04219423  
    0.67102520
```

```
rnorm(10, mean = 0, sd = 1)
```

```
[1] -1.58587380 -0.07775222 2.17126687  
    -1.02938226 0.46392281 0.74896049  
[7] -0.24556829 1.39034371 1.09975487  
    -1.44682767
```

定义矩阵的维度

dim()和**matrix()**

```
x <- 1:12
```

```
dim(x) <- c(3,4)
```

```
      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

```
matrix.x <- matrix(1:12,nrow=3,byrow=T)
```

t(x) #转置

为行或列添加名称:

```
row.names()
```

```
col.names()
```

数据框的创建

cbind() # 按列组合成数据框

country.data <-

cbind(character, numeric, logical)

rbind() # 按行组合成数据框

data.frame() # 生成数据框

d <-

data.frame(character, numeric, logical)

head(d) # 访问数据的前6行:

列表的创建

列表可以是不同类型甚至不同长度的向量(数值型, 逻辑型, 字符型等等)、数据框甚至是列表的组合。

list()

例如

list(character, numeric, logical, matrix.x)

对象的类型的判断

对象类型判断

mode()

class()

is.numeric() #返回值为**TRUE**或**FALSE**

is.logical()

is.character()

is.data.frame()

对象的类型的判断

对象类型转换

`as.numeric()` #转换为数值型

`as.logical()`

`as.character()`

`as.matrix()`

`as.dataframe()`

练习七：因子生成

将 100, 200, 400, 600, 800 输入R中，保存到**numeric**对象中

```
numeric <- c(100, 200, 400, 600, 800)
```

将**numeric**转换为**factor**

```
factor.numeric <- as.factor(numeric)
```

查看**factor.numeric**的内容

```
factor.numeric
```

引用向量内的元素

```
intake.pre <- c(5260, 5470, 5640, 6180,  
  6390, 6515, 6805, 7515, 7515, 8230,  
  8770)
```

```
intake.post <- c(3910, 4220, 3885, 5160,  
  5645, 4680, 5265, 5975, 6790, 6900,  
  7335)
```

```
intake.pre[5]; intake.pre[c(3,5,7)]
```

```
v <- c(3,5,7);intake.pre[v]
```

```
intake.pre[1:5]; intake.pre[-c(3,5,7)]
```

引用数据框中的元素

```
d <- data.frame(intake.pre, intake.post)
```

\$ 引用

```
d$intake.pre
```

[,] 方括号引用

```
d[,1]; d[5,]
```

访问数据框内的元素

直接调用数据框内的列向量

attach()

detach()

在函数内部，对数据进行相应调整

with()

subset()

within()

transform()

条件筛选

条件筛选是先对变量是否满足条件进行判断，满足为**TRUE**，不满足为**FALSE**。之后再用逻辑值对向量内的元素进行筛选。

```
intake.pre > 7000
```

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

```
intake.pre[intake.pre > 7000]
```

```
intake.post[intake.pre > 7000 &  
intake.pre <= 8000]
```

```
intake.pre > 7000 & intake.pre <= 8000
```

练习八：条件筛选

创建一个2到50的向量 `vector1`

`2, 4, 6, 8, ..., 48, 50`

`vector1 <- seq(from=2, to=50, by=2)`

选取 `vector1` 中的第20个元素 `vector1[20]`

选取 `vector1` 中的第10, 15, 20个元素

`vector1[c(10,15,20)]`

选取 `vector1` 中的第10到20个元素

`vector1[10:20]`

选取 `vector1` 中值大于40的元素

`vector1[vector > 40]`

排序

将向量中的元素按照一定顺序排列。

sort() 按数值大小排序

举例:

```
intake$post
```

```
sort(intake$post)
```

order() 给出从小到大的出现序号。

```
order(intake$post)
```

```
o <- order(intake$post)
```

工作空间

ls() 列出工作空间中的对象

rm() 删除工作空间中的对象

rm(list=ls()) 删除空间中所有对象

save.image() 保存工作镜像

sink() 将运行结果保存到指定文件中

getwd() 显示当前工作文件夹

setwd() 设定工作文件夹

练习九：了解工作路径

1 查看当前R工作的空间目录

```
getwd()
```

2 将R工作的路径设置为 `d:/data/`

```
setwd("d:/data")
```

三 编写脚本

编写脚本（Scripting）

脚本是什么？

脚本是一系列命令。

可以先批量的编号程序，或者对别人已经编好的程序进行修改。之后输入到控制台进行调试，以满足数据分析的需求。

语言高亮显示

在代码较多的情况下，有时需要对行数、函数、括号、函数选项等进行高亮显示，设置成不同的颜色，以减少错误。

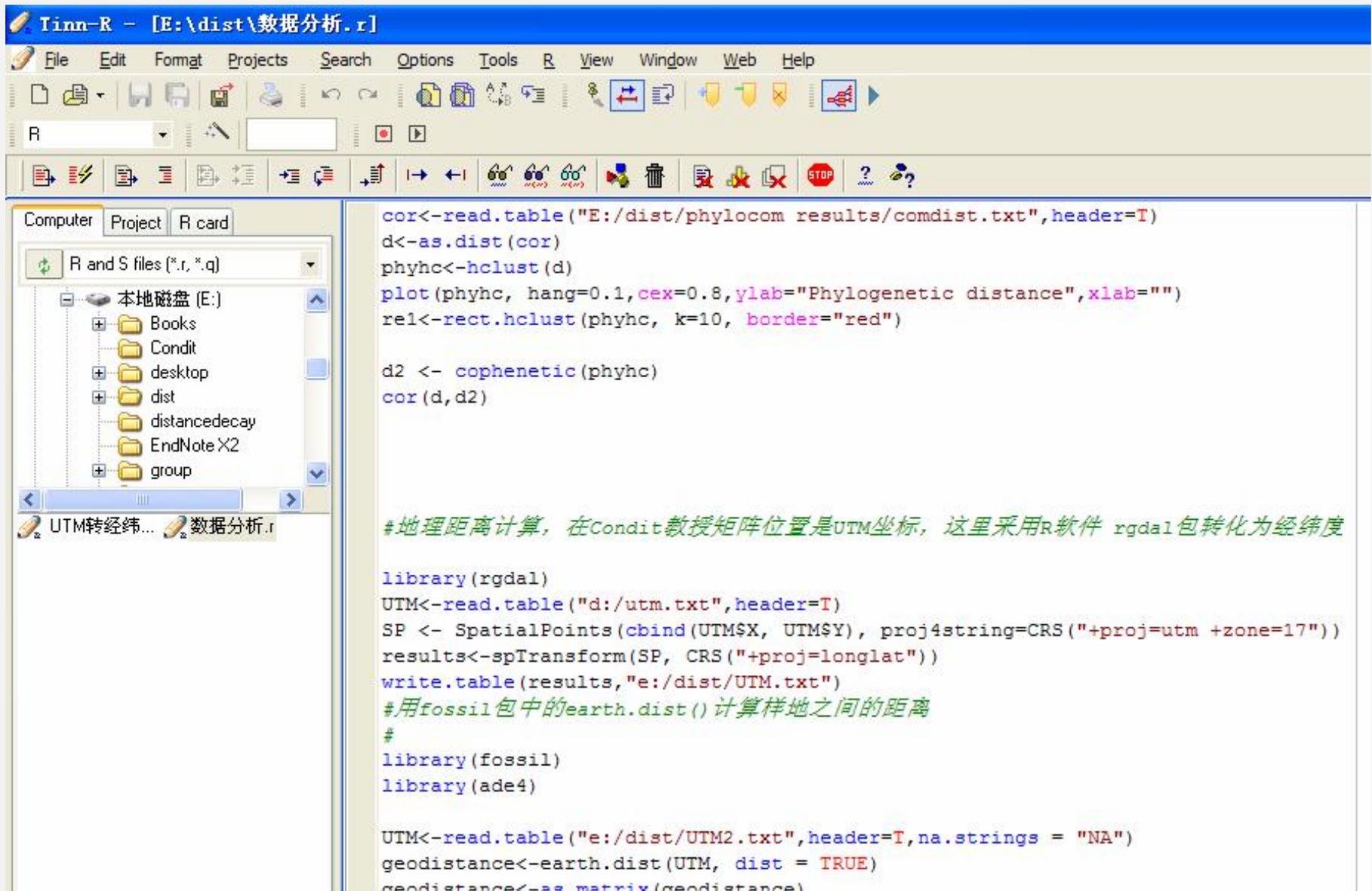
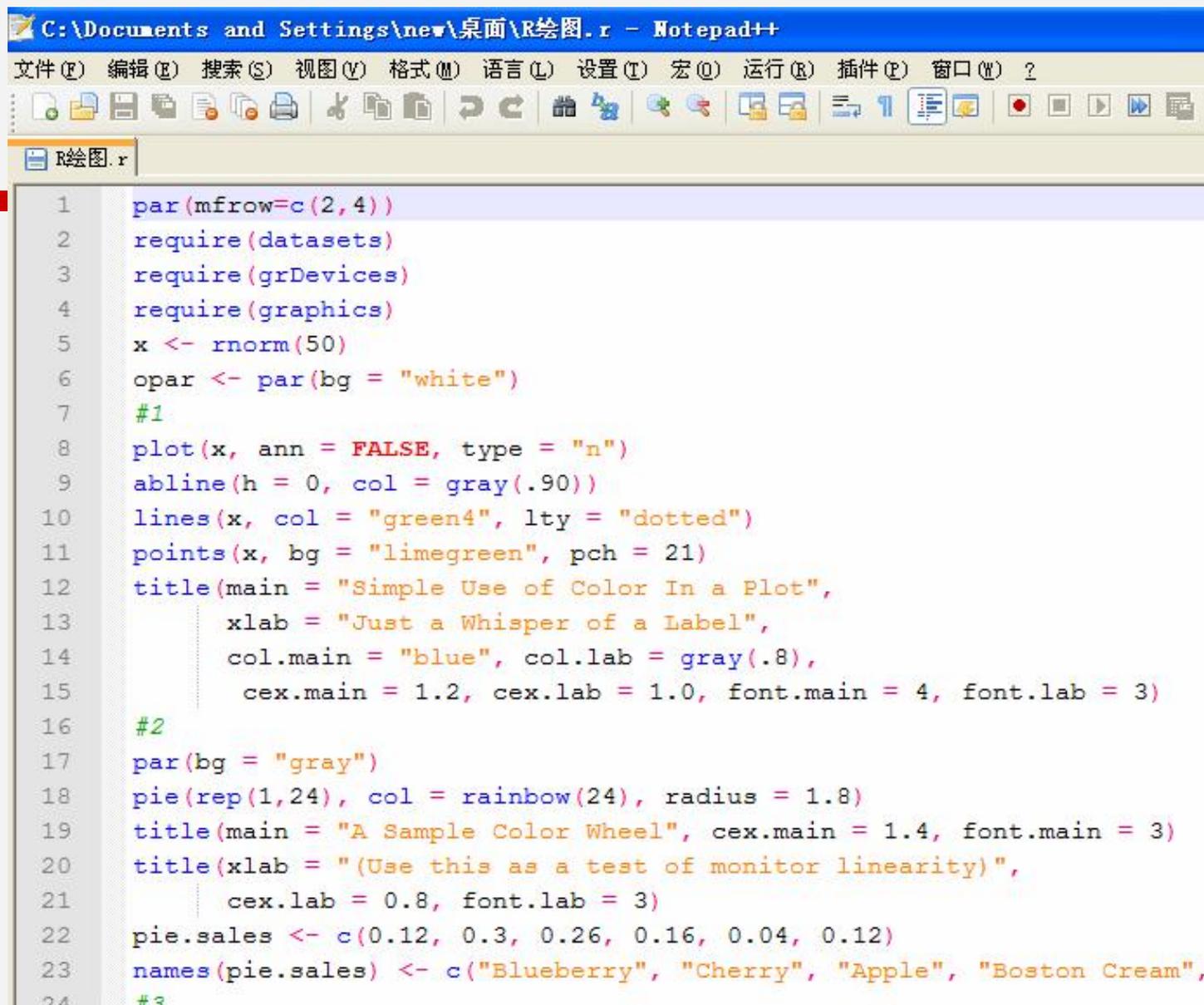


图11 TinnR对R脚本的高亮显示

A screenshot of the Notepad++ text editor window. The title bar reads "C:\Documents and Settings\new\桌面\R绘图.r - Notepad++". The menu bar includes "文件(F)", "编辑(E)", "搜索(S)", "视图(V)", "格式(O)", "语言(L)", "设置(T)", "宏(O)", "运行(R)", "插件(P)", "窗口(W)", and "?". The toolbar contains various icons for file operations and editing. The active window is titled "R绘图.r". The code is as follows:

```
1  par(mfrow=c(2,4))
2  require(datasets)
3  require(grDevices)
4  require(graphics)
5  x <- rnorm(50)
6  opar <- par(bg = "white")
7  #1
8  plot(x, ann = FALSE, type = "n")
9  abline(h = 0, col = gray(.90))
10 lines(x, col = "green4", lty = "dotted")
11 points(x, bg = "limegreen", pch = 21)
12 title(main = "Simple Use of Color In a Plot",
13       xlab = "Just a Whisper of a Label",
14       col.main = "blue", col.lab = gray(.8),
15       cex.main = 1.2, cex.lab = 1.0, font.main = 4, font.lab = 3)
16 #2
17 par(bg = "gray")
18 pie(rep(1,24), col = rainbow(24), radius = 1.8)
19 title(main = "A Sample Color Wheel", cex.main = 1.4, font.main = 3)
20 title(xlab = "(Use this as a test of monitor linearity)",
21       cex.lab = 0.8, font.lab = 3)
22 pie.sales <- c(0.12, 0.3, 0.26, 0.16, 0.04, 0.12)
23 names(pie.sales) <- c("Blueberry", "Cherry", "Apple", "Boston Cream",
24 #3
```

图12 Notepad++对R脚本的高亮显示

编辑器

- R自带的脚本编辑器
 - Editplus (www.editplus.com)
 - TinnR (<http://www.sciviews.org/Tinn-R/>)
 - Ultraedit (www.ultraedit.com/)
 - Emacs (www.gnu.org/software/emacs/)
 - Notepad++ 与NpptoR组合
(<http://notepad-plus.sourceforge.net/>)
 - 记事本或写字板 等等
-

举例-回归分析脚本

- n 对一批涂料进行研究，确定搅拌速度对杂质含量的影响，数据如下，试进行回归分析

表3 搅拌速度对涂料中杂质的影响

转速 rpm	20	22	24	26	28	30	32	34	36	38	40	42
杂质率 %	8.4	9.5	11.8	10.4	13.3	14.8	13.2	14.7	16.4	16.5	18.9	18.5

脚本举例

#将以下代码粘贴到编辑器中，另存为**regression.r**文件。

```
rate<-c(20, 22, 24, 26, 28, 30, 32, 34, 36, 38,  
        40, 42)
```

```
impurity <-c(8.4, 9.5, 11.8, 10.4, 13.3, 14.8,  
            13.2, 14.7, 16.4, 16.5, 18.9, 18.5)
```

```
plot(impurity~rate)
```

```
reg<-lm(impurity~rate)
```

```
abline(reg,col="red")
```

```
summary(reg)
```

运行脚本

三种运行方式

1 通过 `source()` 函数运行

```
source("d:/regression.r")
```

2 通过R脚本编辑器运行

路径: **RGui>File>Open Script** **#Ctrl+R**运行

3 直接粘贴到R控制台

```
ctrl+c, ctrl+v
```

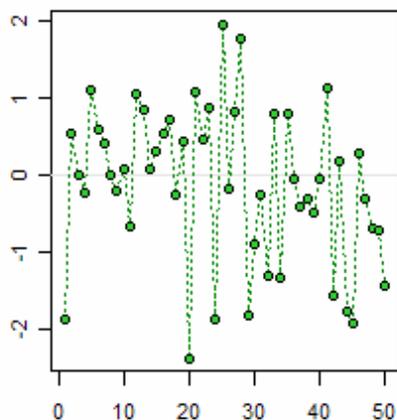
练习十:R脚本运行

将R命令粘贴到记事本中，另存为**regression.R**文件。

分别通过三种方式运行R脚本。

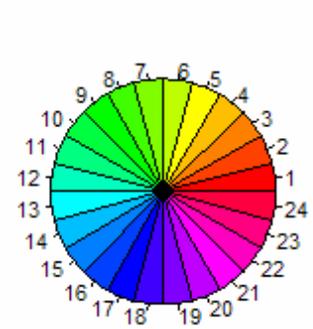
四 R绘图

Simple Use of Color In a Plot



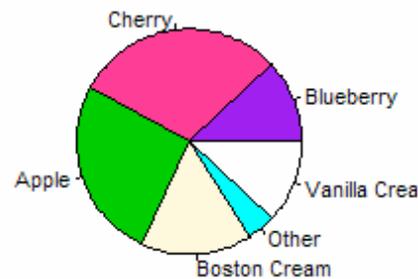
Just a Whisper of a Label

A Sample Color Wheel



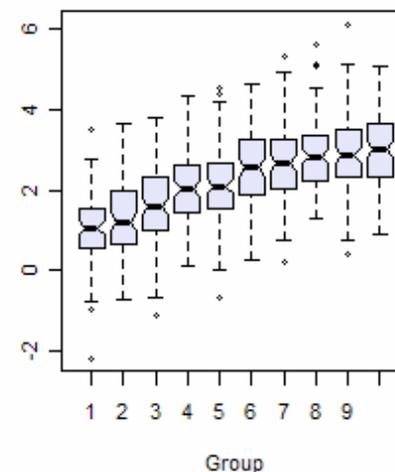
(Use this as a test of monitor linearity)

January Pie Sales

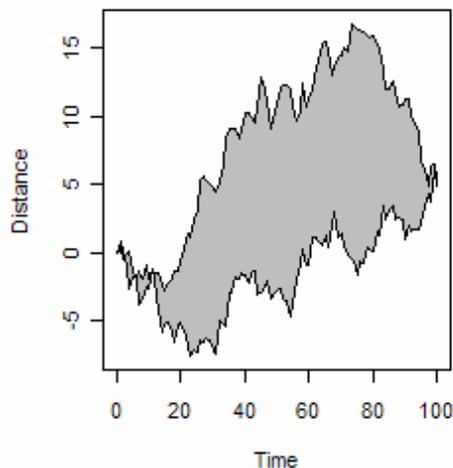


(Don't try this at home kids)

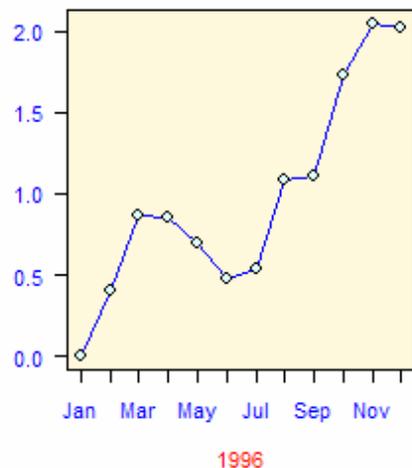
Notched Boxplots



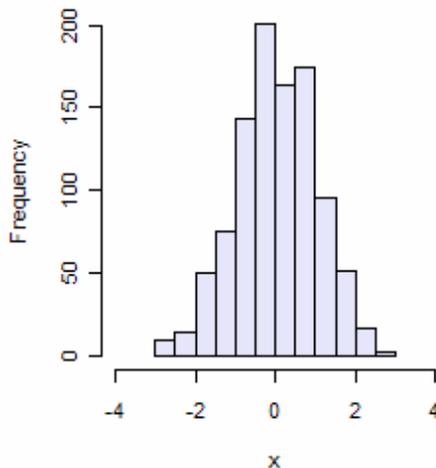
Distance Between Brownian Motions



The Level of Interest in R



1000 Normal Random Variates



A Topographic Map of Maunga Whau

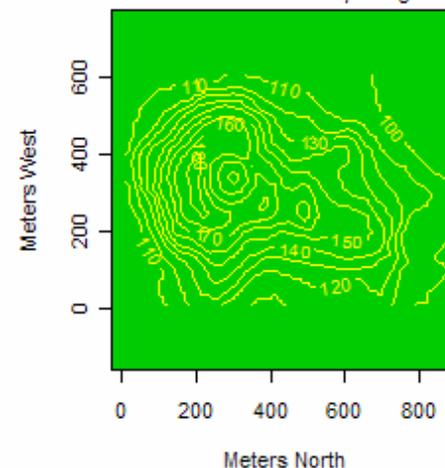


图13 R绘制的图形

R绘图功能

R具备卓越的绘图功能，通过参数设置对图形进行精确控制。绘制的图形能满足出版印刷的要求，可以输出JPEG、TIFF、EPS、emf、pdf、png等各种格式。

绘图是通过绘图函数结合相应的选项完成的。

绘图函数包括高水平绘图函数和低水平绘图函数。

高水平绘图函数

`plot()` 绘制散点图等多种图形

`hist()` 直方图

`boxplot()` 箱线图

`stripchart()` 点图

`barplot()` 条形图

`dotplot()` 点图

`piechart()` 饼图

`interaction.plot()`

`matplot()`

.....

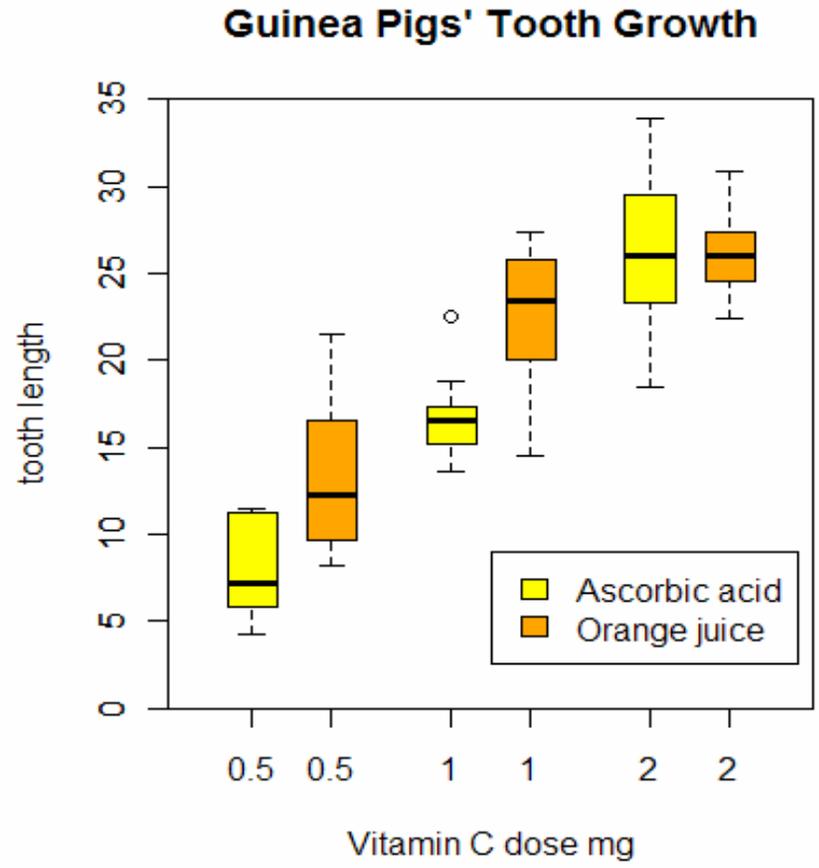
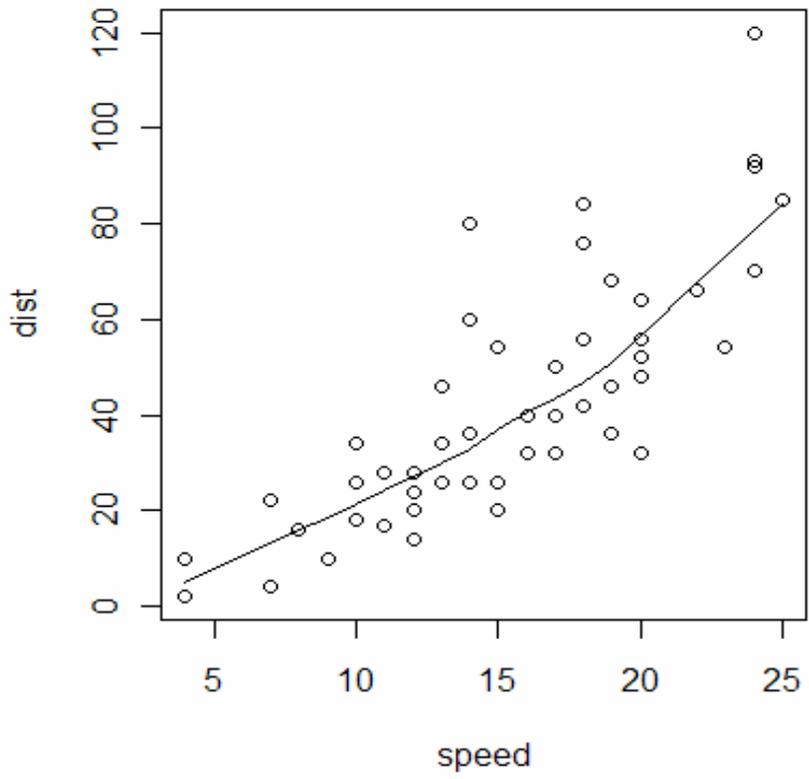


图14 散点图与箱线图举例

低水平绘图函数

lines() 添加线

curve() 添加曲线

abline() 添加给定斜率的线

points() 添加点

segments() 折线

arrows() 箭头

axis() 坐标轴

box() 外框

title() 标题

text() 文字

mtext() 图边文字

... ..

绘图参数

参数用在函数内部，在没有设定值时使用默认值。

font= 字体

lty= 线类型

lwd= 线宽度

pch= 点的类型，

xlab= 横坐标

ylab= 纵坐标

xlim= 横坐标范围

ylim= 纵坐标范围

举例：绘图

生成0到2之间的50个随机数，分别命名为x,y

```
x <- runif(50,0,2)
```

```
y <- runif(50,0,2)
```

绘图：将主标题命名为“散点图”，横轴命名为“横坐标”，纵轴命名为“纵坐标”

```
plot(x, y, main="散点图", xlab="横坐标", ylab="纵坐标")
```

```
text(0.6,0.6,"text at (0.6,0.6)")
```

```
abline(h=.6,v=.6)
```

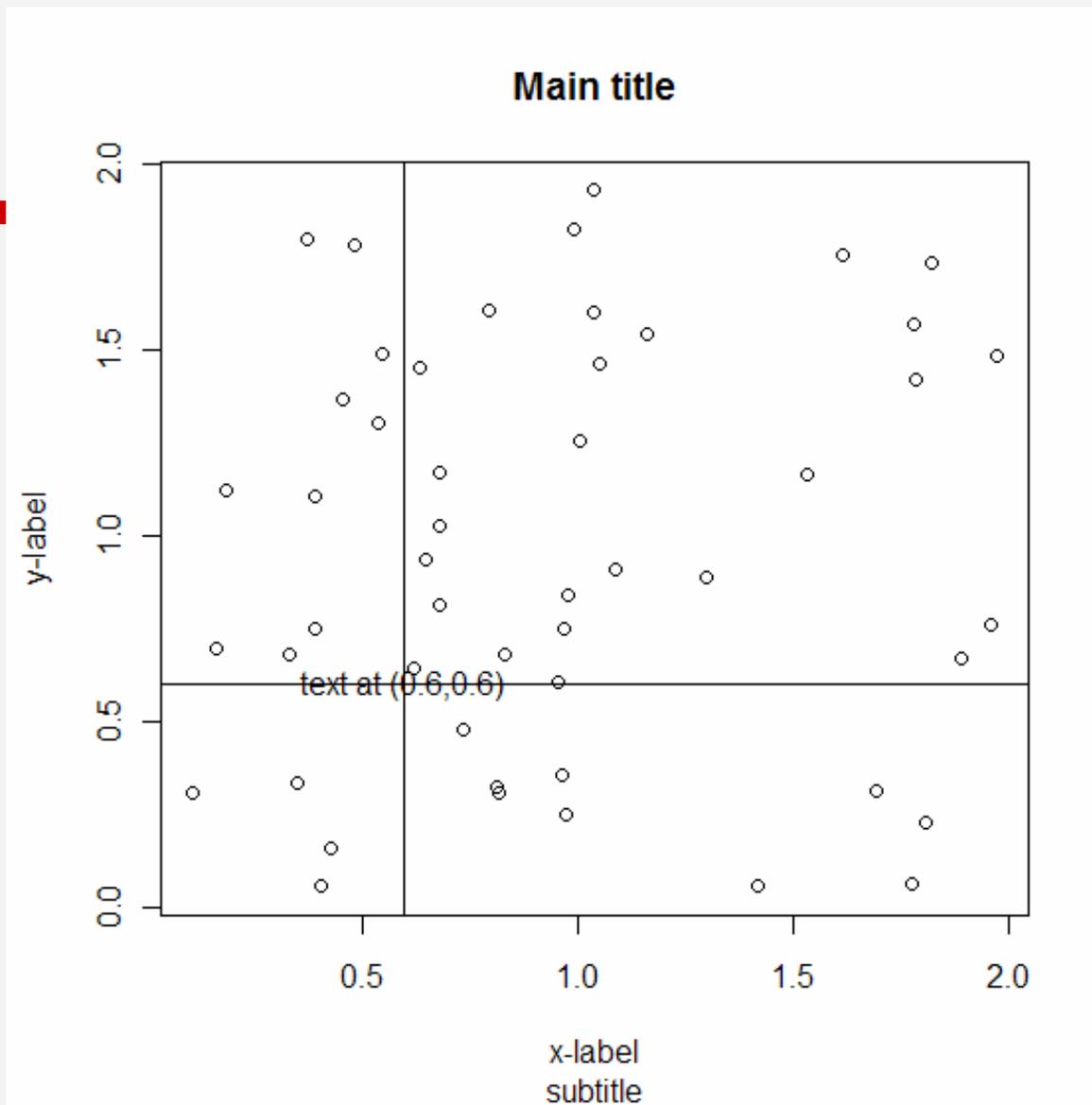


图15 绘图举例

例：分步绘图

```
plot(x, y, type="n", xlab="", ylab="", axes=F) #打  
    开绘图窗口，不绘制任何对象  
points(x,y) #添加坐标点  
axis(1) #添加横轴  
axis(at=seq(0.2,1.8,0.2), side=3) #添加纵轴  
box() #补齐散点图的边框  
title(main="Main title", sub="subtitle", xlab="x-  
    label", ylab="y-label")  
#添加标题、副标题、横轴说明、纵轴说明
```

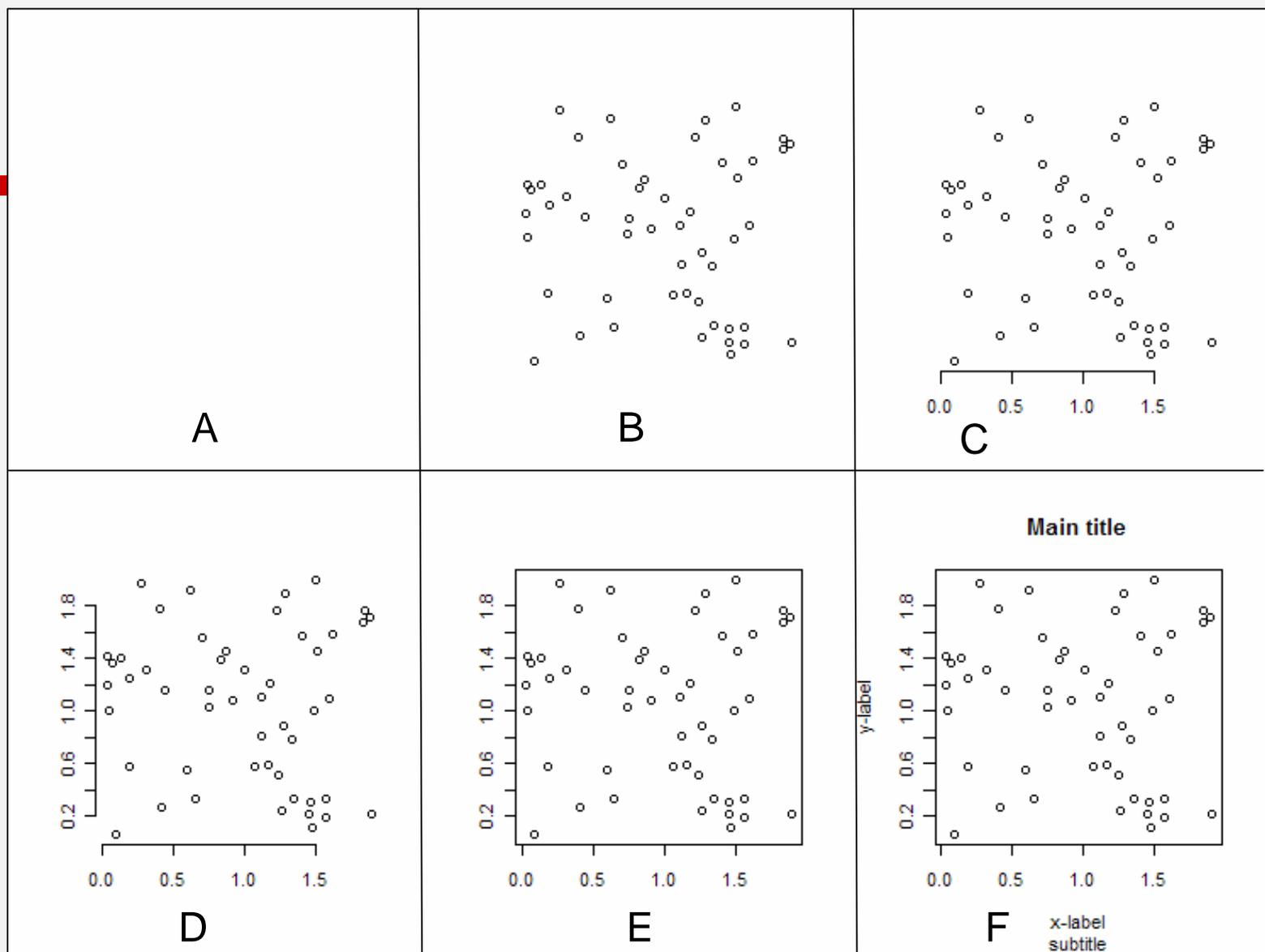


图16 分步绘图

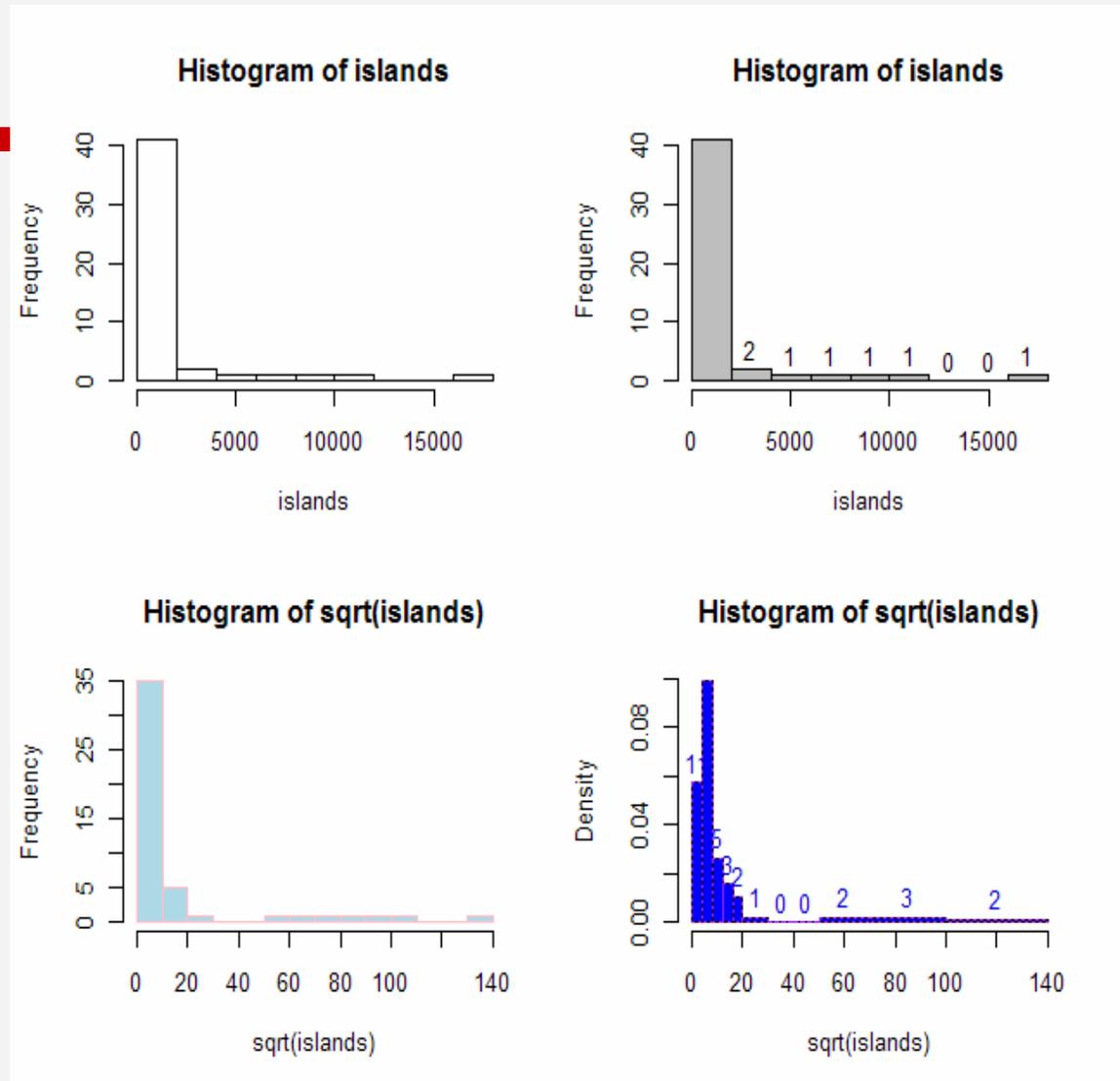
一页多图

`par()`

`par(mfrow=c(2,2))`

...

图17一页多图



在原有图形上添加元素

举例:

```
x <- rnorm(100) # 生成随机数
```

```
hist(x, freq=F) # 绘制直方图
```

```
curve(dnorm(x), add=T) # 添加曲线
```

```
h <- hist(x, plot=F) # 绘制直方图
```

```
ylim <- range(0, h$density, dnorm(0)) # 设定纵轴的取值范围
```

```
hist(x, freq=F, ylim=ylim) # 绘制直方图
```

```
curve(dnorm(x), add=T, col="red") # 添加曲线
```

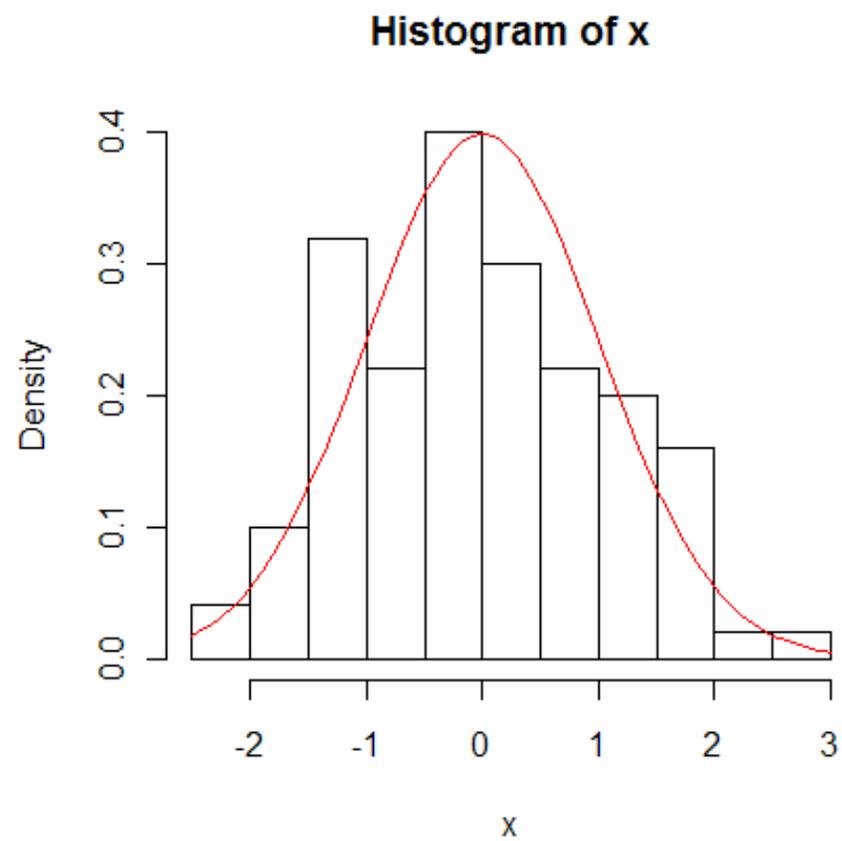
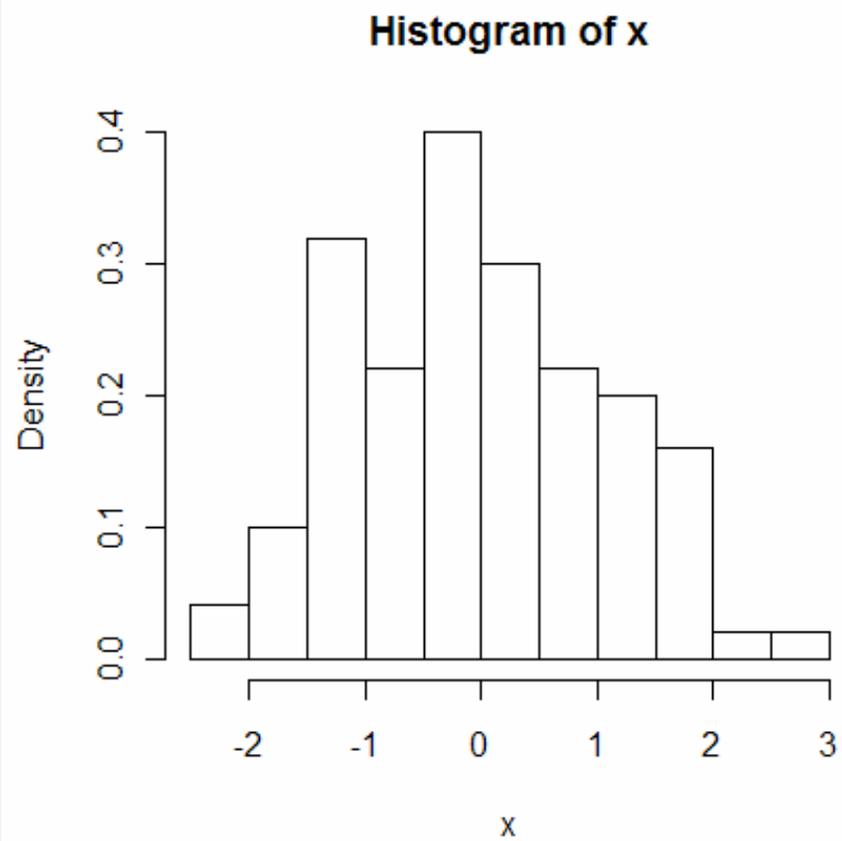


图18 在原有直方图上添加曲线

练习十一：绘图练习

练习1 将**Rplots.r**中的代码拷贝到R控制台中，查看R绘制的图形。

练习2 对例进行回归分析，并绘制散点图，并为散点图添加回归直线。

```
plot(impurity~rate)
reg<-lm(impurity~rate)
abline(reg,col="red")
summary(reg)
```

五 编写函数

编程基础

R可以灵活的编写程序，用户自己编写的程序可以直接调用。R语言编程时无需声明变量的类型，这与C,C++等语言不同。

基本格式

```
function.name <- function(x, y)  
  {  
    表达式  
  }
```

函数内部也可用#添加注释

程序流程控制 if

if 表达式的写法

if(条件) 表达式

if(条件) 表达式1 **else** 表达式2

举例:

```
if(p<=0.05)
```

```
    print("p<=0.05!")
```

```
else
```

```
    print("p>0.05!")
```

循环 for, while

for(变量 **in** 向量) 表达式

```
for(i in 1:10) print(i)
```

while(条件) 表达式

```
i <- 1
```

```
while(i<10)
```

```
{
```

```
  print(i)
```

```
  i <- i + 1
```

```
}
```

函数举例

定义函数:

```
rca1<-function(x,y)
{
  z <- x^2 + y^2;
  result<-sqrt(z) ;
  result;
}
```

调用函数:

```
rca1(3,4)
```

练习十二 编写函数

编写一个函数，给出两个数之后，直接给出这两个数的平方和。

```
sqtest<-function(x, y)  
{  
  z1=x^2;  
  z2=y^2;  
  z3=z1+z2;  
  z3  
}
```

六 数据保存

数据保存

sink()

unlink()

若有**LaTeX**基础，可以用

Sweave() 函数

该函数能将脚本、程序说明和运算结果直接保存成**.tex**文件，用**LaTeX**编译成**pdf**文件。

总结

- R是开源的统计绘图软件，也是一种自解释的语言，有大量的程序包可以利用。
 - R中的向量、列表、数组、统计结果等都是对象，可以方便的生成所需对象，并进行筛选。
 - R脚本是输入的多个命令行。
 - R具有精确控制的绘图功能，生成的图可以另存为多种格式。
 - R编写函数无需声明变量的类型，能利用循环、条件语句，控制程序的流程。
-

推荐的教材

- Peter Dalgaard *Introductory statistics with R*
 - Crawley *Statistics an introduction using R*
 - E. Paradis *R for Beginners*
 - Verzani *SimpleR.*
 - D G Rossiter *Introduction to the R Project for Statistical Computing for use at ITC*
 - J. Maindonald *Using R for data analysis and graphics introduction code and commentary using R*
 - Venables, W. N. & Ripley, B. D. *Modern Applied Statistics with S*
-

R资源

R主页: <http://www.r-project.org>

CRAN 北京镜像(数学所)

<http://ftp.ctex.org/mirrors/CRAN/>

群落生态学数据分析

<http://ecology.msu.montana.edu/labdsv/R/labs/>

Statistics with R

http://zoonek2.free.fr/UNIX/48_R/all.html

NCEAS提供的R资源

<http://www.nceas.ucsb.edu/scicomp/RProgTutorialsLatest.html>

谢谢各位
敬请指正!
