

Chapter-08. 高级语言程序设计

```
5 import math
6 import pytz
7 import io, base64
8
9
10 class AdmissionExtensionOnlineController(http.Controller):
11     @http.route('/get/type_wise_program', website=True, auth='none')
12     def type_wise_program(self, **kwargs):
13         if len(kwargs['types']) <= 0:
14             return "None"
15
16         types = kwargs['types']
17         program_list = []
18         domain = []
19
20         if types == 'local_bachelor_program_hsc':
21             domain = [('course_id.is_local_bachelor_program_hsc', '=')]
22         elif types == 'local_bachelor_program_a_level':
23             domain = [('course_id.is_local_bachelor_program_a_level', '=')]
24         elif types == 'local_bachelor_program_diploma':
25             domain = [('course_id.is_local_bachelor_program_diploma', '=')]
26         elif types == 'local_masters_program_bachelor':
27             domain = [('course_id.is_local_masters_program_bachelor', '=')]
28         elif types == 'international_bachelor_program':
29             domain = [('course_id.is_international_bachelor_program', '=')]
30         elif types == 'international_masters_program':
31             domain = [('course_id.is_international_masters_program', '=')]
32
33         domain.append(('state', '=', 'application'))
34
35         admission_register_list = http.request.registry.get('admission_register_list')
36
37         for program in program_list:
38             if program['id'] in admission_register_list:
39                 program['state'] = 'registered'
40
41         return {
42             'program_list': program_list,
43             'domain': domain
44         }
```

本章内容提要

■ 8.1 大数据与数据挖掘

■ 8.2 常见的数据挖掘工具

■ 8.3 R语言是数据挖掘的一把利器

■ 8.4 R语言基础-统计分析与绘图

■ 8.5 R语言应用：可视化与图像处理

■ 8.6 本章小结



第1节：大数据与数据挖掘

- 当前全球形势复杂多变，一场以大数据与信息技术为主导的科技风暴席卷全球，互联网技术创新、应用服务和两化深度融合方兴未艾。
- 2015年8月，国务院出台了《促进大数据发展行动纲要》，明确提出“加快政府数据开放共享，推动资源整合”的主要任务。
- 2017年9月，重庆市人民政府在新发布的《重庆市“十三五”信息化规划》中更是明确将“大数据技术”作为一项重大战略部署，并提出加快推进重庆国家级大数据综合试验区建设，以期提高现有相关产业的智能化水平，促进经济社会的全面快速发展。



R and Data Mining

----Ronald Varian

The great beauty of R, is that you can modify it to do all sorts of things.



Using **R** !

*You have a lot of prepackaged stuff
that's already available, so you're
standing on the shoulders of giants.*

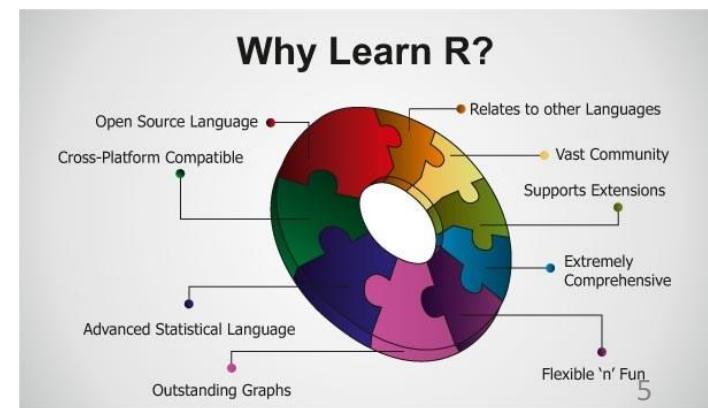


*So if you wait for the **robins**,
spring will be over.*

---- Warren Buffett



如果你还不学一些**R**, 大数
据对你来说就快结束了。





第2节：常见的数据挖掘工具

RapidMiner



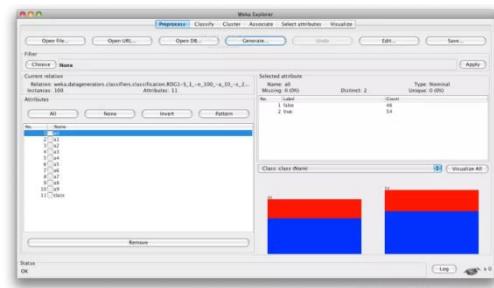
http://blog.csdn.net/Aylee_liu

该工具是用Java语言编写的，通过基于模板的框架提供先进的分析技术。该款工具最大的好处就是，用户无需写任何代码。它是作为一个服务提供，而不是一款本地软件。值得一提的是，该工具在数据挖掘工具榜上位列榜首。

另外，除了数据挖掘，RapidMiner还提供如数据预处理和可视化、预测分析和统计建模、评估和部署等功能。更厉害的是它还提供来自WEKA（一种智能分析环境）和R脚本的学习方案、模型和算法。

RapidMiner分布在AGPL开源许可下，可以从SourceForge上下载。SourceForge是一个开发者进行开发管理的集中式场所，大量开源项目在此落户，其中就包括维基百科使用的MediaWiki。

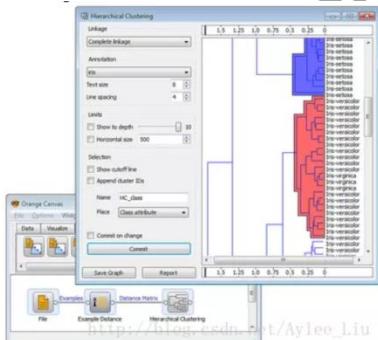
WEKA



WEKA原生的非Java版本主要是为了分析农业领域数据而开发的。该工具基于Java版本，是非常复杂的，并且应用在许多不同的应用中，包括数据分析以及预测建模的可视化和算法。与RapidMiner相比优势在于，它在GNU通用公共许可证下是免费的，因为用户可以按照自己的喜好选择自定义。

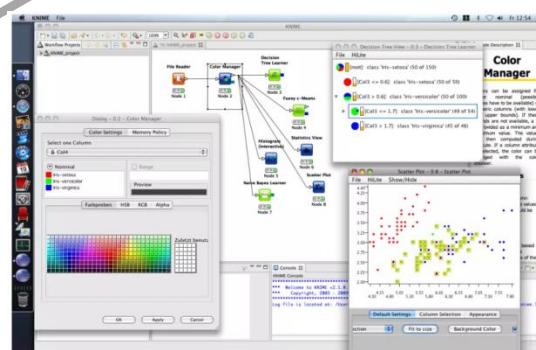
WEKA支持多种标准数据挖掘任务，包括数据预处理、收集、分类、回归分析、可视化和特征选取。

Orange



Python之所以受欢迎，是因为它简单易学并且功能强大。如果你是一个Python开发者，当涉及到需要找一个工作的工具时，那么没有比Orange更合适的了。它是一个基于Python语言，功能强大的开源工具，并且对初学者和专家级的大神均适用。

KNIME



数据处理主要有三个部分：提取、转换和加载。而这三者KNIME都可以做到。KNIME为您提供了一个图形化的用户界面，以便对数据节点进行处理。它是一个开源的数据分析、报告和综合平台，同时还通过其模块化数据的流水线概念，集成了各种机器学习的组件和数据挖掘，并引起了商业智能和财务数据分析的注意。

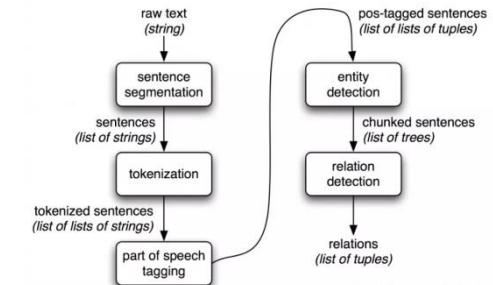
KNIME是基于Eclipse，用Java编写的，并且易于扩展和补充插件。其附加功能可随时添加，并且其大量的数据集成模块已包含在核心版本中。



http://blog.csdn.net/Aylee_liu

如果我告诉你R项目，一个GNU项目，是由R（R-programming简称，以下统称R）自身编写的，你会怎么想？它主要是由C语言和FORTRAN语言编写的，并且很多模块都是由R编写的，这是一款针对编程语言和软件环境进行统计计算和制图的免费软件。R语言被广泛应用于数据挖掘，以及开发统计软件和数据分析中。近年来，易用性和可扩展性也大大提高了R的知名度。

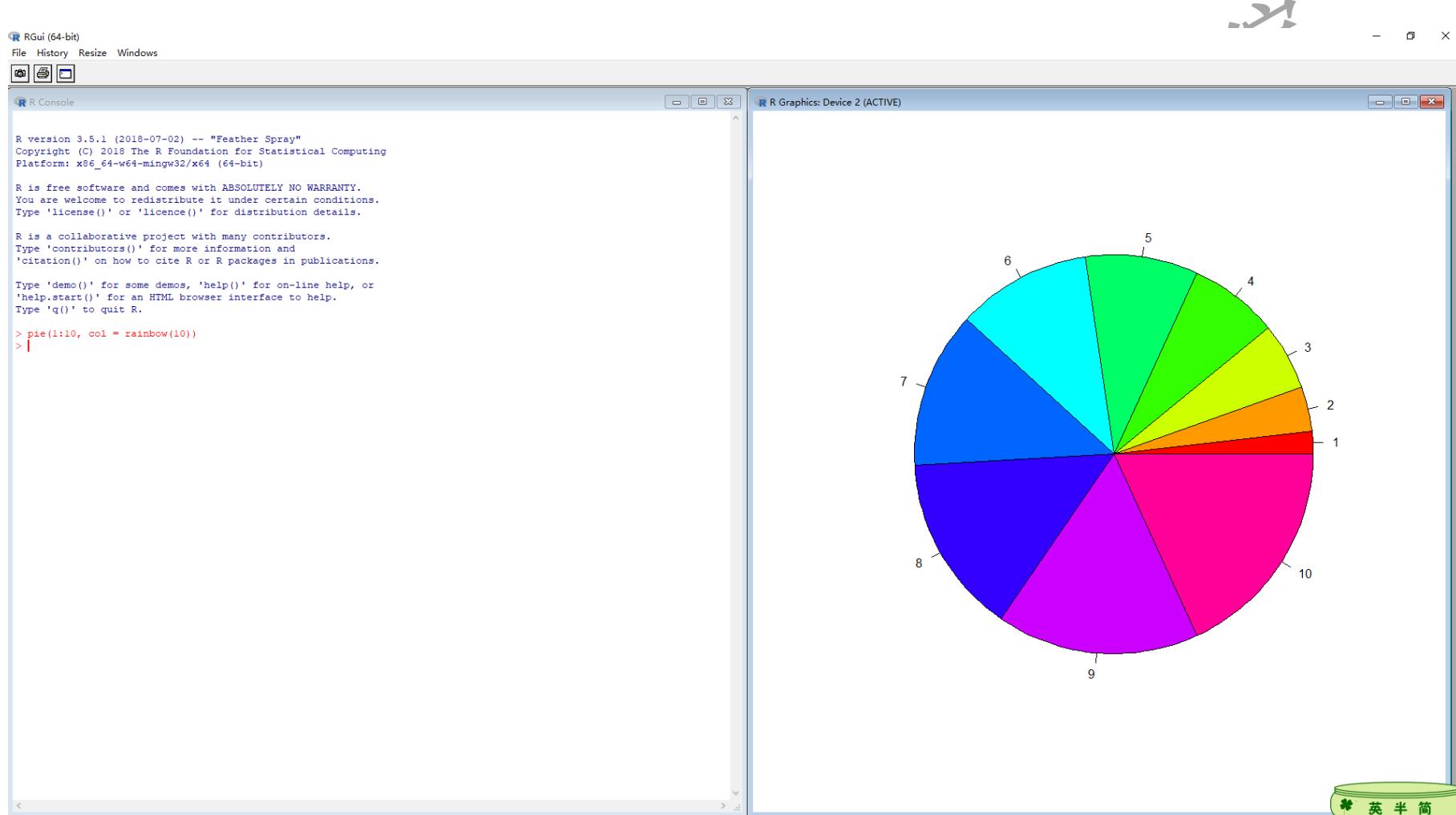
NLTK



http://blog.csdn.net/Aylee_liu

当涉及到语言处理任务，没有什么可以打败NLTK。NLTK提供了一个语言处理工具，包括数据挖掘、机器学习、数据抓取、情感分析等各种语言处理任务。而您需要做的只是安装NLTK，然后将一个包拖拽到您最喜爱的任务中，您就可以去做其他事了。因为它是用Python语言编写的，你可以在上面建立到它的连接，还可以自定义它的小任务。

❖ 2.1 R语言的基础运行环境



```
31
32 # (2) Construct our dataset about MSC msc.data.
33
34 scRNA.dat <- read.csv("MSC_SWH_gene_bar.csv", header = TRUE, row.names = 1)
35
36 # transform the ensembl IDs to official symbols.
37
38 symbol_seq <- mapIds(org.Hs.eg.db,
39                         keys = row.names(scRNA.dat),
40                         column = "SYMBOL",
41                         keytype = "ENSEMBL",
42                         multiVals = "first")
43 #. length(symbol_seq)
44 #. length(na.omit(symbol_seq))
45
46 # Find the positions of NA values in symbol_seq.
47
48 na.pos <- which(symbol_seq %in% NA)
49
50 # Replace the symbols of NA values with Ensembl IDs.
51
52 symbol_seq[na.pos] <- rownames(scRNA.dat)[na.pos]
53
54 #
55
56 stat <- table(symbol_seq)
57
58 # length(unique(na.omit(symbol_seq)))
59
60 # Find the duplicates of symbols, times are more than once.
61
62 dup_seq <- names(which(stat > 1))
63
64 # Replace the gene symbols with the Ensembl IDs, for duplicated genes, one by one.
65
66 for (i in 1:length(dup_seq)) {
67
68   p <- grep(dup_seq[i], symbol_seq)
69
70   symbol_seq[p] <- rownames(scRNA.dat)[p]
```

❖ 2.2 R语言的集成开发环境：Rstudio

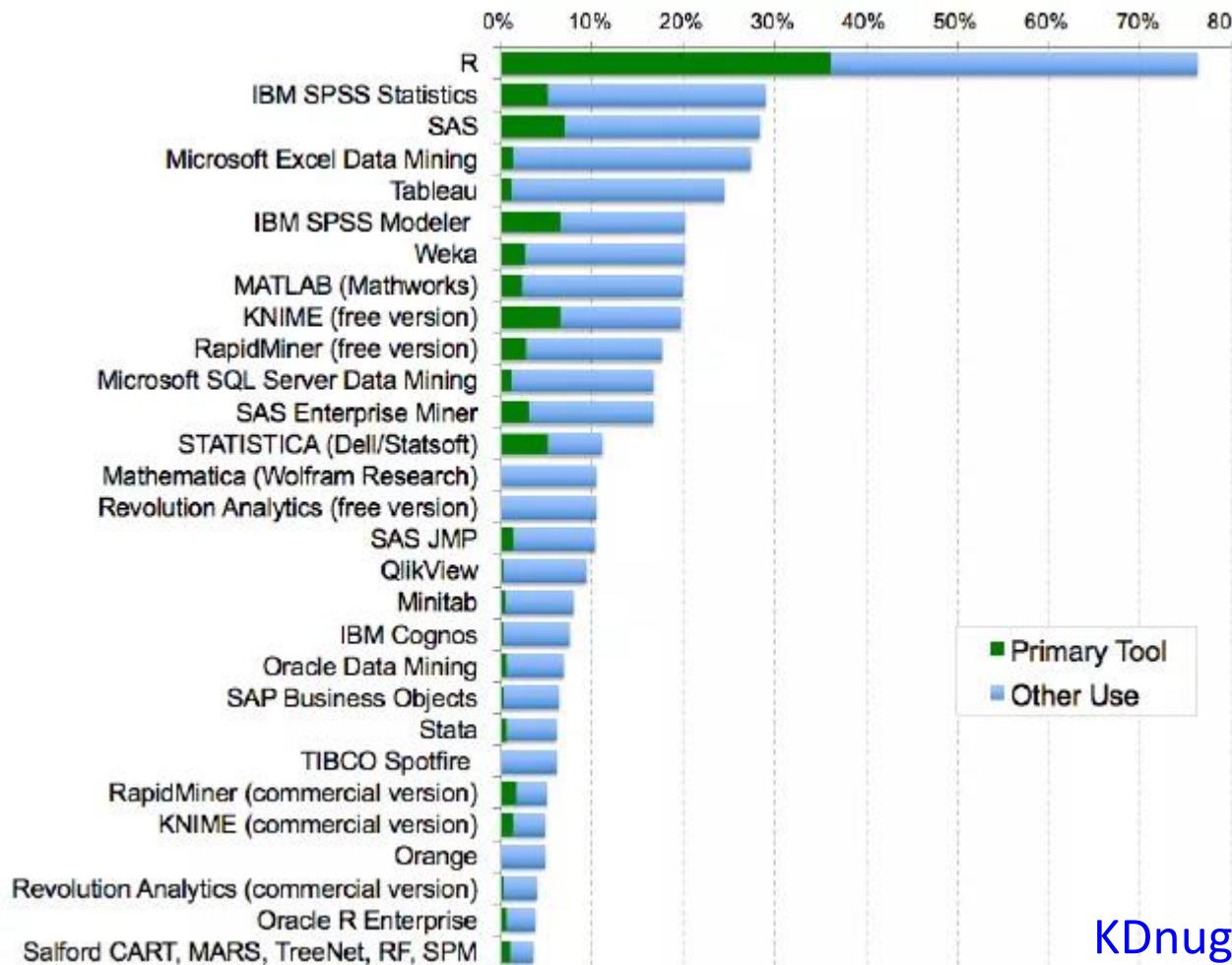
The screenshot displays the RStudio interface with the following components:

- Top Bar:** Shows the RStudio logo and the title "RStudio".
- Menu Bar:** Includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help.
- Toolbar:** Contains icons for file operations like Open, Save, and Run, along with Source On Save and Addins buttons.
- Code Editor:** Displays R script code for data manipulation and plotting. The code uses ggplot2 and scales packages to create a scatter plot with a loess smoother and two histograms for depth and price.
- Environment Tab:** Shows the global environment with objects like col_mat, diamonds, p1, p1_npg, p2, p2_npg, col_lancet, and mypal.
- Data Tab:** Lists data frames and lists: col_mat (3 obs. of 9 variables), diamonds (53940 obs. of 10 variables), p1, p1_npg, p2, p2_npg, col_lancet, and mypal.
- Values Tab:** Shows character vectors for col_lancet and mypal.
- Plots:** Two plots are shown. The left plot is a scatter plot of price vs. table with a loess smoother and colored by cut quality. The right plot is a histogram of depth with bars colored by cut quality.
- Console:** Shows the R command history and output, including a warning message about a manual palette size limit.



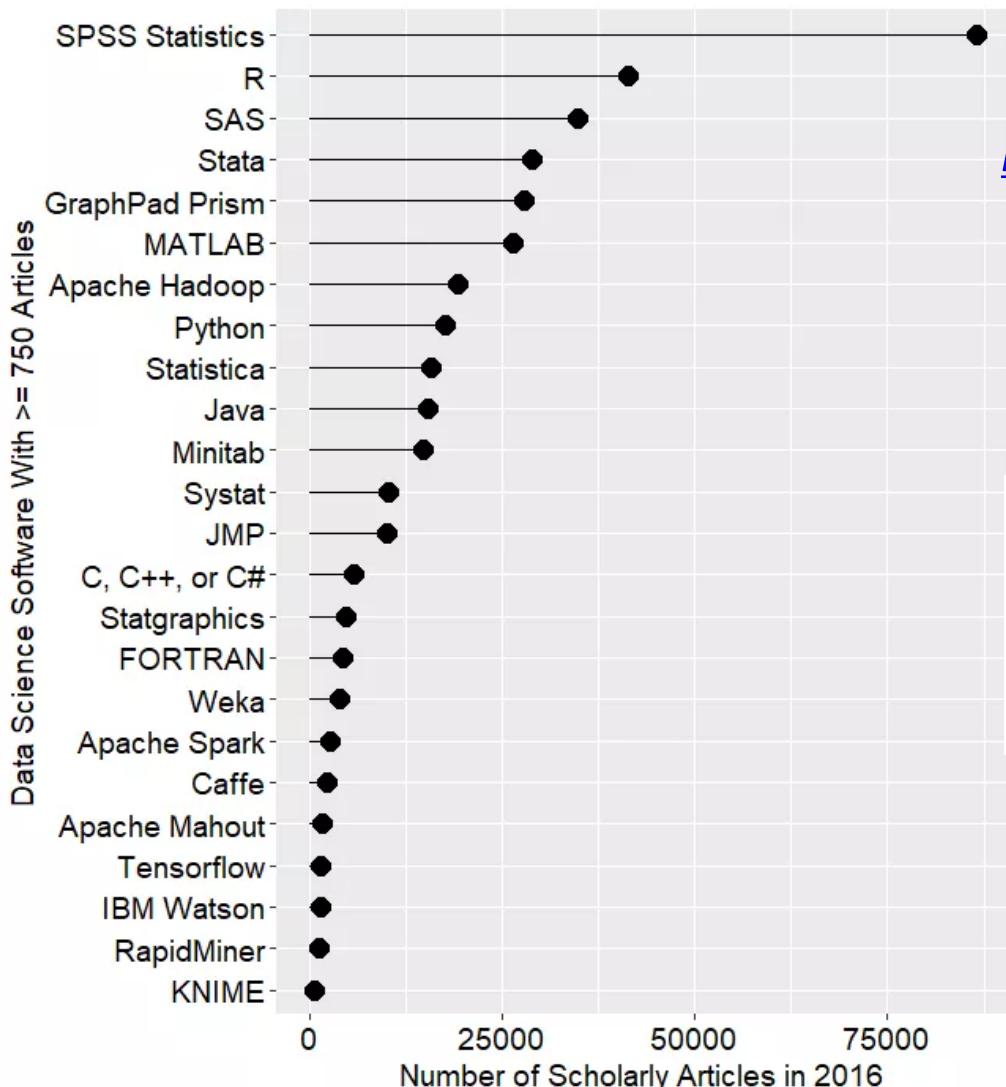
第3节：R语言是数据挖掘的一把利器

(1) R是备受数据科学家青睐的数据挖掘工具



KDnuggets' 2015

(2) R也是学术界所推崇的数据挖掘工具



Orchestrating high-throughput genomic analysis with Bioconductor

Wolfgang Huber¹, Vincent J Carey^{2,3}, Robert Gentleman⁴, Simon Anders¹, Marc Carlson⁵, Benilton S Carvalho⁶, Hector Corrada Bravo⁷, Sean Davis⁸, Laurent Gatto⁹, Thomas Girke¹⁰, Raphael Gottardo¹¹, Florian Hahne¹², Kasper D Hansen^{13,14}, Rafael A Irizarry^{3,15}, Michael Lawrence⁴, Michael I Love^{3,15}, James MacDonald¹⁶, Valerie Obenchain⁵, Andrzej K Oleś¹, Hervé Pagès⁵, Alejandro Reyes¹, Paul Shannon⁵, Gordon K Smyth^{17,18}, Dan Tenenbaum⁵, Lori Waldron¹⁹ & Martin Morgan⁵

Nature Methods, 2015, 12(2):115-21

Bioconductor is an open-source, open-development software project for the analysis and comprehension of high-throughput data in genomics and molecular biology. The project aims to enable interdisciplinary research, collaboration and rapid development of scientific software. Based on the statistical programming language R, Bioconductor comprises 934 interoperable packages contributed by a large, diverse community of scientists. Packages cover a range of bioinformatic and statistical applications. They undergo formal initial review and continuous automated testing. We present an overview for prospective users and contributors.

programming environment offered by the R project². It supports many types of high-throughput sequencing data (including DNA, RNA, chromatin immunoprecipitation, Hi-C, methylomes and ribosome profiling) and associated annotation resources; contains mature facilities for microarray analysis³; and covers proteomic, metabolomic, flow cytometry, quantitative imaging, cheminformatic and other high-throughput data. Bioconductor enables the rapid creation of workflows combining multiple data types and tools for statistical inference, regression, network analysis, machine learning and visualization at all stages of a project from data generation to publication.

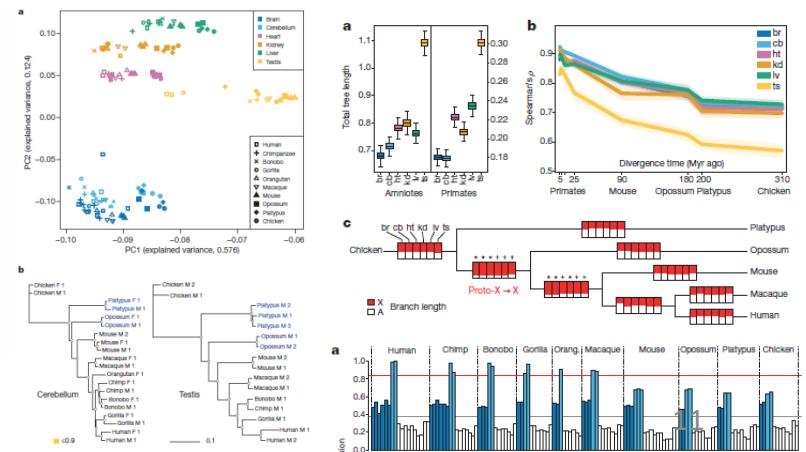
Bioconductor is also a flexible software engineering

ARTICLE

doi:10.1038/nature10532

The evolution of gene expression levels in mammalian organs

David Brawand^{1,2*}, Magali Soumillon^{1,2*}, Anand Mehta^{1,2}, Philipp Jaitin^{1,2}, Sushmita Sengupta¹, Paola Di Pio¹, Manuela Weier¹, Daniel Reiter¹, Alexander Zeller¹, Michael A. Patti¹, Daniel Zeller¹, Daniel Hausslein¹, Frank Grützner³, Claus Fieguth⁴, Matthias Niesler⁴, Svante à比ö⁴ & Heinrich Kaessmann¹





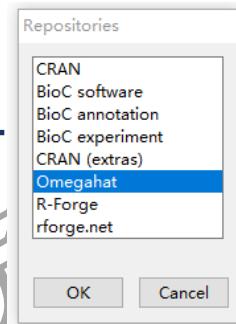
第4节：R语言基础——统计分析与可视化

- 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。



❖ 4.1 R语言及R包典藏库

- 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**13468**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。



❖ 4.2 R语言的下载与安装

(支持Linux, MacOS以及Windows等几乎所有的操作系统)



The R Project for Statistical Computing

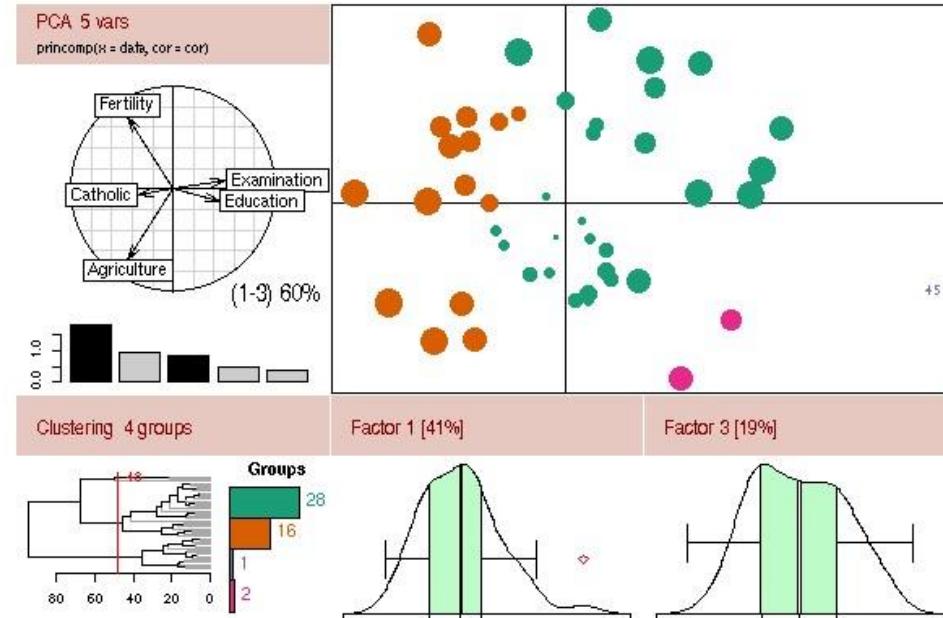
[About R](#)
[What is R?](#)
[Contributors](#)
[Screenshots](#)
[What's new?](#)

[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](#)



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 Mac OS. 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/>

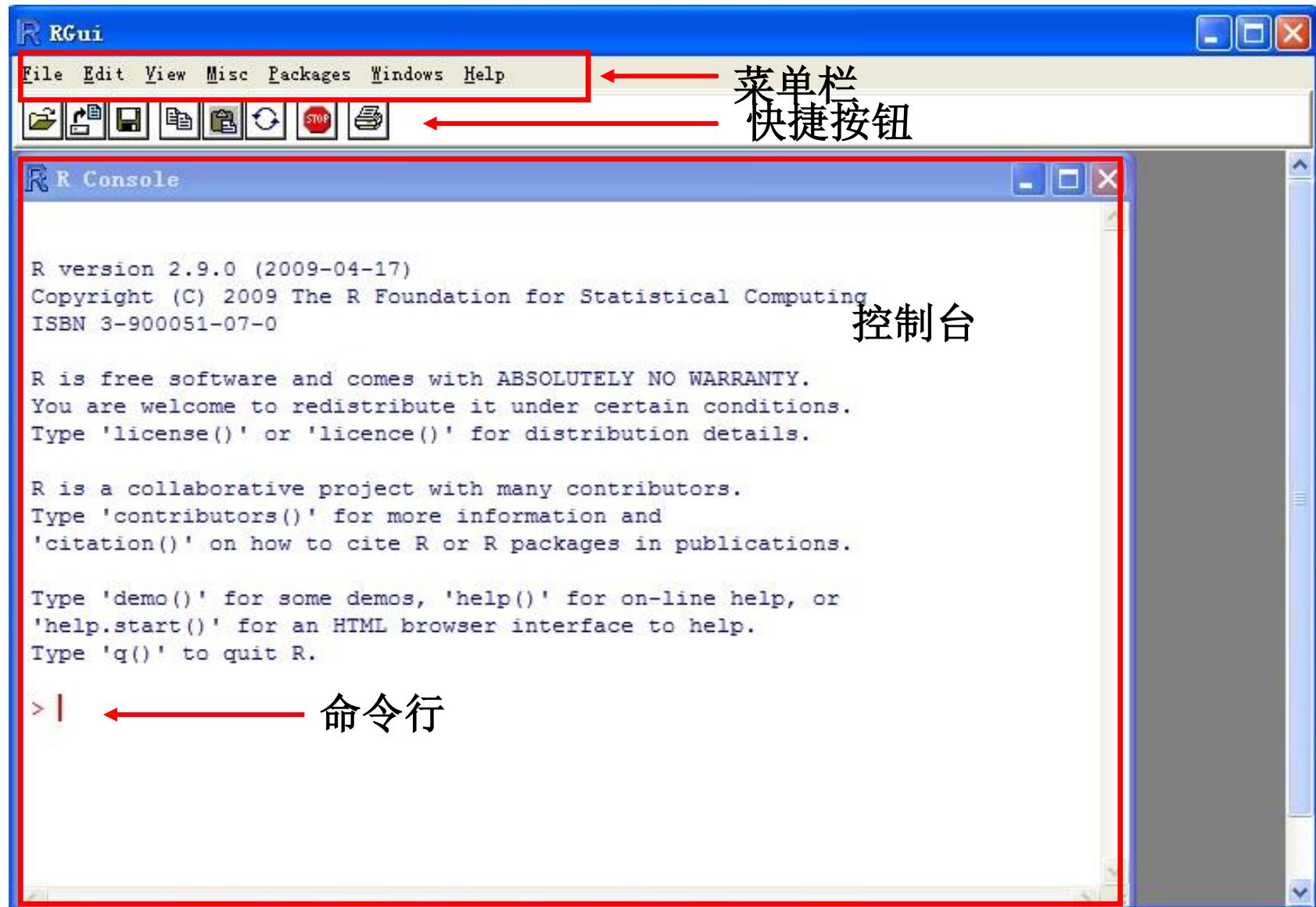


图2 R登陆界面(Windows版)

路径： 开始>所有程序>R 2.9.2

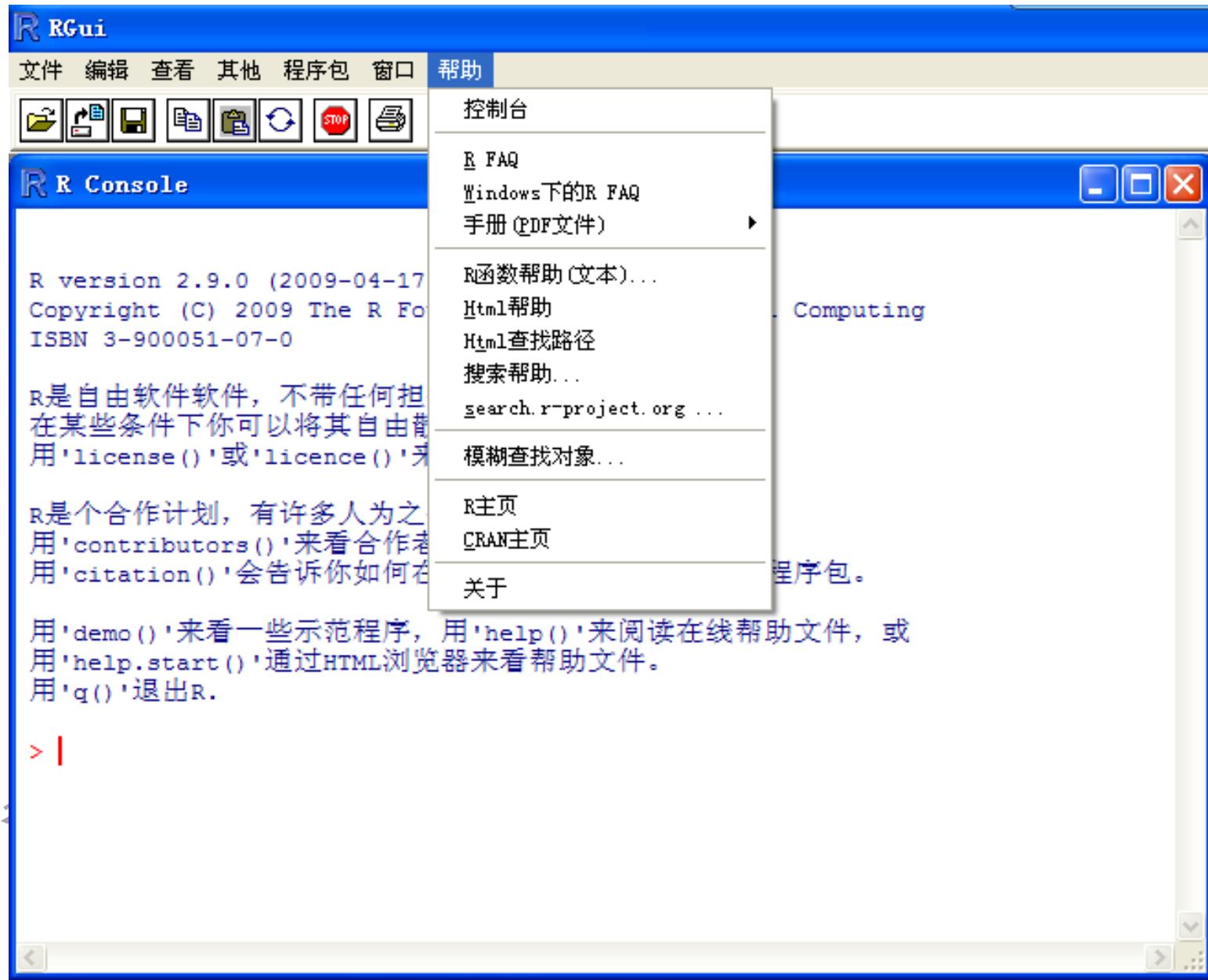


图3 R Gui 的菜单介绍

❖ 4.3 R语言程序包

(R packages是R语言的精华所在！)

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
apeTreeshape-	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



Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
ExtremeValue	Extreme Value Analysis
Finance	Empirical Finance
FunctionalData	Functional Data Analysis
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
MetaAnalysis	Meta-Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
NumericalMathematics	Numerical Mathematics
OfficialStatistics	Official Statistics & Survey Methodology
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods
Psychometrics	Psychometric Models and Methods
ReproducibleResearch	Reproducible Research
Robust	Robust Statistical Methods

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
```

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

❖ 4.4 R语言的函数与对象

(1) R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`
`...`

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

R函数调用及其选项

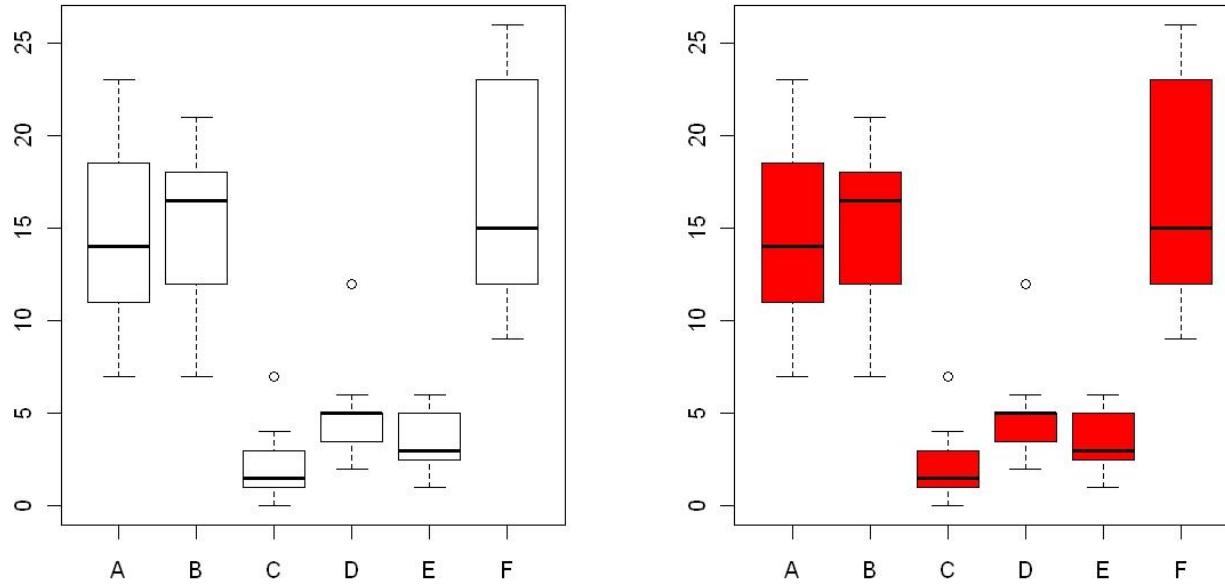


图7 箱线图修饰前后(左: 默认值, 右: 修改属性后)

```
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**

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")
```

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：

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

调用函数：

```
rcal(3,4)
```

Eg-1 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
}
```

(2) R中的对象

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)

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

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

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

字符串 Row names

每列可看做带名称的向量

字符串

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，

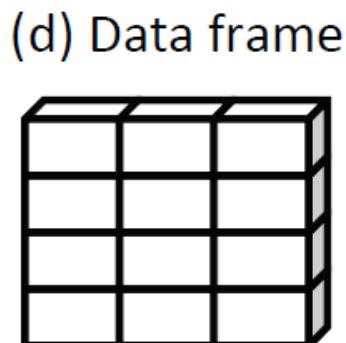
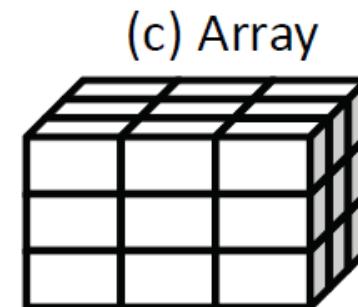
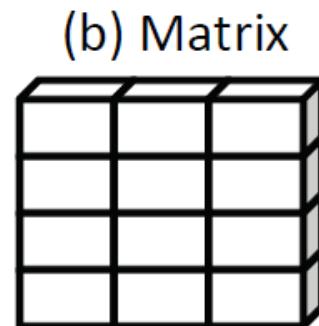
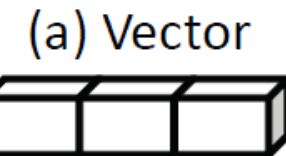
a<-c(10,15,21,18) 该向量为数值型，长度为**4**。

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

对象的类别

必修课



(e) List

Columns can be different modes

Vectors
Arrays
Data frames
Lists

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()
```

Listing 2.1 Creating matrices

```
> y <- matrix(1:20, nrow=5, ncol=4)           1 Create a 5x4 matrix
> y
     [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20
> 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(rnames, cnames))
> mymatrix
      C1  C2
R1  1 26
R2 24 68
> mymatrix <- matrix(cells, nrow=2, ncol=2, byrow=FALSE,
                      dimnames=list(rnames, cnames))           2 2x2 matrix filled
                                                               by rows
> mymatrix
      C1  C2
R1  1 24
R2 26 68
                                                               3 2x2 matrix filled
                                                               by columns
```

Listing 2.2 Using matrix subscripts

```
> x <- matrix(1:10, nrow=2)
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
> x[2,]
[1] 2 4 6 8 10
> x[,2]
[1] 3 4
> x[1,4]
[1] 7
> x[1, c(4,5)]
[1] 7 9
```

重庆师范大

Listing 2.3 Creating an array

```
> dim1 <- c("A1", "A2")
> dim2 <- c("B1", "B2", "B3")
> dim3 <- c("C1", "C2", "C3", "C4")
> z <- array(1:24, c(2, 3, 4), dimnames=list(dim1, dim2, dim3))
> z
, , C1
```

```
      B1  B2  B3
A1    1    3    5
A2    2    4    6
```

```
, , C2
```

```
      B1  B2  B3
A1    7    9   11
A2    8   10   12
```

```
, , C3
```

```
      B1  B2  B3
A1  13  15  17
A2  14  16  18
```

```
, , C4
```

```
      B1  B2  B3
A1  19  21  23
A2  20  22  24
```



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:
```

Listing 2.4 Creating a data frame

```
> patientID <- c(1, 2, 3, 4)
> age <- c(25, 34, 28, 52)
> diabetes <- c("Type1", "Type2", "Type1", "Type1")
> status <- c("Poor", "Improved", "Excellent", "Poor")
> patientdata <- data.frame(patientID, age, diabetes, status)
> patientdata
  patientID age diabetes      status
1          1   25     Type1      Poor
2          2   34     Type2  Improved
3          3   28     Type1 Excellent
4          4   52     Type1      Poor
```



Listing 2.5 Specifying elements of a data frame

```
> patientdata[1:2]
  patientID age
1          1   25
2          2   34
3          3   28
4          4   52
> patientdata[c("diabetes", "status")]
  diabetes      status
1     Type1      Poor
2     Type2  Improved
3     Type1 Excellent
4     Type1      Poor
> patientdata$age
[1] 25 34 28 52
```

① Indicates age variable in patient data frame

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)**

Listing 2.7 Creating a list

```
> g <- "My First List"
> h <- c(25, 26, 18, 39)
> j <- matrix(1:10, nrow=5)
> k <- c("one", "two", "three")
> mylist <- list(title=g, ages=h, j, k)      ← Create list
> mylist                                ← Print entire list
$title
[1] "My First List"

$ages
[1] 25 26 18 39

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

[[4]]
[1] "one"    "two"    "three"

> mylist[[2]]                                ← Print second component
[1] 25 26 18 39
> mylist[["ages"]]
[[1]] 25 26 18 39
```

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()`

Eg-2 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
```

Listing 2.6 Using factors

```
> patientID <- c(1, 2, 3, 4)                                1 Enter data as vectors
> age <- c(25, 34, 28, 52)
> diabetes <- c("Type1", "Type2", "Type1", "Type1")
> status <- c("Poor", "Improved", "Excellent", "Poor")
> diabetes <- factor(diabetes)
> status <- factor(status, order=TRUE)
> patientdata <- data.frame(patientID, age, diabetes, status)
> str(patientdata)
'data.frame':   4 obs. of  4 variables:
$ patientID: num  1 2 3 4
$ age       : num  25 34 28 52
$ diabetes : Factor w/ 2 levels "Type1","Type2": 1 2 1 1
$ status    : Ord.factor w/ 3 levels "Excellent"<"Improved"<...: 3 2 1 3
> summary(patientdata)
  patientID      age      diabetes      status
  Min.   :1.00   Min.   :25.00   Type1:3   Excellent:1
  1st Qu.:1.75  1st Qu.:27.25  Type2:1   Improved :1
  Median  :2.50  Median :31.00          Poor     :2
  Mean    :2.50  Mean   :34.75
  3rd Qu.:3.25  3rd Qu.:38.50
  Max.    :4.00  Max.   :52.00
```

← 1 Enter data as vectors

← 2 Display object structure

← 3 Display object summary

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()

Function	Purpose
<code>length(object)</code>	Number of elements/components.
<code>dim(object)</code>	Dimensions of an object.
<code>str(object)</code>	Structure of an object.
<code>class(object)</code>	Class or type of an object.
<code>mode(object)</code>	How an object is stored.
<code>names(object)</code>	Names of components in an object.
<code>c(object, object, ...)</code>	Combines objects into a vector.
<code>cbind(object, object, ...)</code>	Combines objects as columns.
<code>rbind(object, object, ...)</code>	Combines objects as rows.
<code>object</code>	Prints the object.
<code>head(object)</code>	Lists the first part of the object.
<code>tail(object)</code>	Lists the last part of the object.
<code>ls()</code>	Lists current objects.
<code>rm(object, object, ...)</code>	Deletes one or more objects. The statement <code>rm(list = ls())</code> will remove most objects from the working environment.
<code>newobject <- edit(object)</code>	Edits object and saves as newobject.
<code>fix(object)</code>	Edits in place.

条件筛选

条件筛选是先对变量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
```

Eg-3 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[value>40]`

Eg-4 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")

(3) R中的运算符

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`)

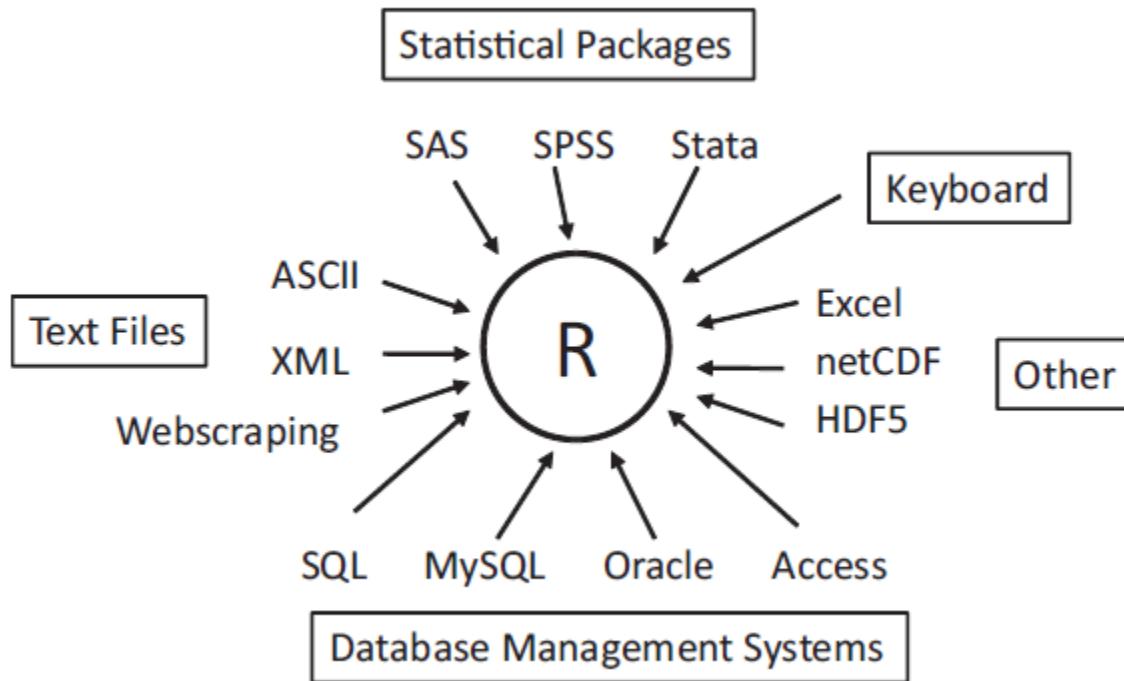
`>, <, <=, >=, ==, !=`

逻辑运算 与、或、非

`!, &, &&, |, ||`

❖ 4.5 R语言中的数据读取

- 第一种方式：从控制台读入
- 第二种方式：从外部数据读取



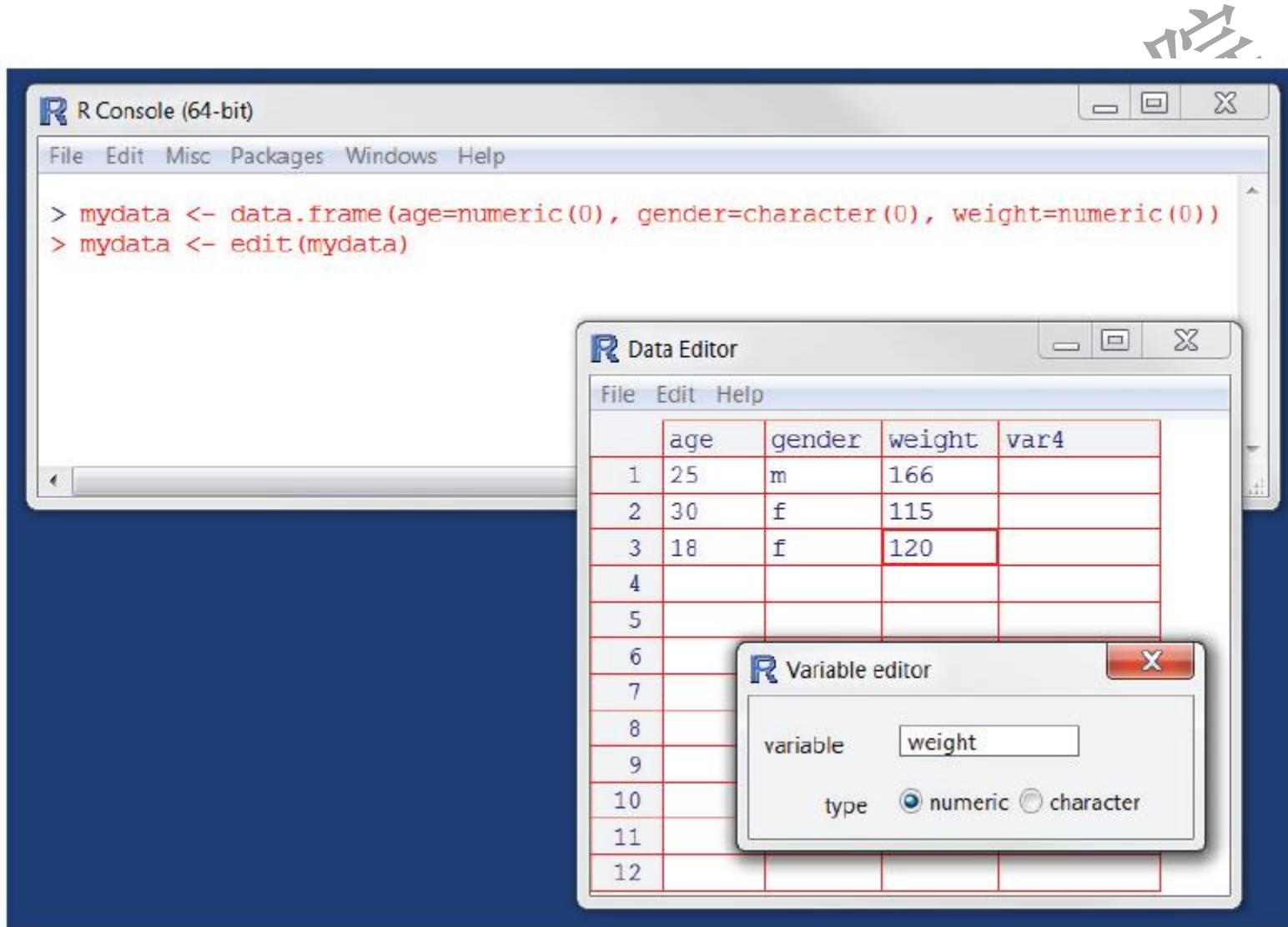
- 第三种方式：快捷的临时文件读取

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)
```

从键盘输入的另一种方式



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。

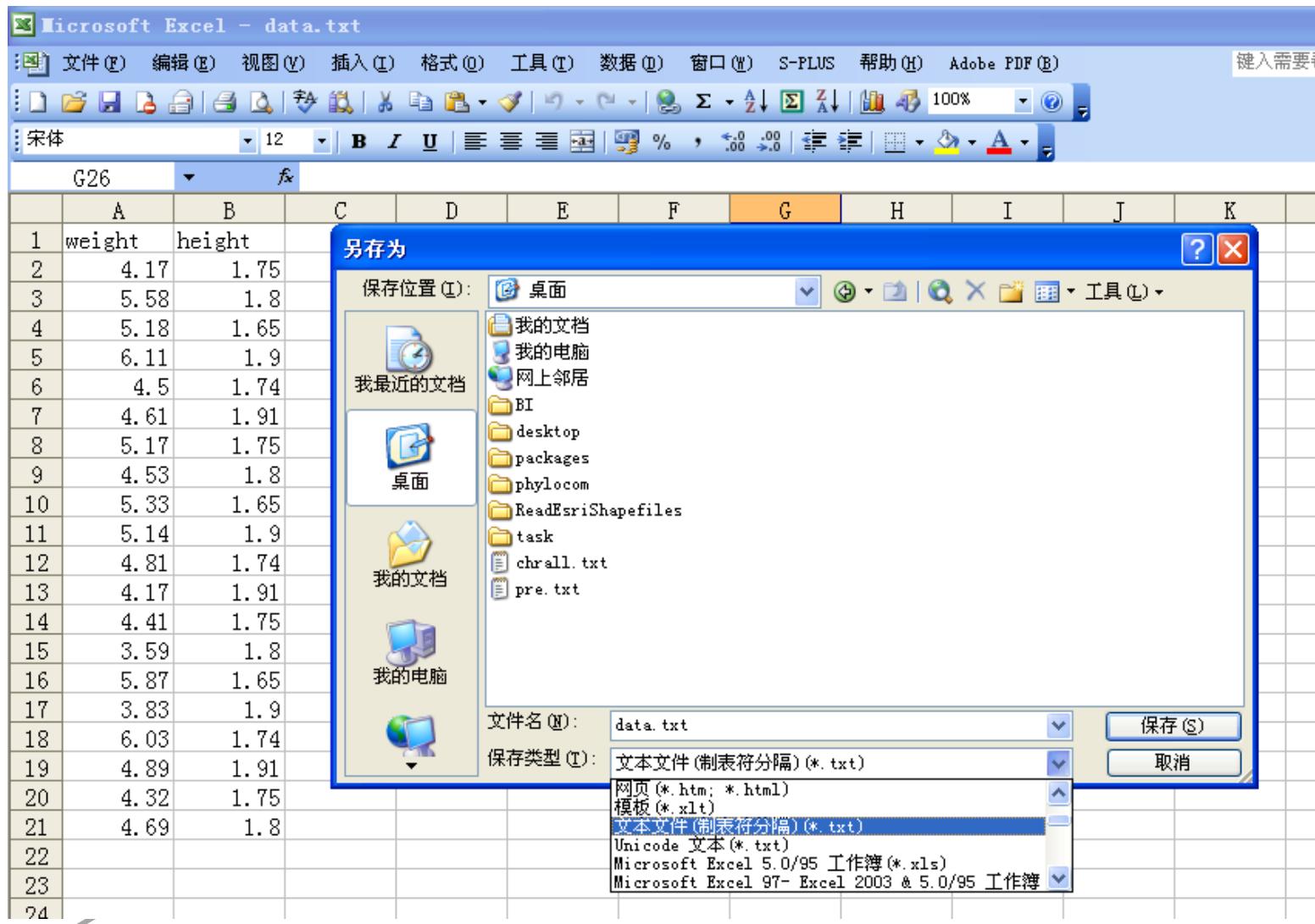


图8 在Excel中将数据存为txt文件

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。

Eg-5 从输入到假设检验

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

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$weight/data1$height^2
```

```
t.test(bmi, mu=22.5) #t检验
```

Eg-6 读取数据进行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)
```

Eg-7 编写函数从输入数据到单因素方差分析

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

Eg-8 编写函数读取数据进行方差分析

#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")
```

重慶

學院

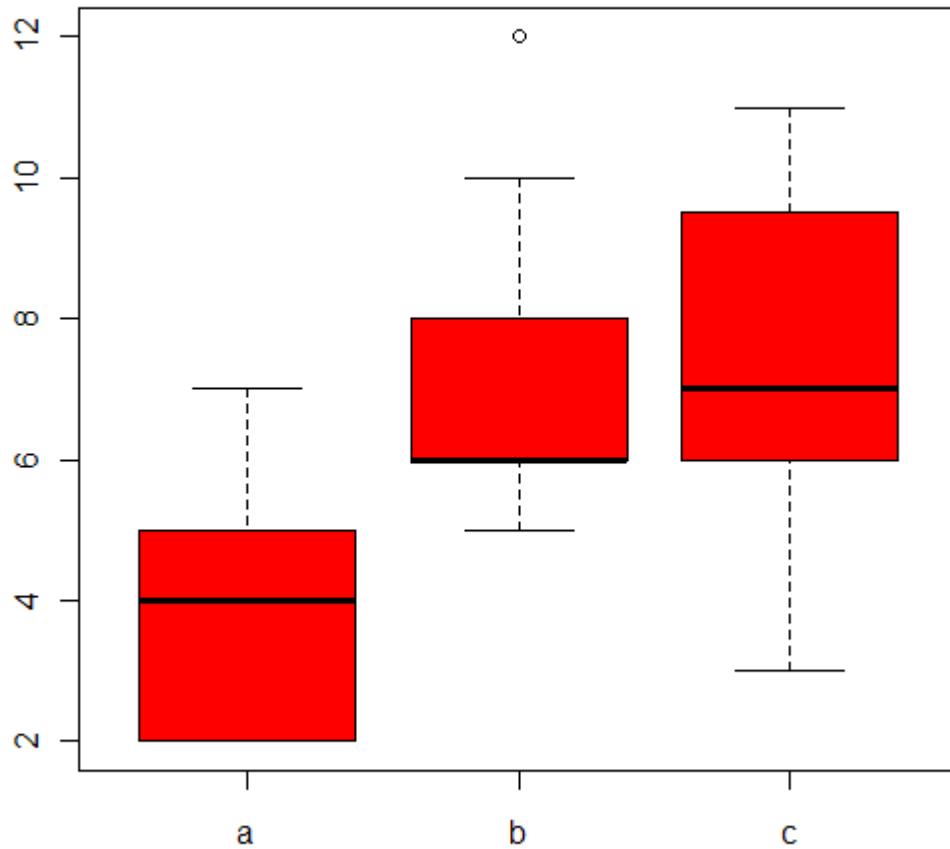


图10 三种菌型对小白鼠影响的箱线图

Eg-9 方差分析和箱线图绘图

- 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)
```

❖ 4.6 R语言的脚本编写

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。

编辑器与IDE

- 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
- **Rstudio** (推荐使用!)

The screenshot shows the TinnR IDE interface. The title bar reads "Tinn-R - [E:\dist\数据分析.r]". The menu bar includes File, Edit, Format, Projects, Search, Options, Tools, R, View, Window, Web, Help. The toolbar has various icons for file operations like Open, Save, Print, and navigation. The left sidebar shows a file tree under "Computer" tab, with "R and S files (*.r, *.q)" selected, showing files like "本地磁盘 (E:)" and "数据分析.r". The main workspace displays an R script with syntax highlighting:

```
cor<-read.table("E:/dist/phylocom results/comdist.txt",header=T)
d<-as.dist(cor)
phyhc<-hclust(d)
plot(phyhc, hang=0.1,cex=0.8,ylab="Phylogenetic distance",xlab="")
rect.hclust(phyhc, k=10, border="red")

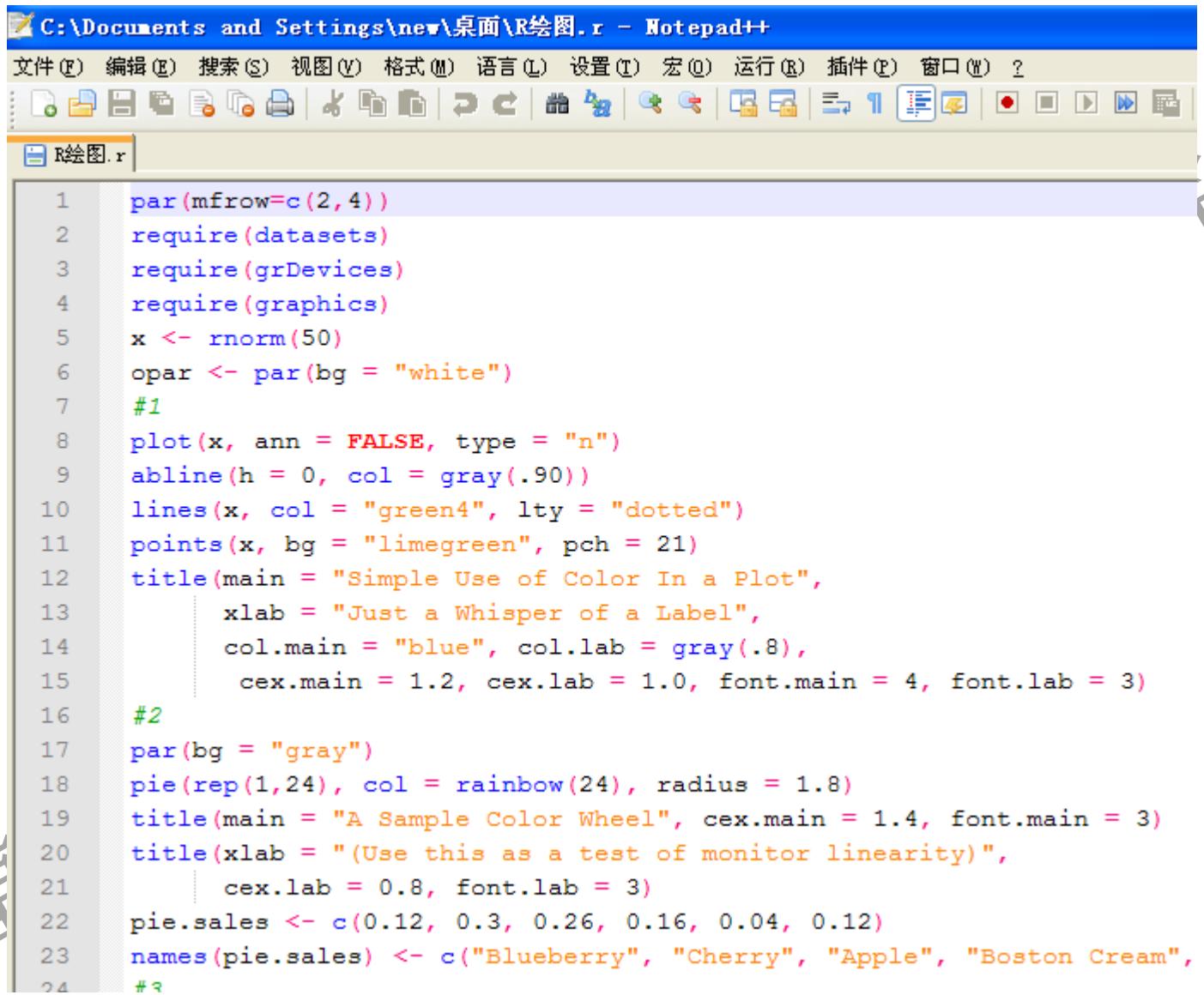
d2 <- cophenetic(phyhc)
cor(d,d2)

#地理距离计算, 在Condit教授矩阵位置是UTM坐标, 这里采用R软件 rgdal包转化为经纬度

library(rgdal)
UTM<-read.table("d:/utm.txt",header=T)
SP <- SpatialPoints(cbind(UTM$X, UTM$Y), proj4string=CRS ("+proj=utm +zone=17"))
results<-spTransform(SP, CRS ("+proj=longlat"))
write.table(results,"e:/dist/UTM.txt")
#用fossil包中的earth.dist()计算样地之间的距离
#
library(fossil)
library(ade4)

UTM<-read.table("e:/dist/UTM2.txt",header=T,na.strings = "NA")
geodistance<-earth.dist(UTM, dist = TRUE)
geodistance<-as.matrix(geodistance)
```

图11 TinnR对R脚本的高亮显示

A screenshot of the Notepad++ text editor. The title bar shows the path "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(M)", "语言(L)", "设置(I)", "宏(O)", "运行(R)", "插件(P)", "窗口(W)", and "?". The toolbar contains various icons for file operations like Open, Save, Print, and Find. The main window displays an R script named "R绘图.r". The code uses color-coded syntax highlighting: blue for numbers and strings, red for keywords like `par`, `require`, and `plot`, green for comments, and orange for symbols like `<-` and `#`. The script itself is a series of R commands for creating plots, including setting up a 2x4 grid, loading datasets, and plotting a scatter plot with specific colors and titles.

```
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 #?
```

图12 Notepad++对R脚本的高亮显示

The screenshot shows the RStudio interface with several panes:

- Code Editor (Left):** Displays the R script `test_O.R`. The code includes comments like `## Starting with R`, `## R Objects`, and variable assignments like `x <- 945`. Syntax highlighting is used throughout.
- Environment (Top Right):** Shows the global environment with objects like `news_table` (23 obs. of 2 variables), `IS_range` (int [1:99] 2 3 4 5 6 7 8 9 10 11 ...), and `tables1` (List of 4). It also displays a warning message: "Error(s) detected in the uploaded dataset".
- Console (Bottom Left):** Displays the output of the R script, including the creation of `news_table` and the result of `tables1[[4]]` which is `NULL`.

图13 RStudio对R脚本的高亮显示

Eg-10 回归分析脚本

- 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

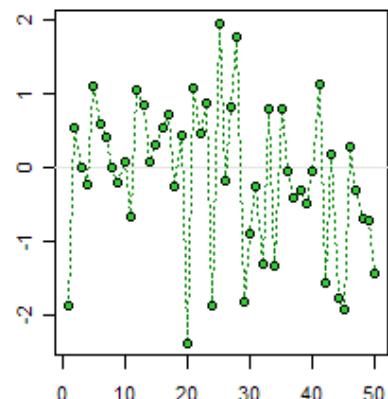
路径: RGui>File>Open Script #**Ctrl+R**运行

3 直接粘贴到R控制台

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

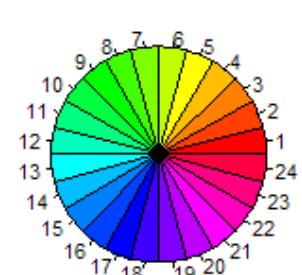
(5) R绘图

Simple Use of Color In a Plot



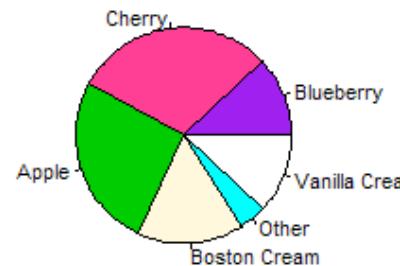
Just a Whisper of a Label

A Sample Color Wheel



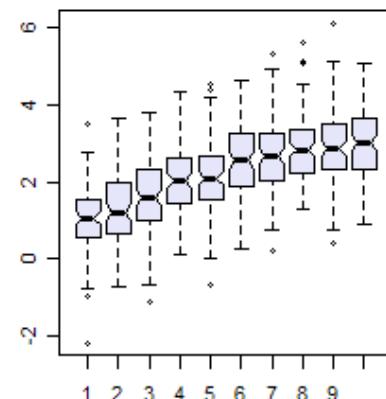
(Use this as a test of monitor linearity)

January Pie Sales



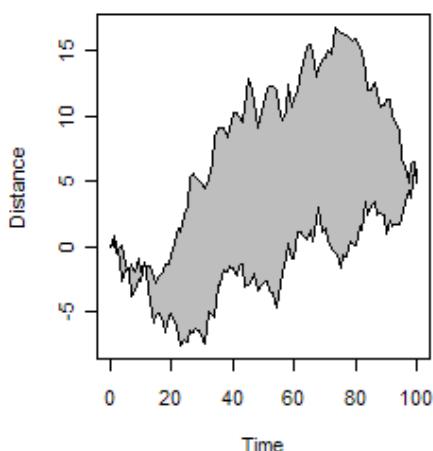
(Don't try this at home kids)

Notched Boxplots



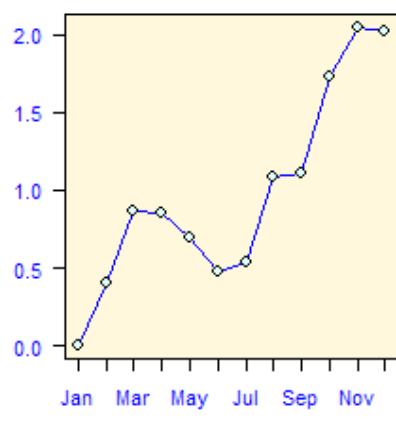
Group

Distance Between Brownian Motions



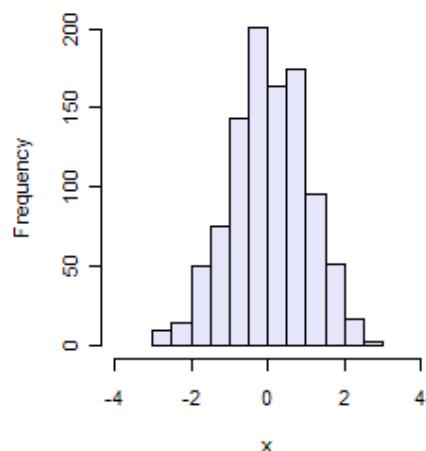
Time

The Level of Interest in R

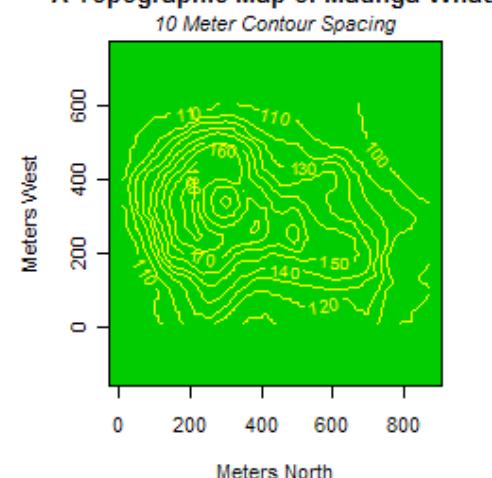


1996

1000 Normal Random Variates



A Topographic Map of Maunga Whau



Meters North

Meters West

图13 R绘制的图形

❖ 4.7 R的统计绘图功能

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

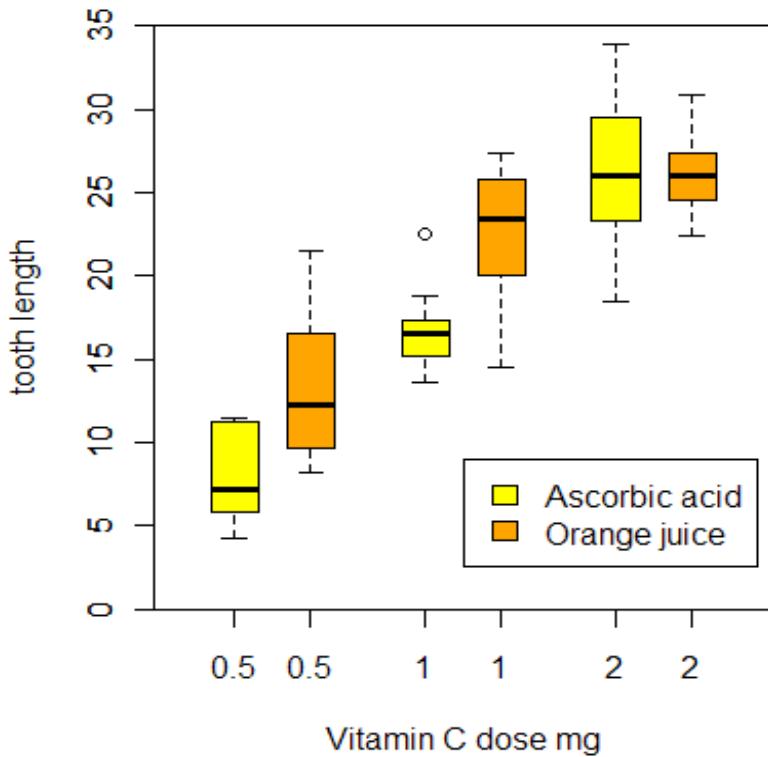
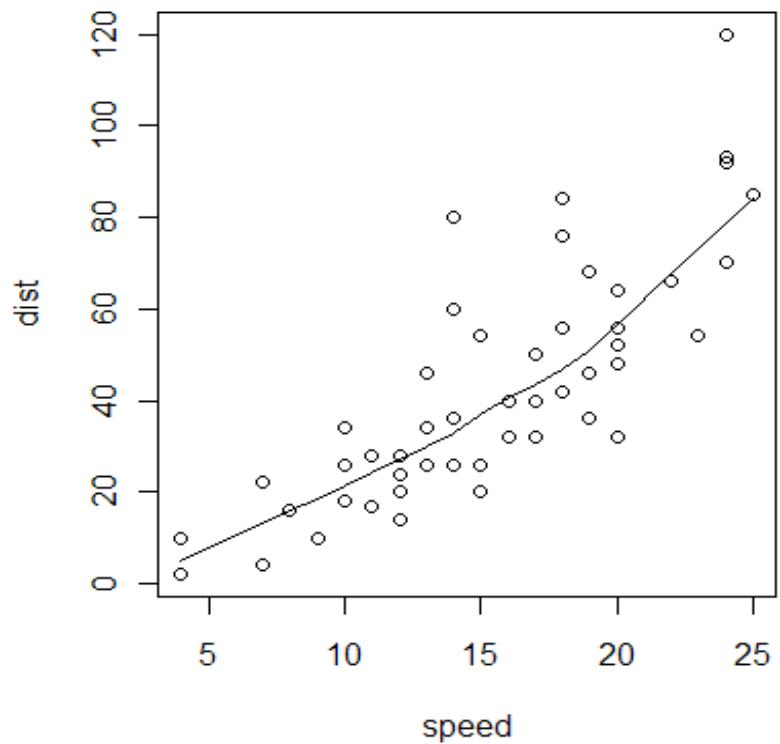
piechart() 饼图

interaction.plot()

matplot()

.....

Guinea Pigs' Tooth Growth



里

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

Eg-11 实例绘图

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)
```

重庆

学院

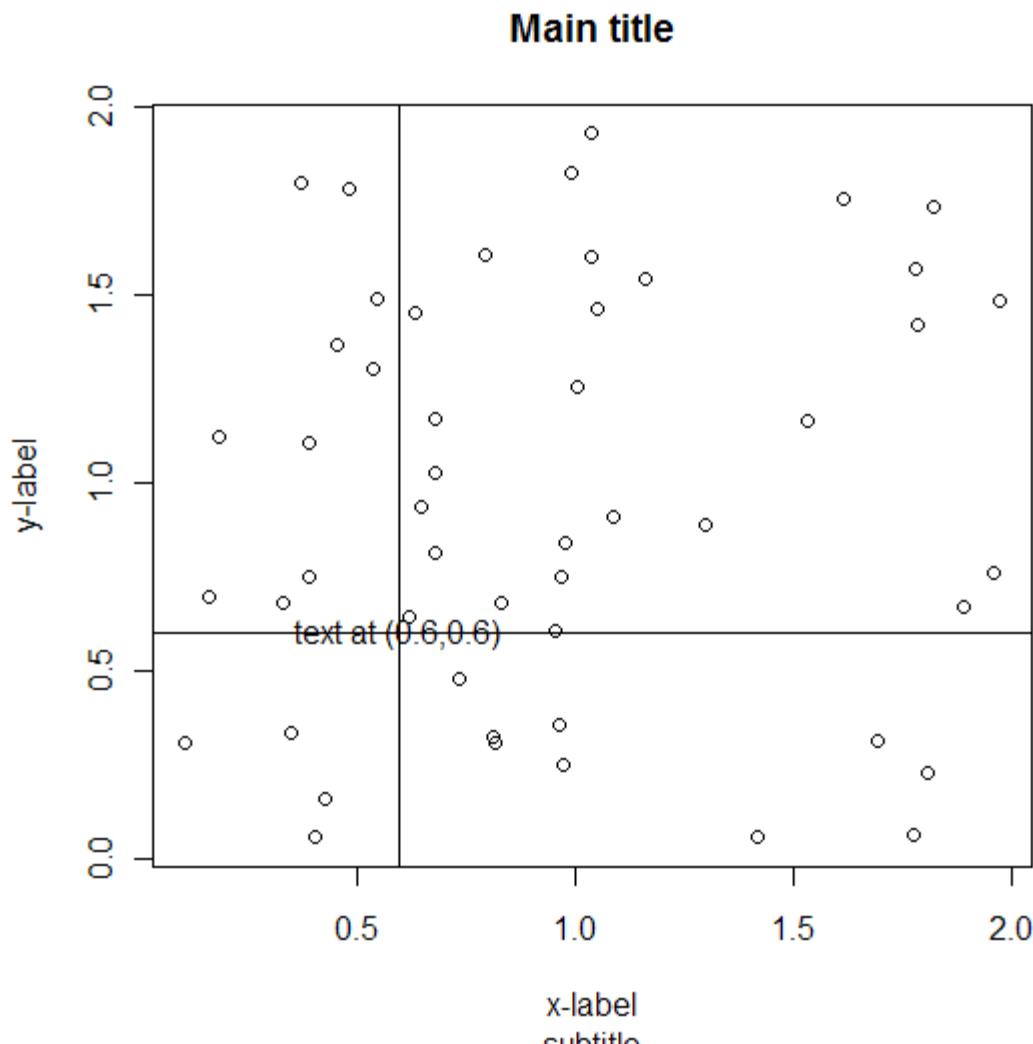


图15 绘图举例

Eg-12 分步骤绘图

```
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
```

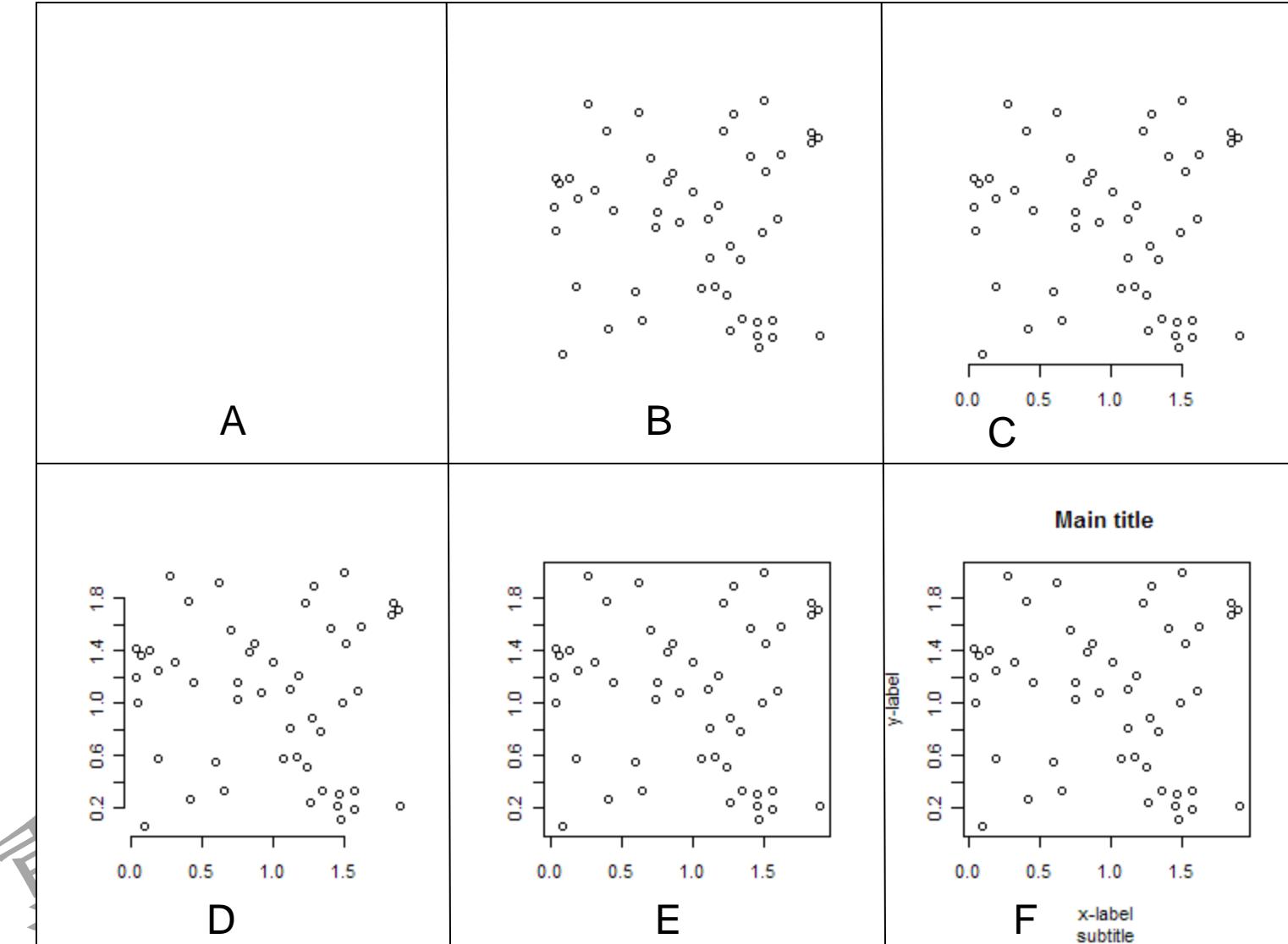


图16 分步绘图

一页多图

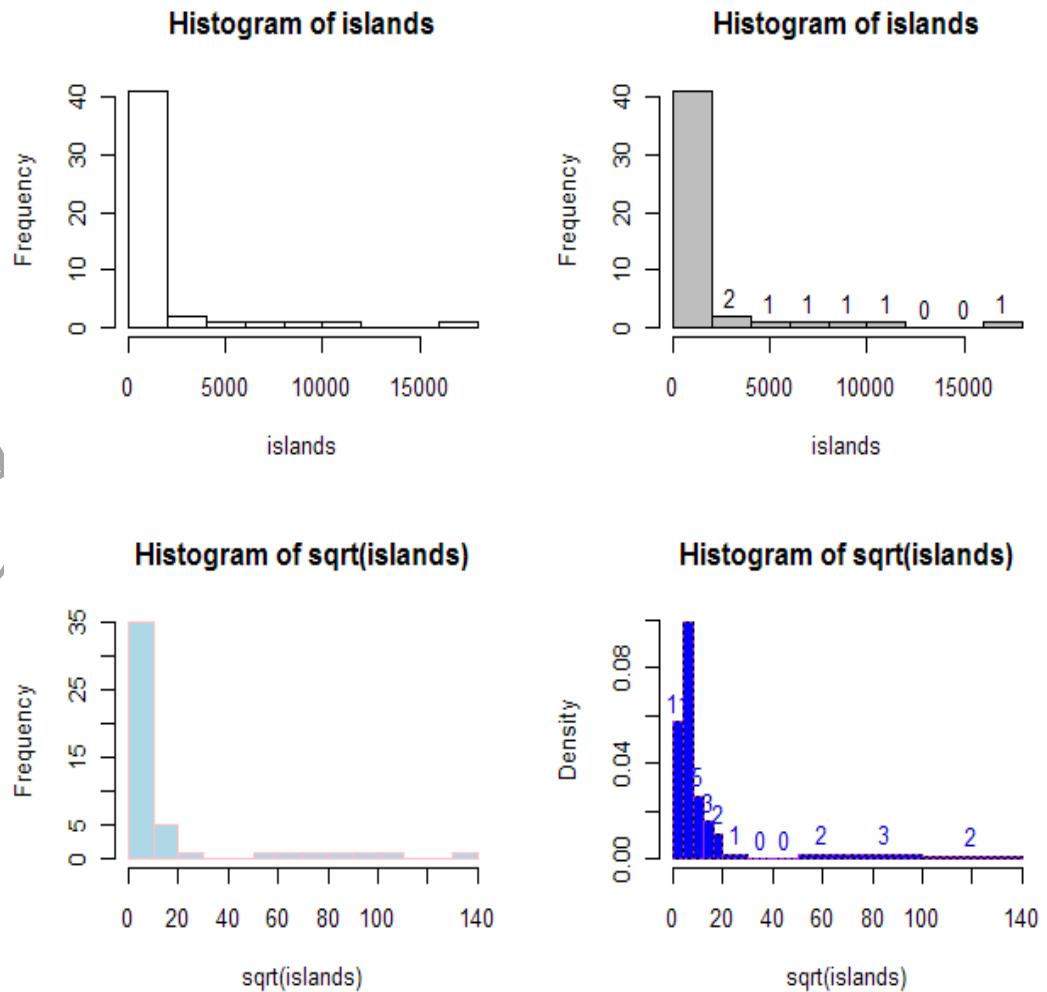


par()

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
```

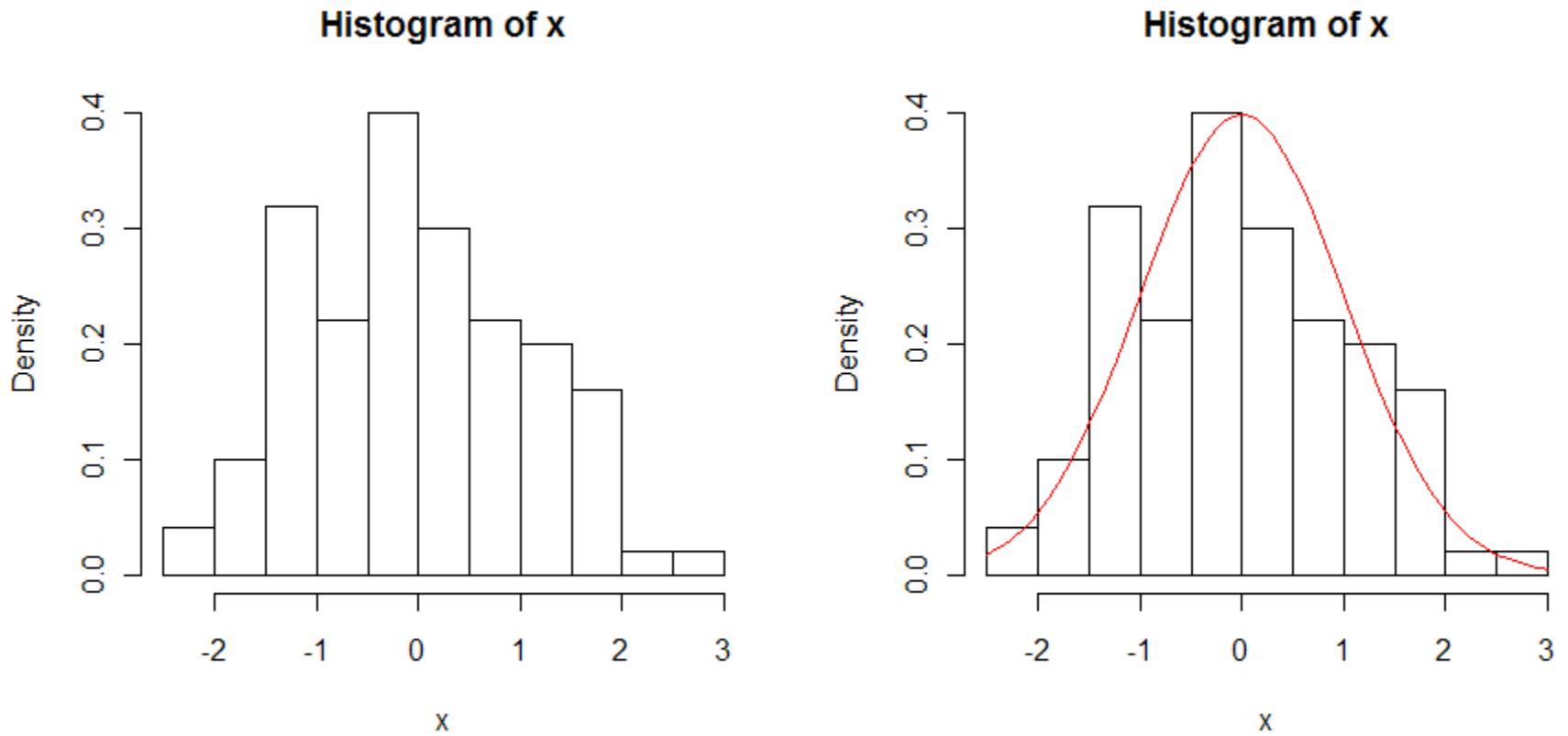


图18 在原有直方图上添加曲线

Eg-12 绘制饼图



Listng 6.5 Pie charts

```
par(mfrow=c(2, 2))
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")

pie(slices, labels = lbls,
     main="Simple Pie Chart")

pct <- round(slices/sum(slices)*100)
lbls2 <- paste(lbls, " ", pct, "%", sep="")
pie(slices, labels=lbls2, col=rainbow(length(lbls2)),
     main="Pie Chart with Percentages")

library(plotrix)
pie3D(slices, labels=lbls,explode=0.1,
      main="3D Pie Chart ")

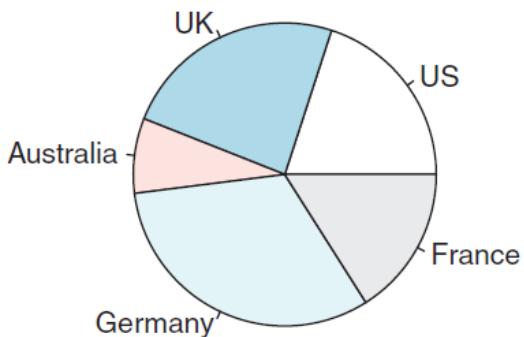
mytable <- table(state.region)
lbls3 <- paste(names(mytable), "\n", mytable, sep="")
pie(mytable, labels = lbls3,
     main="Pie Chart from a Table\n (with sample sizes)")
```

1 Combine four graphs into one

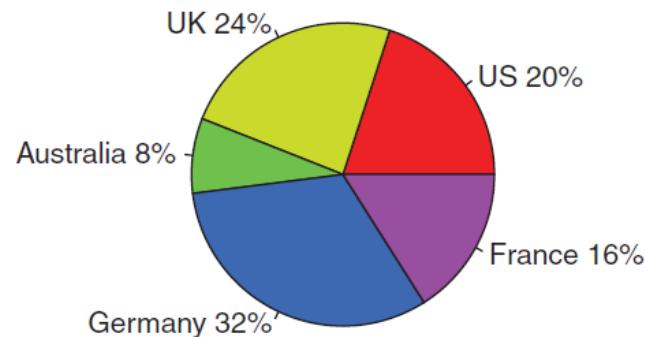
2 Add percentages to pie chart

3 Create chart from table

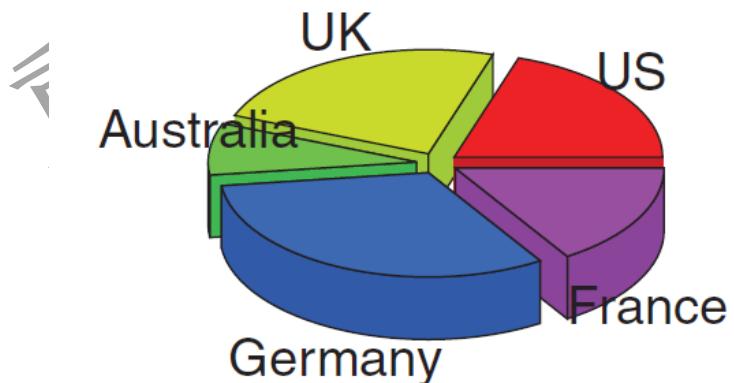
Simple Pie Chart



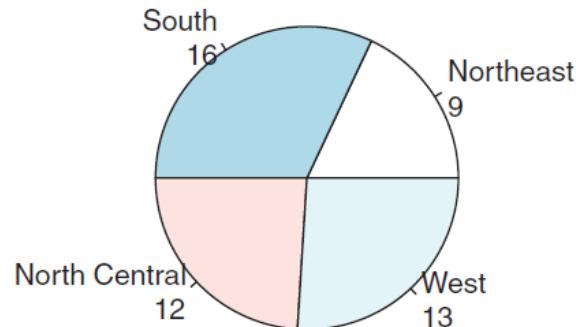
Pie Chart with Percentages



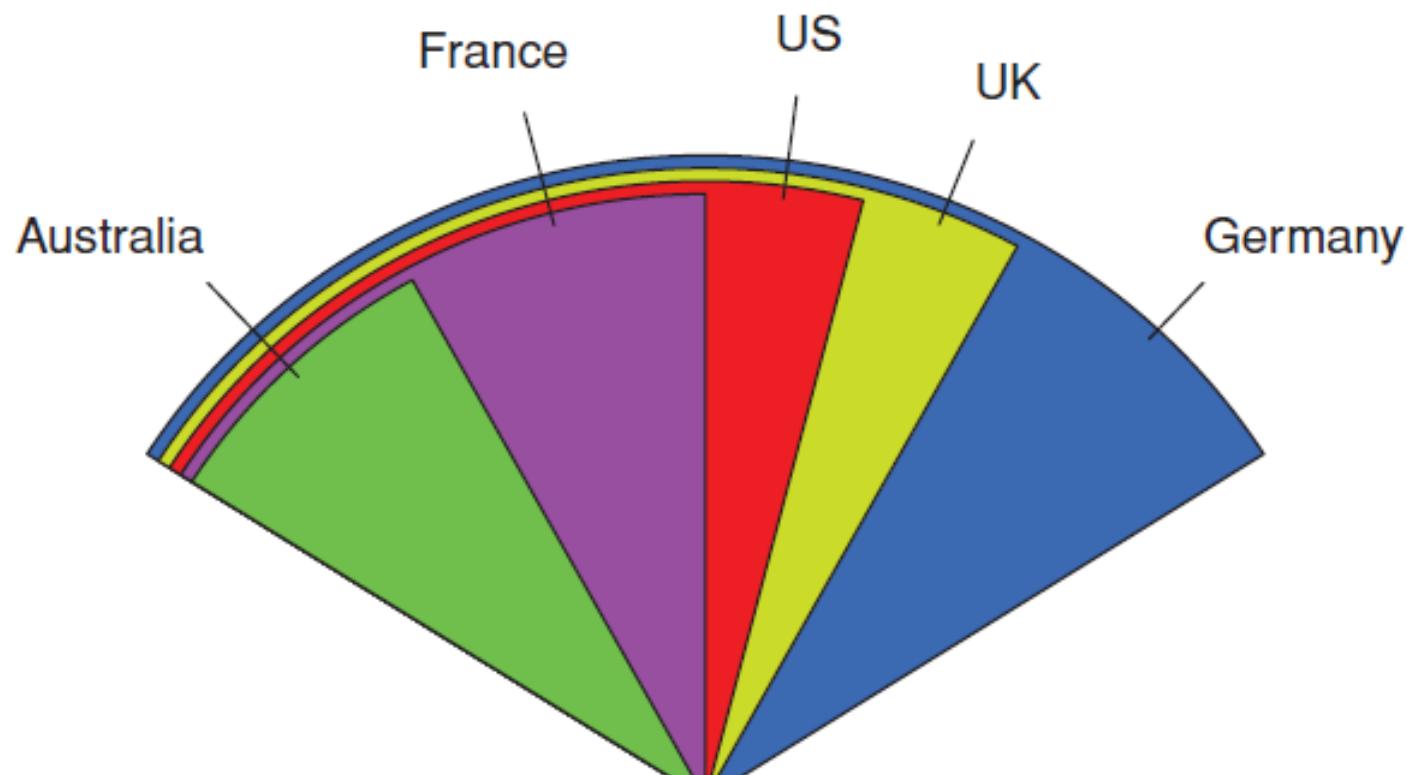
3D Pie Chart



Pie Chart from a Table (with sample sizes)



Fan Plot



```
library(plotrix)
slices <- c(10, 12, 4, 16, 8)
lbls <- c("US", "UK", "Australia", "Germany", "France")
fan.plot(slices, labels = lbls, main="Fan Plot")
```



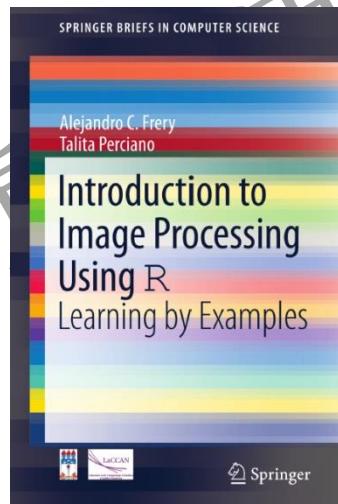
第5节：R语言应用实例---数据可视化、图像处理

Case-1: Data structure & Plotting

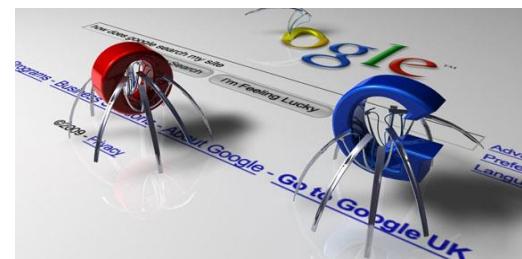
```
2  ### A function for drawing the map of the specific province.  
3  ### Bo Li, 2017-08-06.  
4  
5  if (!require(REmap)) {library(devtools); install_github('lchiffon/REmap')}
```

```
6  map_for_province <- function(x) {  
7    data <- data.frame(country = mapNames(x), value = 5 * sample(length(mapCList[[x]])) + 200)  
8    out <- remapC(data, maptype = x, color = 'skyblue')  
9    plot(out)  
10 }  
11  
12 map_for_province("chongqing")
```

Case-2: Image processing with R



Case-3: Web Crawlers via R



```
library(XML)  
u <- 'http://tech.163.com/special/00094IGJ/top1000.html'  
url <- htmlParse(u, encoding="GB2312")  
tables <- readHTMLTable(url)  
names(tables) # 查看该页面由几部分组成，并显示各个部分的名称  
raw <- tables[[6]] # 6代表查看第六部分的表格内容  
raw
```



第6节：本章小结—R语言、数据挖掘

- 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语言具有极其强大的统计计算能力！
- R是大数据分析的利器，熟练掌握R可使数据分析过程事半功倍。
- R天生具备强大的数据挖掘和分析能力，是数据挖掘的首选工具。



重庆师范大学
CHONG QING NORMAL UNIVERSITY

Thanks for your attention!

Acknowledgement

重庆师范

College of Life Sciences, Chongqing Normal University

2022, Chongqing of P. R. C