

# 统计图形和 R

2017 年 12 月

邹海翔

深圳市规划国土发展研究中心





# 大纲

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

① 统计图形

② 统计绘图工具

③ R 简介

④ R 绘图

⑤ R 图形混合嵌入



# 目录

P&LRC

统计图形和 R

3

## 1 统计图形

### 2 统计绘图工具

### 3 R 简介

### 4 R 绘图

### 5 R 图形混合嵌入

82



# 线图 (line chart)

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

统计绘图工具

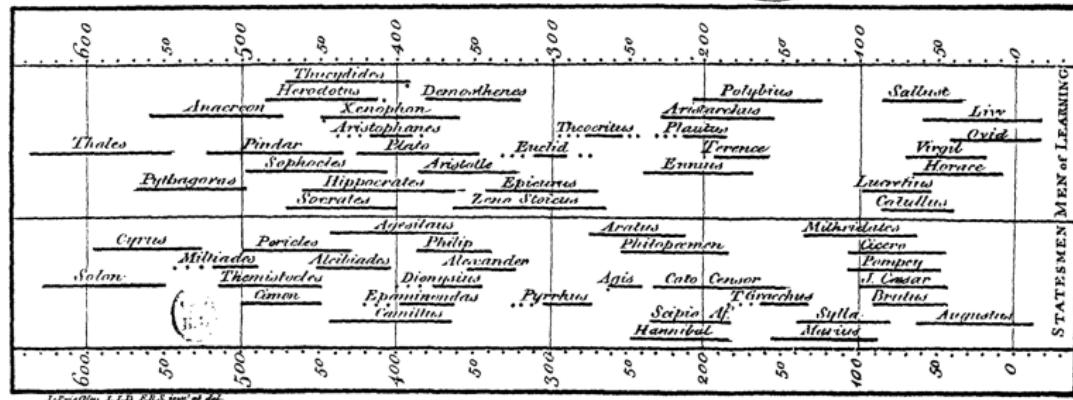
R 简介

R 绘图

R 图形混合嵌入

4

## *A Specimen of a Chart of Biography.*



图：英国化学家 Joseph Priestley 于 1765 年绘制的时间线图是历史上最早的统计图形，这幅图展示了多个历史人物在历史长河中的跨度

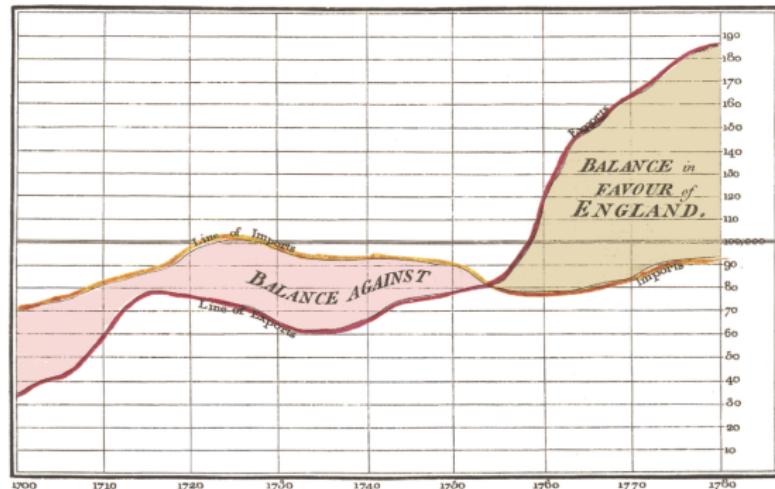


# 线图 (line chart)

统计图形和 R

4

Exports and Imports to and from DENMARK & NORWAY from 1700 to 1780.



The Bottom line is divided into Years, the Right hand line into £10,000 each.  
Published as the Act directs, 10<sup>th</sup> May 1786, by W<sup>m</sup> Playfair  
Made under S.R.C. demand, London.

图：苏格兰工程师和政治经济学家 William Playfair 于 1786 年在《Commercial and Political Atlas》一书中绘制的线图，这幅图展示了 1700 年至 1780 年间英格兰的进出口时序数据。Playfair 是历史上第一个系统使用统计图形的人，被称为“统计图形奠基人”

# 柱状图 (bar chart)

统计图形和 R

5

统计图形

线图

柱状图

饼图

主题统计地图

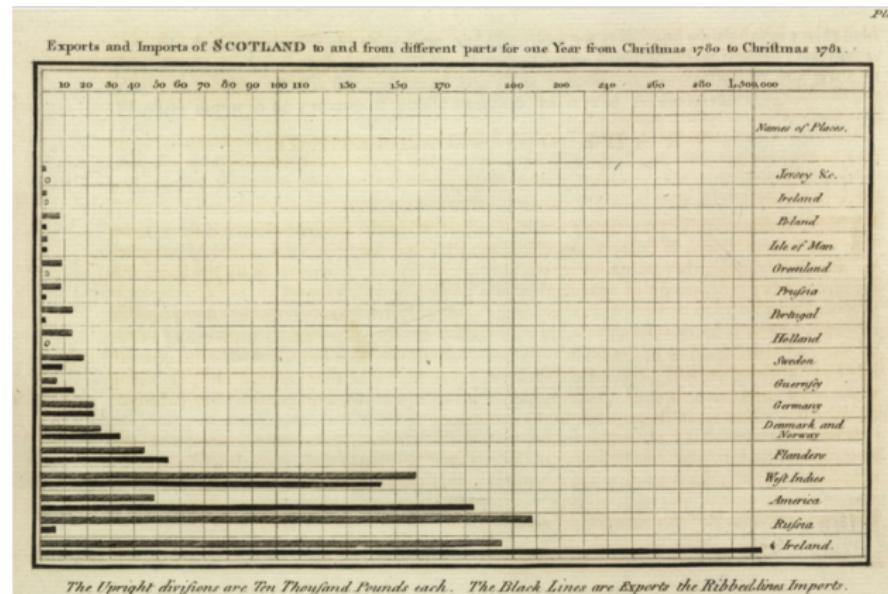
统计图形的应用

统计绘图工具

R 简介

R 绘图

R 图形混合嵌入



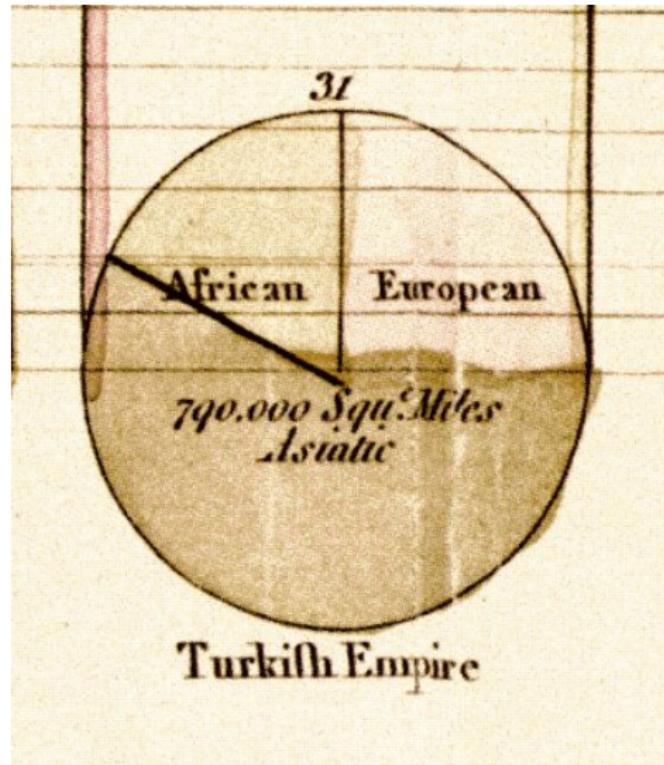
图：受到 Priestley 时间线图的影响，Playfair 于 1786 年同样在《Commercial and Political Atlas》这本书中绘制了历史上最早的柱状图，这幅图展示了不同国家的进出口数据



# 饼图 (pie chart)

统计图形和 R

6



图：Playfair 于 1801 年在《Statistical Breviary》这本书中绘制了历史上第一幅饼图，这幅图展示了土耳其帝国在三大洲的国土面积分布情况



# 主题统计地图

## 法国文盲率分布图

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

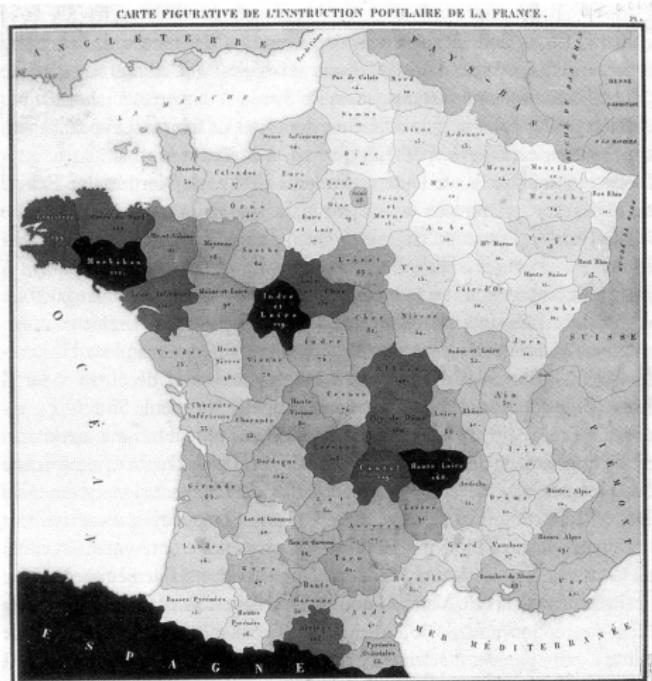
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

7



图：法国文盲分布图  
(Charles Dupin, 1826).  
Charles Dupin 是法国数学家，工程师，经济学家以及政客。1826 年，他首次运用区域灰度地图的表现手段来展示法国当时的文盲率分布情况，这是第一张现代形式的主题统计地图

82



# 主题统计地图

## 拿破仑东征图

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

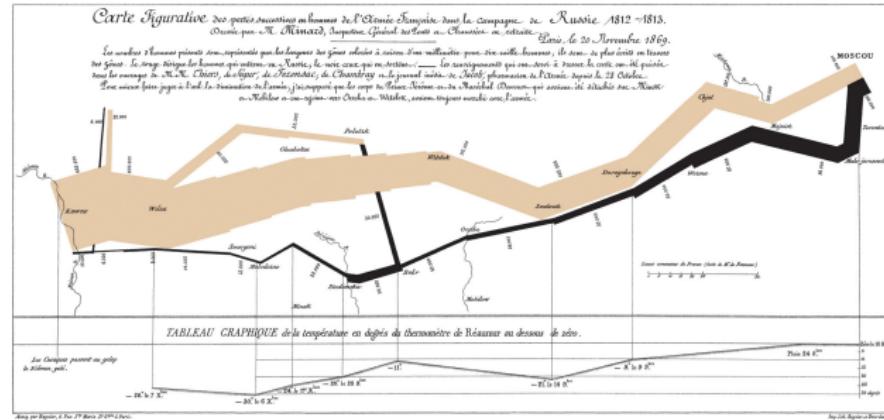
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

8



图：拿破仑东征图 (Charles Joseph Minard, 1861).

1861 年，由法国工程师 Charles Joseph Minard 绘制，描述了 1812 年拿破仑东征俄罗斯的失败战役。图中同时包含了多个信息，粗细代表军队规模，配合日期标明了军队位置经纬度，棕色进军黑色撤退，下方折线展现气温，另标注了战斗的关键事件等。

这幅图形在统计图形界内享有至高无上的地位，经常被一些统计、设计课程当作教学案例，被 Edward Tufte<sup>1</sup>誉为“有史以来最好的统计图形”

<sup>1</sup>Tufte 是统计图形和信息可视化领域的领军人物，人称“数据达芬奇”



# 主题统计地图

## 汉尼拔远征图

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

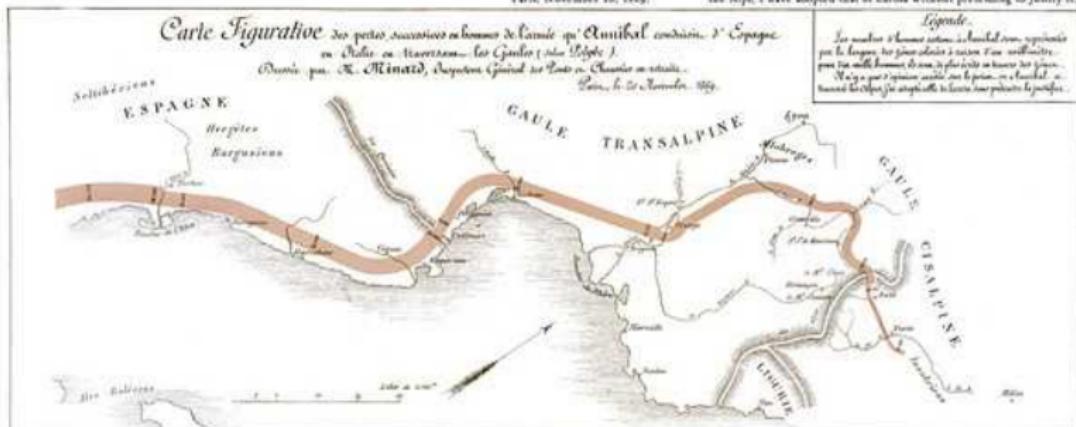
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

9



图：汉尼拔远征图 (Charles Joseph Minard, 1861)



# 主题统计地图

法国各地向巴黎输送牲畜产品的分布情况图

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

10

**CARTE FIGURATIVE** et approche des quantités de Viandes de Boucherie envoyées sur pied par les Départements et communautés à Paris.

Dessin par M<sup>e</sup> MINARD, Ingénieur Général des Ponts et Chaussées en retraite.

**Note explicative.**

La surface d'un cercle dans ce diagramme représente la quantité de viande de boucherie envoyée à Paris, celle des autres colis depuis l'expédition.

Un cercle de six millimètres de diamètre représente 100000 kilogrammes d'expédition à Paris, soit 222 millions de kilogrammes pour tout le pays.

Le diamètre du cercle est proportionnel à la quantité expédiée et l'angle auquel il est dévié de la verticale indique la direction de l'expédition à Paris.

Le diamètre du cercle indique la quantité expédiée à Paris.

Le département dont se trouve un cercle doit évidemment être dépeint sur le plan et il n'y a pas de cercles sur les îles, mais les îles sont dans le département tout entier.

On trouvera dans les notes des deux dernières pages une explication plus détaillée de l'application pratique de l'art de l'enseignement.

Paris, 1<sup>er</sup> Avril 1858. Charles Joseph MINARD.

图：法国各地向巴黎输送牲畜产品的分布情况图 (Charles Joseph Minard, 1858)

此图的作者也是 Joseph Minard, 经典之处在于首次将饼图融合到地图中

这位法国工程师，将一生大部分时间都贡献给了水坝，运河和桥梁的工程建造与教育事业，直到 1851 年退休，已近 70 岁高龄才正式开始研究数据信息图形可视化。在他最后的 20 年里，Minard 在这个领域贡献了许多创新，共绘制了 51 幅各种形式的可视化图形，是那个可视化黄金时代当之无愧的大师，被称为法国的“William Playfair”

82



# 主题统计地图

## 美国内战对欧洲棉花贸易的影响

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

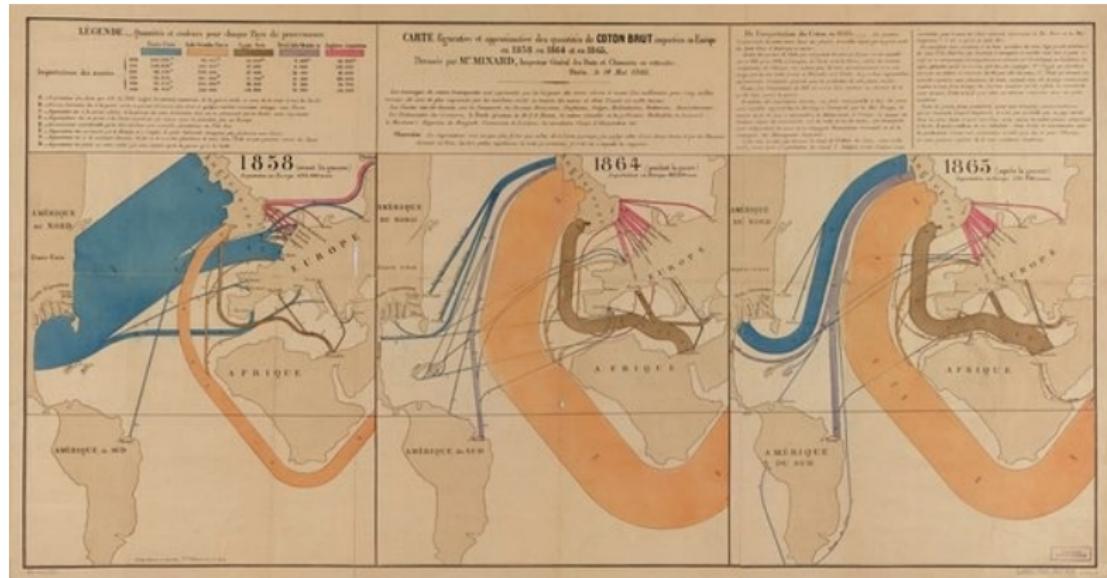
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

11



图：美国内战对欧洲棉花贸易的影响 (Charles Joseph Minard, 1865)

82

# 主题统计地图

## 法国红酒出口情况

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

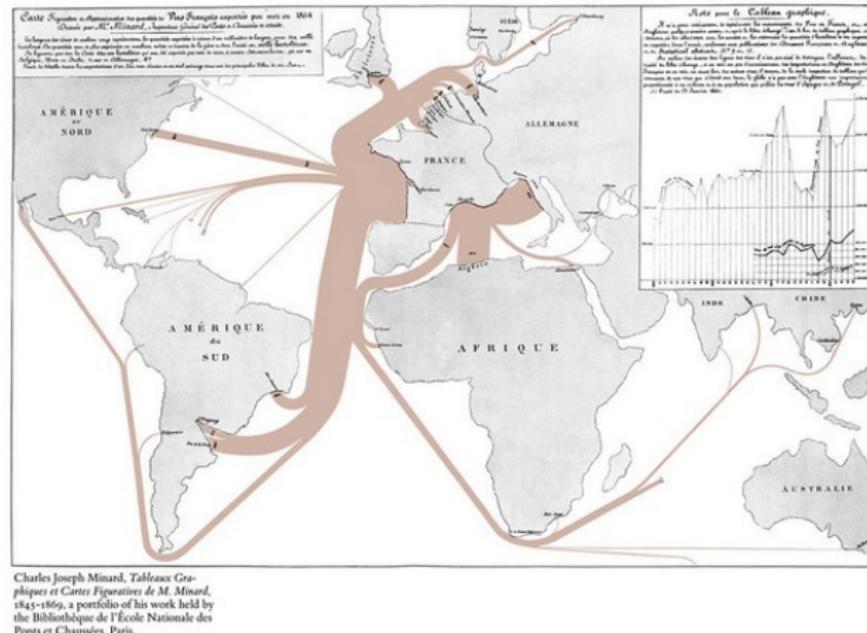
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

12



图：美国内战对欧洲棉花贸易的影响法国红酒出口情况 (Charles Joseph Minard, 1864)

82



# 主题统计地图

## 地铁路线图

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

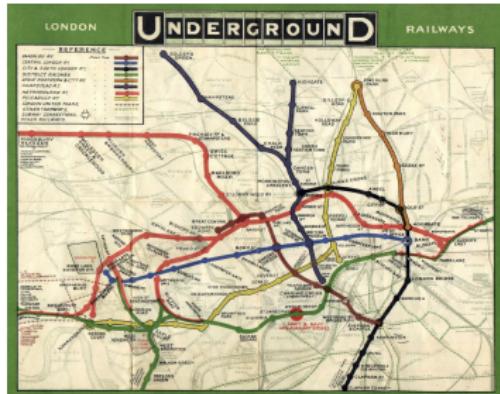
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

13



(a) 1908 年版



(b) 1933 年版

图：道路网络形状更新在最早的地图中都是按照实地比例进行绘制的，比如左图的1908版伦敦地铁线路图。1931年，英国技术制图员 Harry Beck 在替伦敦地铁的讯号室绘制电路图时受到启发，设计出了经典的1933年版路线图（右图）



# 主题统计地图

## 变形地图 (cartogram)

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

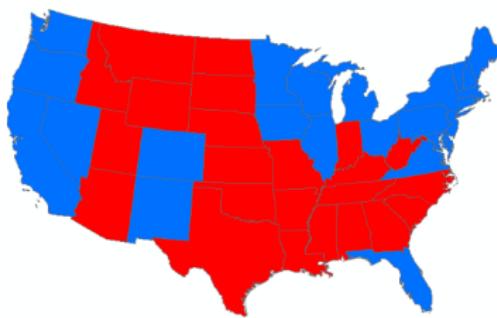
统计绘图工具

R 简介

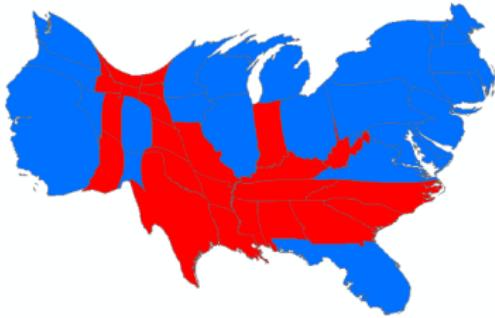
R 绘图

R 图形混合嵌入

14



(a) 普通色块地图表达



(b) 变形地图表达

图：变形地图的历史可以追溯到 1868 年，其作用是用夸张的地图变形来表达真实的数量关系

上面左图是 2012 年美国大选的结果，从普通色块图上看似乎是红色代表的罗姆尼获胜，但其实获胜的是蓝色代表的奥巴马；右图是根据选票数量进行变形处理后的地图，可以很清楚看到蓝色多于红色，这才是真实数量的正确表达！

82



# 统计图形的应用

## 南丁格尔玫瑰图

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

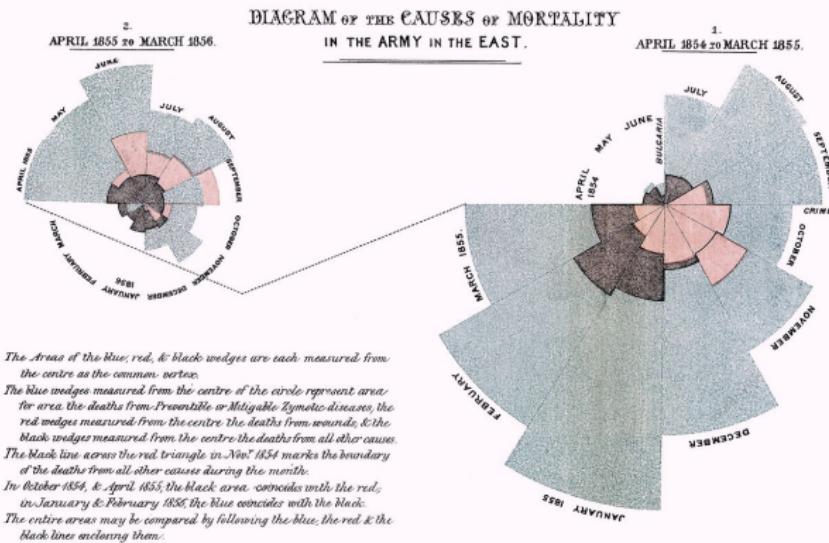
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

15



图：上图是南丁格尔玫瑰图。两幅图分别表示 1854 年和 1855 年军队的伤亡人数，一年 12 个月在极坐标上被分为 12 等分，每一个花瓣表示一个月；不同颜色表示死亡原因

南丁格尔通过这幅图使英国政府意识到真正影响战争伤亡的并非战争本身，而是由于军队缺乏有效的医疗护理。由此，英国政府于 1857 年开设了专门的军医学校，培养专门的战地医护人员，这就是统计图形在近代护理学最早的应用

82



# 统计图形的应用

## 霍乱传播之谜

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

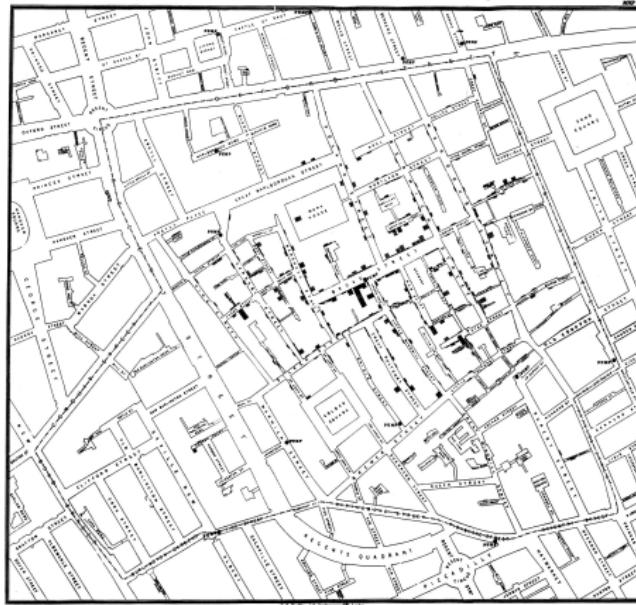
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

16



图：1854 年英国 Broad 大街大规模爆发霍乱，流行病学家 John Snow 对此次霍乱进行了大量调研分析，并且发表了霍乱传播理论的论文，左图是其论文的主要依据：图中心东西方向的街道是 Broad 大街，黑点表示死亡地点

这幅图形形象揭示了一个重要现象，就是死亡地点都在街道中部一处水源（水井）周围，而市内其它水源周围极少发现死者，通过进一步调查他发现这些死者都饮用过这里的井水，从而发现了霍乱传播的源头是水井的把手，**这就是统计图形在公共卫生领域最早的应用**

82



# 统计图形的应用

## 霍乱传播之谜

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

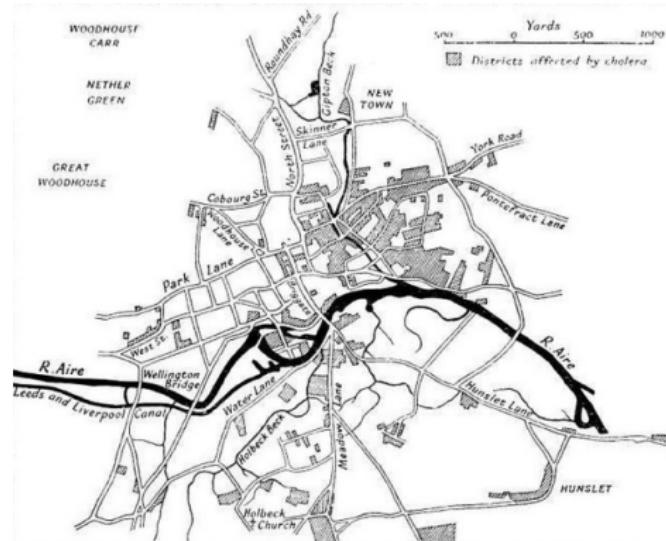
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

17



图：其实，在 John Snow 之前，有个叫 Robert Baker 的医生也研究了这个区域的霍乱问题，而且在 1833 年就绘制了左边这张霍乱的分布图。虽然 Baker 在这幅图中揭示了疾病和居住环境的联系：缺乏清洁用水和排水系统的居民点是疾病的高发区，**但是并没有显示发病率**。关于疾病起因的认知，他尽管方向正确但是并不完备，最终与伟大的发现擦肩而过。因此，只有充分完备的统计图形才能够真正应用于实践，这是一个漫长的科学过程。

82

# 统计图形的应用

## 切尔诺夫脸谱图 (Chernoff Faces)

统计图形和 R

统计图形

线图

柱状图

饼图

主题统计地图

统计图形的应用

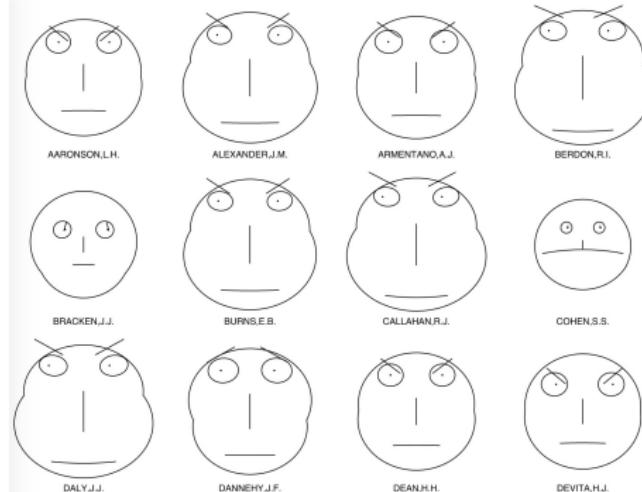
统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

18



图：切尔诺夫脸谱图 (Herman Chernoff, 1973)

这张很喜感的脸谱图其实是一种统计图，叫切尔诺夫脸谱图，是统计学家 Herman Chernoff 于 1973 年发明的，其基本思想是把多维数据的特征映射到卡通人脸中。由于人类非常善于识别脸部特征，脸谱化使得多维度数据容易被分析人员消化理解，有助于数据的规律和不规律性的可视化。目前这种方法已被广泛应用于多地域经济战略指标数据分析，空间数据可视化等领域。



# 目录

P&LRC

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

82

19

## 1 统计图形

## 2 统计绘图工具

## 3 R 简介

## 4 R 绘图

## 5 R 图形混合嵌入



# 为什么用绘图工具

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

20

R 简介

R 绘图

R 图形混合嵌入



謝避揮刀將大樹斜砍割斷。張翠山等三人看那大樹的斜剖面時，只見樹心中一條條通水的筋脈已大半震斷，有的扭曲，有的粉碎，有的裂為數截，有的若斷若續。

工欲善其事，必先利其器  
—《论语·卫灵公》

# 为什么用绘图工具

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

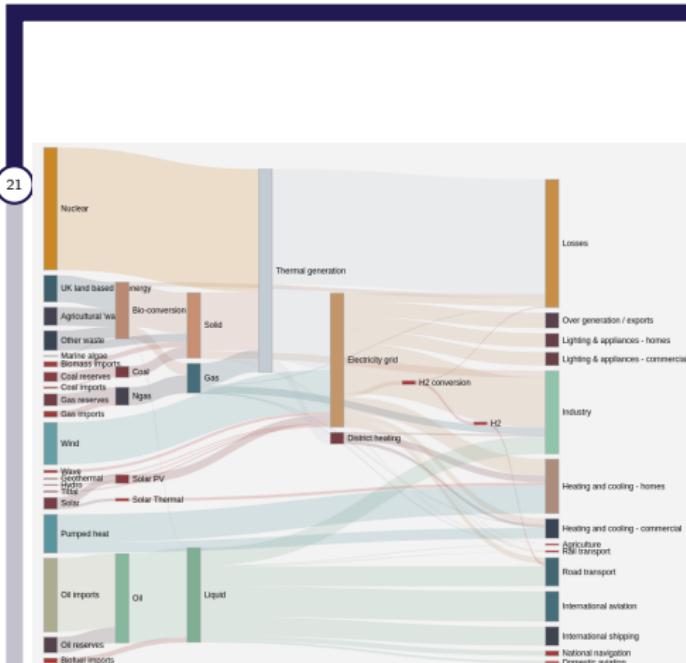
所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入



图：看上去复杂但是直观的桑基图 (Sankey diagram)

## 直观与简单

 **统计量是统计图形最关键的构成因素**，因此，优秀的统计图形背后必然隐藏着重要的统计量

 图形的首要作用是“**直观**”展示统计量信息，但是**能够直观理解的信息未必是“简单”的**

 使用合适的工具可以让信息的表达既“**直观**”又“**简单**”



# 何为利器

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

22

- 统计计算功能齐全
- 图形元素易于控制
- 统计图形种类丰富



图：常见的一些统计绘图工具

82

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

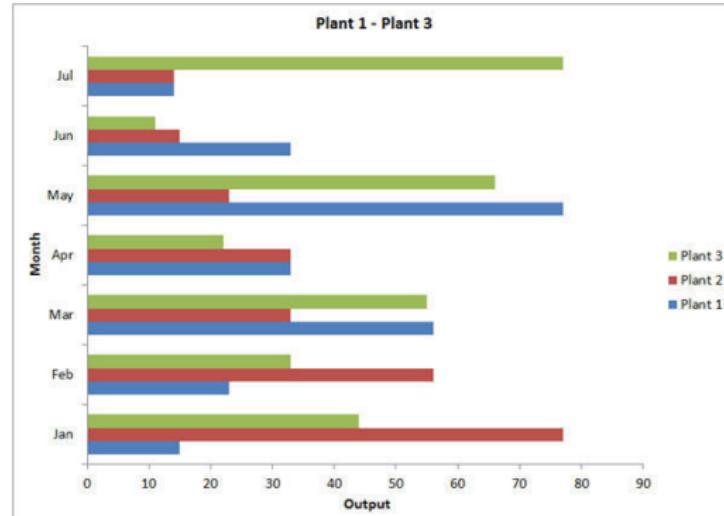
R 简介

R 绘图

R 图形混合嵌入

23

- 微软公司开发的电子数据表软件，从 1985 年发布 1.0 版本至今已经 30 多年历史，是电子数据表类软件的工业标准
- 优秀的人机实时交互设计，所见即所得
- 能够完成简单的统计功能以及数据绘图，但并不是统计软件



图：表示绝对数值大小的条形图

82



# 所见即所得工具

## excel

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

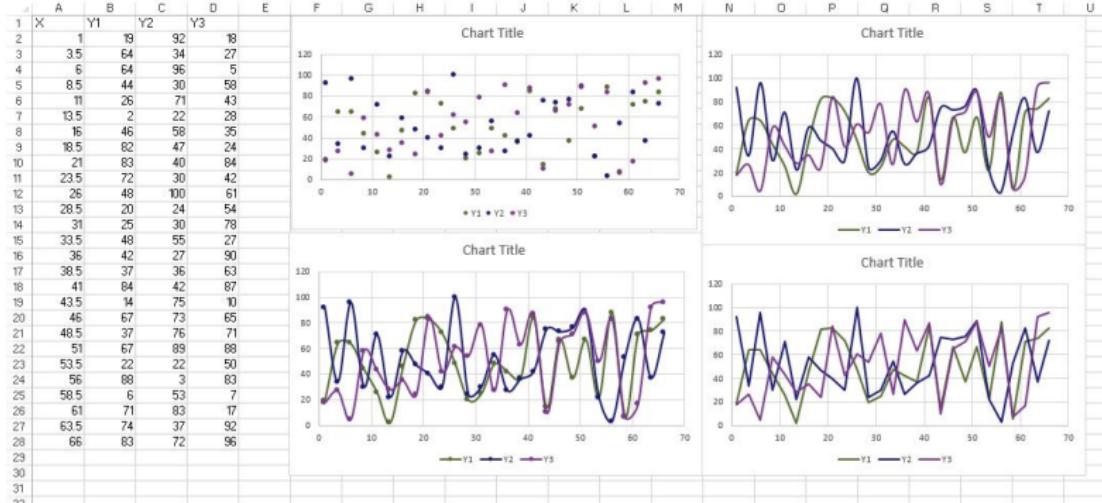
R 简介

R 绘图

R 图形混合嵌入

23

- 微软公司开发的电子数据表软件，从 1985 年发布 1.0 版本至今已经 30 多年历史，是电子数据表类软件的工业标准
- 优秀的人机实时交互设计，所见即所得
- 能够完成简单的统计功能以及数据绘图，但并不是统计软件



图：表示绝对数值大小的折线图

82



# 所见即所得工具

## excel

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

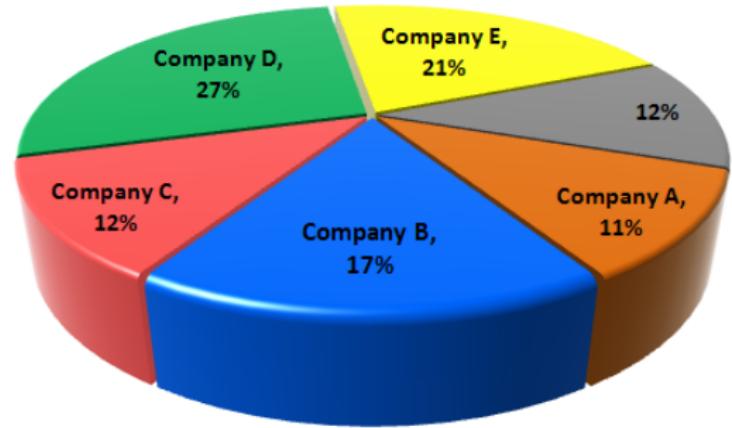
R 简介

R 绘图

R 图形混合嵌入

23

	A	B	C	D	E	F	G	H	I	J
1	The Pie Chart									
2										
3	2009	Part								
4	Company A	11%								
5	Company B	17%								
6	Company C	12%								
7	Company D	27%								
8	Company E	21%								
9	Other	12%								
10	Total	100%								
11										
12										
13										
14										
15										
16										
17										
18										
19										



图：表示比例大小的饼图

82



# 所见即所得工具

## excel

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

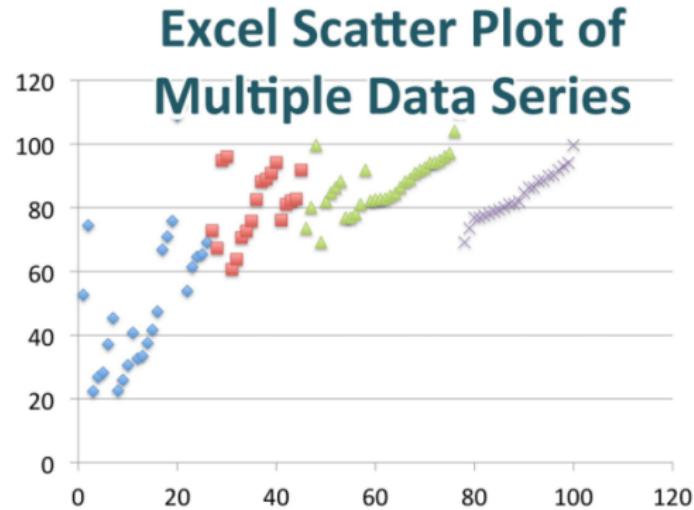
R 简介

R 绘图

R 图形混合嵌入

23

- 微软公司开发的电子数据表软件，从 1985 年发布 1.0 版本至今已经 30 多年历史，是电子数据表类软件的工业标准
- 优秀的人机实时交互设计，所见即所得
- 能够完成简单的统计功能以及数据绘图，但并不是统计软件



图：表示二维变量关系的散点图

82



# 所见即所得工具

## excel

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

23

- 微软公司开发的电子数据表软件，从 1985 年发布 1.0 版本至今已经 30 多年历史，是电子数据表类软件的工业标准
- 优秀的人机实时交互设计，所见即所得
- 能够完成简单的统计功能以及数据绘图，但并不是统计软件

## excel 作为统计绘图工具的缺点

- ⌚ excel 并非专业统计软件，只擅长对原始数据的展示，统计分析和建模功能十分有限。因此，著名统计学家 Leland Wilkinson<sup>1</sup> 说“**给统计刊物投稿时永远不要用 Excel 作图**”
- ⌚ 只能在 windows 单机上运行，处理数据量有限而且速度慢，不具备大规模数据管理功能
- ⌚ 绘图功能十分有限，只提供部分基本统计图形

82



# 所见即所得工具

## excel

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

23

- 微软公司开发的电子数据表软件，从 1985 年发布 1.0 版本至今已经 30 多年历史，是电子数据表类软件的工业标准
- 优秀的人机实时交互设计，所见即所得
- 能够完成简单的统计功能以及数据绘图，但并不是统计软件

## excel 作为统计绘图工具的缺点

- ⌚ excel 并非专业统计软件，只擅长对原始数据的展示，统计分析和建模功能十分有限。因此，著名统计学家 Leland Wilkinson<sup>1</sup> 说 “**给统计刊物投稿时永远不要用 Excel 作图**”
- ⌚ 只能在 windows 单机上运行，处理数据量有限而且速度慢，不具备大规模数据管理功能
- ⌚ 绘图功能十分有限，只提供部分基本统计图形

---

<sup>1</sup>统计学经典书籍《The Grammar of Graphics》一书的作者，任职 SPSS 公司十年，而且一直领导其可视化小组

82



# 所见即所得工具

## excel

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

23

- 微软公司开发的电子数据表软件，从 1985 年发布 1.0 版本至今已经 30 多年历史，是电子数据表类软件的工业标准
- 优秀的人机实时交互设计，所见即所得
- 能够完成简单的统计功能以及数据绘图，但并不是统计软件

## excel 作为统计绘图工具的缺点

- ⌚ excel 并非专业统计软件，只擅长对原始数据的展示，统计分析和建模功能十分有限。因此，著名统计学家 Leland Wilkinson<sup>1</sup> 说 “**给统计刊物投稿时永远不要用 Excel 作图**”
- ⌚ 只能在 windows 单机上运行，处理数据量有限而且速度慢，不具备大规模数据管理功能
- ⌚ 绘图功能十分有限，只提供部分基本统计图形

<sup>1</sup>统计学经典书籍《The Grammar of Graphics》一书的作者，任职 SPSS 公司十年，而且一直领导其可视化小组

82



统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

23

- 微软公司开发的电子数据表软件，从 1985 年发布 1.0 版本至今已经 30 多年历史，是电子数据表类软件的工业标准
- 优秀的人机实时交互设计，所见即所得
- 能够完成简单的统计功能以及数据绘图，但并不是统计软件

## excel 作为统计绘图工具的缺点

- ⌚ excel 并非专业统计软件，只擅长对原始数据的展示，统计分析和建模功能十分有限。因此，著名统计学家 Leland Wilkinson<sup>1</sup> 说 “**给统计刊物投稿时永远不要用 Excel 作图**”
- ⌚ 只能在 windows 单机上运行，处理数据量有限而且速度慢，不具备大规模数据管理功能
- ⌚ 绘图功能十分有限，只提供部分基本统计图形

---

<sup>1</sup>统计学经典书籍《The Grammar of Graphics》一书的作者，任职 SPSS 公司十年，而且一直领导其可视化小组



# 所见即所得工具

## SPSS

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

24

- 易学易用
- 功能全面
- 界面友好

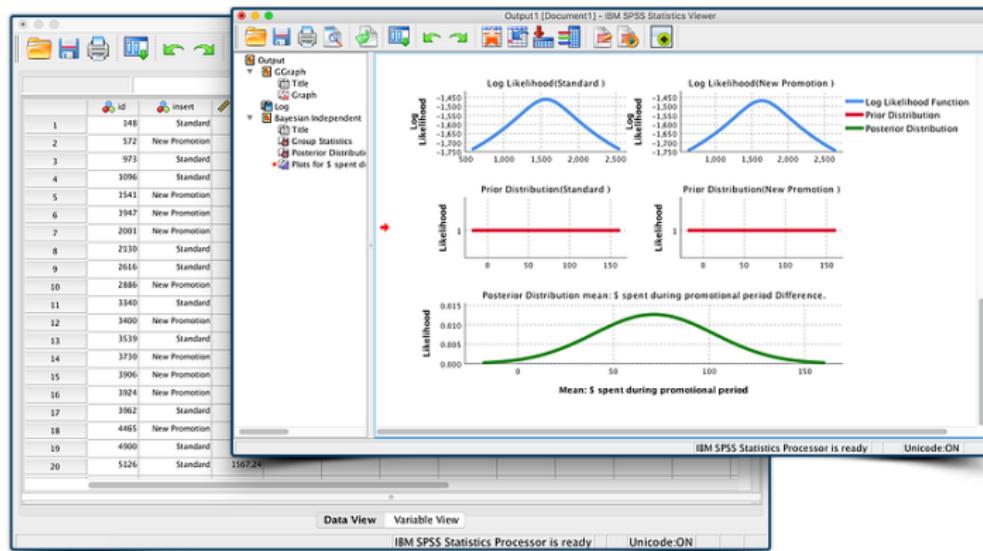


图: SPSS 用户操作界面

82



# 所见即所得工具

## SPSS

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

24

- 易学易用
- 功能全面
- 界面友好

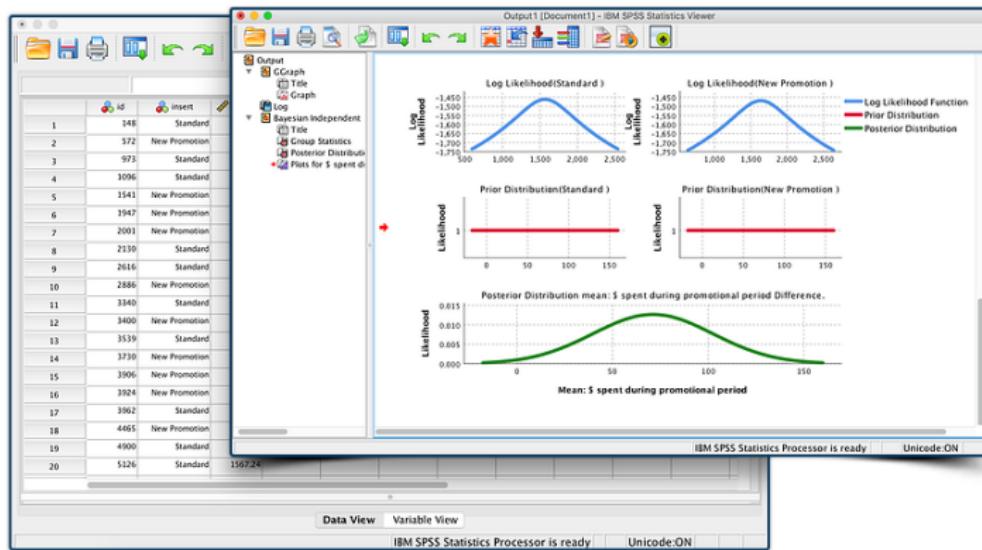


图: SPSS 用户操作界面

82



# 所见即所得工具

## SPSS

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

24

- 易学易用
- 功能全面
- 界面友好

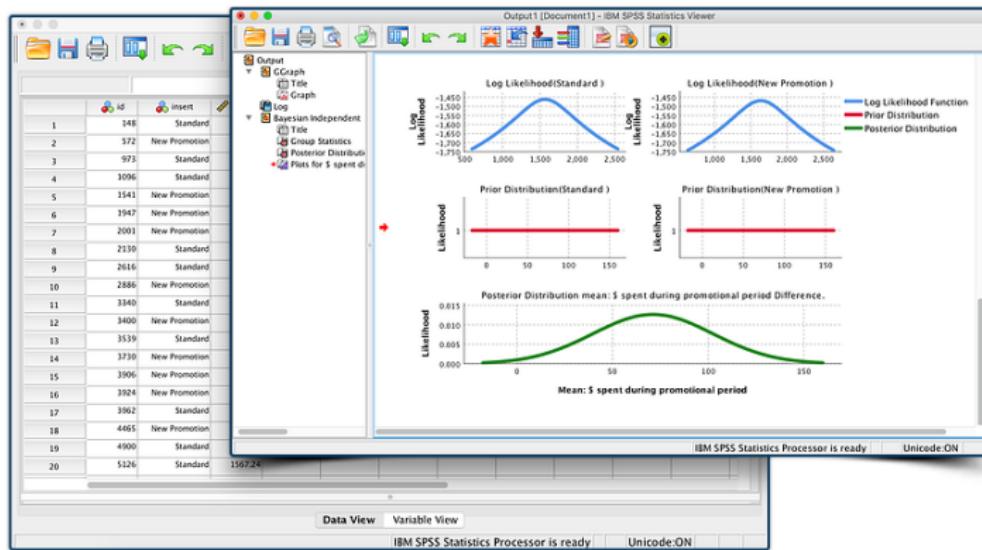


图: SPSS 用户操作界面

82



统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

24

- 易学易用
- 功能全面
- 界面友好

## SPSS 及其他所见即所得工具的缺陷

- ⌚ 按钮的数量总是有限的，而统计模型是无限的
- ⌚ 计算机完成了太多图形要素控制工作，缺乏灵活性
- ⌚ 生成一张图易，生成 N 张图难
- ⌚ 贫穷限制了想象力：商业软件费用高昂，而且部分模块需单独付费

82



# 所见即所得工具

## SPSS

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

24

- 易学易用
- 功能全面
- 界面友好

## SPSS 及其他所见即所得工具的缺陷

- ⌚ 按钮的数量总是有限的，而统计模型是无限的
- ⌚ 计算机完成了太多图形要素控制工作，缺乏灵活性
- ⌚ 生成一张图易，生成 N 张图难
- ⌚ 贫穷限制了想象力：商业软件费用高昂，而且部分模块需单独付费

82



# 所见即所得工具

## SPSS

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

24

- 易学易用
- 功能全面
- 界面友好

## SPSS 及其他所见即所得工具的缺陷

- ⌚ 按钮的数量总是有限的，而统计模型是无限的
- ⌚ 计算机完成了太多图形要素控制工作，缺乏灵活性
- ⌚ 生成一张图易，生成 N 张图难
- ⌚ 贫穷限制了想象力：商业软件费用高昂，而且部分模块需单独付费

82



统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

24

- 易学易用
- 功能全面
- 界面友好

## SPSS 及其他所见即所得工具的缺陷

- ⌚ 按钮的数量总是有限的，而统计模型是无限的
- ⌚ 计算机完成了太多图形要素控制工作，缺乏灵活性
- ⌚ 生成一张图易，生成 N 张图难
- ⌚ 贫穷限制了想象力：商业软件费用高昂，而且部分模块需单独付费

82



# 所想即所得工具 SAS

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

25

- 1960 年诞生于 SAS 软件研究所，老牌专业统计软件
- 基于数据库进行数据管理，具有大型数据集处理分析能力
- 有专门的 SAS 认证考试，包括程序员、业务分析师、数据挖掘、系统开发专家和系统管理专家五种不同角色

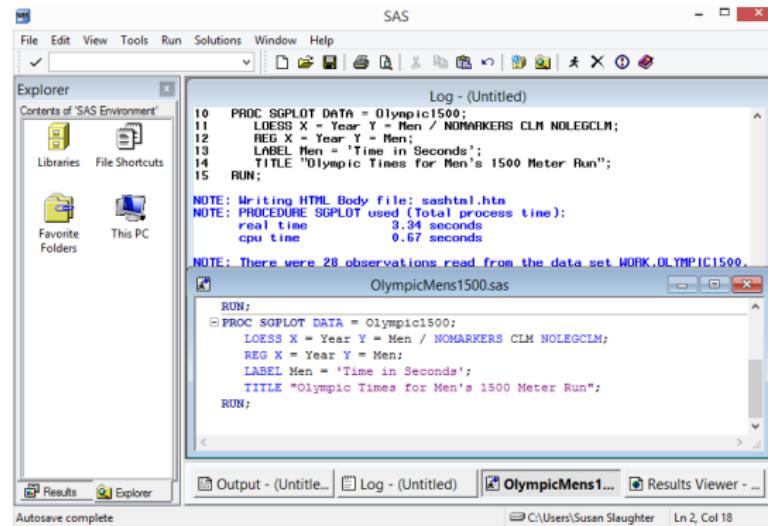


图: SAS 软件界面

82



# 所想即所得工具 SAS

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

25

- 1960 年诞生于 SAS 软件研究所，老牌专业统计软件
- 基于数据库进行数据管理，具有大型数据集处理分析能力
- 有专门的 SAS 认证考试，包括程序员、业务分析师、数据挖掘、系统开发专家和系统管理专家五种不同角色

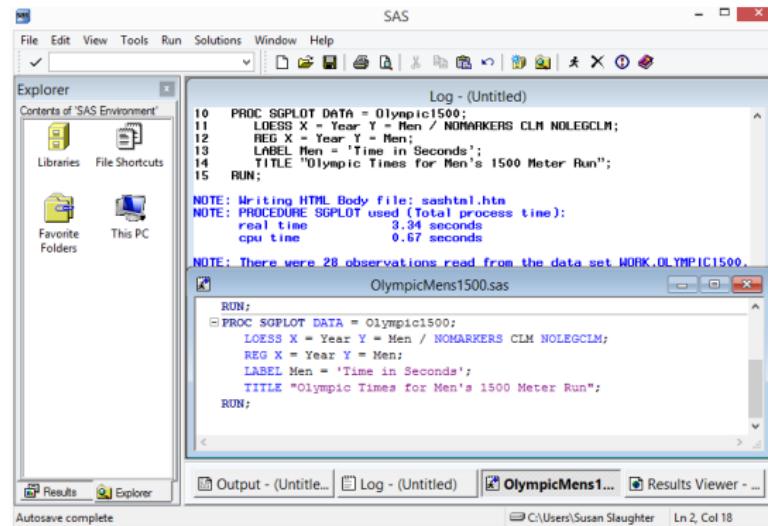


图: SAS 软件界面

82



# 所想即所得工具 SAS

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

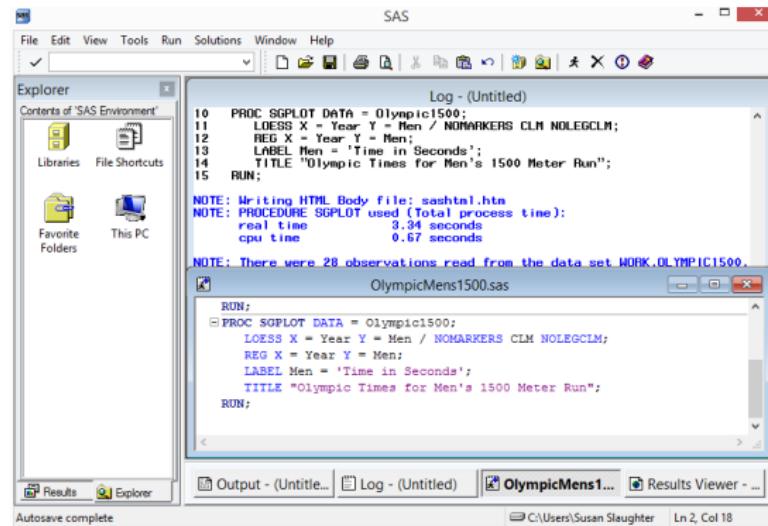
何为利器

所见即所得工具

所想即所得工具

25

- 1960 年诞生于 SAS 软件研究所，老牌专业统计软件
- 基于数据库进行数据管理，具有大型数据集处理分析能力
- 有专门的 SAS 认证考试，包括程序员、业务分析师、数据挖掘、系统开发专家和系统管理专家五种不同角色



图：SAS 软件界面

82



# 所想即所得工具 SAS

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

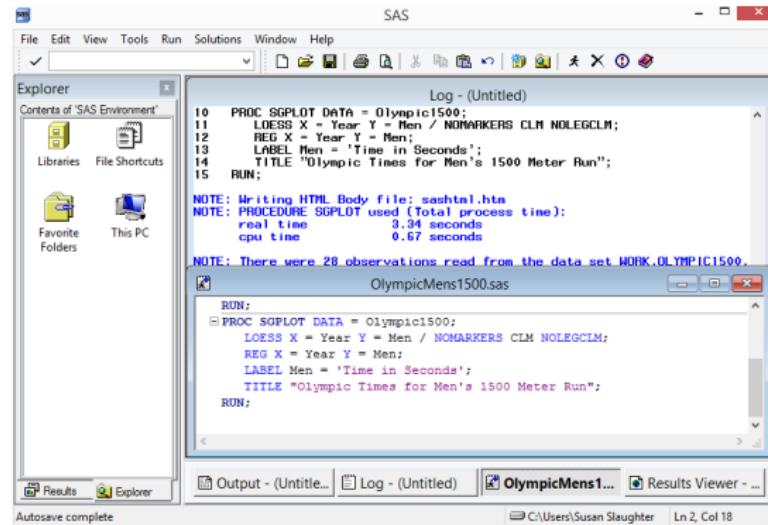
何为利器

所见即所得工具

所想即所得工具

25

- 1960 年诞生于 SAS 软件研究所，老牌专业统计软件
- 基于数据库进行数据管理，具有大型数据集处理分析能力
- 有专门的 SAS 认证考试，包括程序员、业务分析师、数据挖掘、系统开发专家和系统管理专家五种不同角色



图：SAS 软件界面

82

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

26



图：箱形图 (Box plot)

```

1 PROC SQL;
2 create table CARS1 as
3 SELECT make,model,type,invoice,
      horsepower,length,weight
4 FROM SASHELP.CARS
5 WHERE make in ('Audi','BMW');
6 RUN;

7
8 PROC SGPlot DATA=CARS1;
9   VBOX horsepower
  / category = type;
10
11   title 'Horsepower of cars by
12   types';
13 RUN;

```

82

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

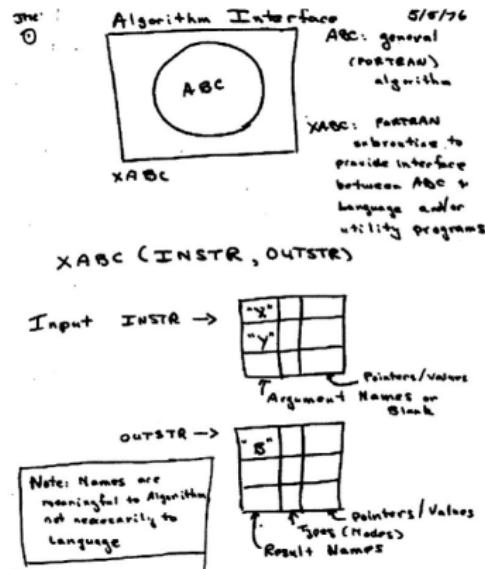
何为利器

所见即所得工具

所想即所得工具

27

- S 语言是 1976 年 AT&T 贝尔实验室开发的一种用于统计计算的解释型编程语言



图：S 语言的设计草图 (1976.5.5)

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

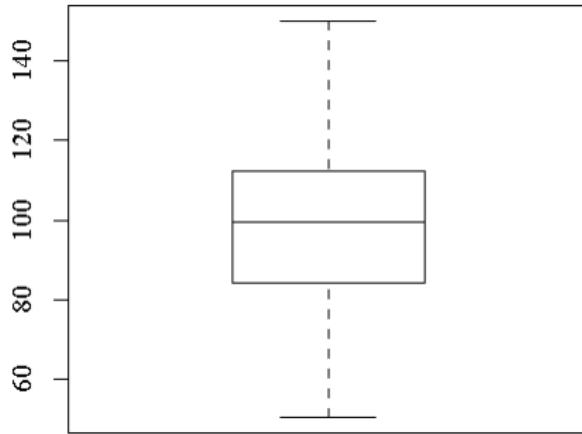
何为利器

所见即所得工具

所想即所得工具

28

- 与 SAS 内置脚本语言相比，S 语言更加符合现代程序语言的设计，方便灵活控制图形输出，制作既精美又专业的统计图形
- 能够与其他主流程序语言集成



1 `boxplot(Weight)`

图：箱形图 (Box plot)

82



# 所想即所得工具

## S-Plus

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

R 简介

R 绘图

R 图形混合嵌入

29

- S-Plus 是基于 S 语言开发的商业化统计软件，1993 年由 MathSoft 公司开发，2008 年起由 TIBCO 负责运维
- 与 SPSS 和 SAS 并称世界三大统计软件，具有专业的统计功能

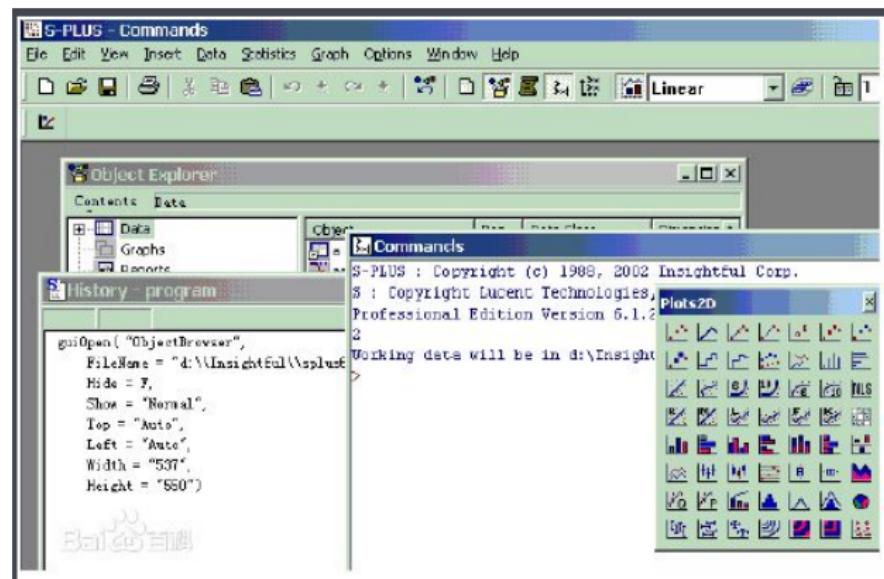


图: S-Plus 软件界面

82



# 所想即所得工具

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

30

## 所想即所得工具的优势

- ⌚ 想象力有多大，世界就有多精彩
- ⌚ 花有重开日，人无再少年
- ⌚ 深入算法内核，由术至道

## 所想即所得工具的劣势

- ⌚ 人机交互全靠指令输入，需要具备编程基础
- ⌚ 需要扎实的统计学基础，学习曲线陡峭
- ⌚ 实现简单的统计功能时不如“所见即所得工具”快捷

82



# 所想即所得工具

统计图形和 R

统计图形

统计绘图工具

为什么用绘图工具

何为利器

所见即所得工具

所想即所得工具

30

## 所想即所得工具的优势

- ✎ 想象力有多大，世界就有多精彩
- ✎ 花有重开日，人无再少年
- ✎ 深入算法内核，由术至道

## 所想即所得工具的劣势

- ✎ 人机交互全靠指令输入，需要具备编程基础
- ✎ 需要扎实的统计学基础，学习曲线陡峭
- ✎ 实现简单的统计功能时不如“所见即所得工具”快捷

82



# 目录

P&LRC

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

31

## 1 统计图形

## 2 统计绘图工具

## 3 R 简介

## 4 R 绘图

## 5 R 图形混合嵌入

82



# 为什么用 R

P&LRC

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

32



82

君子性非异也，善假于物也  
—《荀子·劝学》



# 为什么用 R

## R 的历史

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

33

- R 诞生于 1995 年，由新西兰 Auckland 大学统计学家 Robert Gentleman 和 Ross Ihaka 开发，而且完全开放源代码，是一款基于 GNU General Public License(GPL) 协议的开源软件；由于两位开发者的名字都以“R”开头，而且为了向 S 语言致敬，因此命名为 R
- R 是“所想即所得”工具，其核心是基于 S 语言设计的 R 语言，S 语言的代码可以不经过任何修改就在 R 中运行，因此 R 被看做是 S 语言的非商业化实现



Ross Ihaka



Robert Gentleman

82



# 为什么用 R

## R 的历史

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

33

- R 诞生于 1995 年，由新西兰 Auckland 大学统计学家 Robert Gentleman 和 Ross Ihaka 开发，而且完全开放源代码，是一款基于 GNU General Public License(GPL) 协议的开源软件；由于两位开发者的名字都以“R”开头，而且为了向 S 语言致敬，因此命名为 R
- R 是“所想即所得”工具，其核心是基于 S 语言设计的 R 语言，S 语言的代码可以不经过任何修改就在 R 中运行，因此 R 被看做是 S 语言的非商业化实现



Ross Ihaka



Robert Gentleman

82



# 为什么用 R

## R 的优势

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

34



[CRAN  
Mirror](#)  
[What's new?](#)  
[Task Views](#)  
[Search](#)

[About R](#)  
[R Home Page](#)  
[The R Journal](#)

[Software](#)  
[R Sources](#)  
[R Bitrates](#)  
[Packages](#)  
[Other](#)

[Documentation](#)  
[Manuals](#)  
[FAQs](#)  
[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages. Windows and Mac users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for Mac OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2017-11-30, Kite-Eating Tree) [R-3.4.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R\\_alpha](#) and [beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features](#) and [bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extensions [packages](#)

Questions About R

- 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.

### What are R and CRAN?

R is "GNU S", a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN [mirror](#) nearest to you to minimize network load.

### Submitting to CRAN

To "submit" a package to CRAN, check that your submission meets the [CRAN Repository Policy](#) and then use the [web form](#).

If this fails, upload to <http://CRAN.R-project.org/incoming/> and send an email to [CRAN-submissions@R-project.org](mailto:CRAN-submissions@R-project.org) following the policy. Please do not attach submissions to emails, because this will clutter up the mailboxes of half a dozen people.

图: R 的官方网站 CRAN(<https://cran.r-project.org/>)

82



# 为什么用 R

## R 的优势

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

34



[CRAN  
Mirror](#)  
[What's new?](#)  
[Link Views](#)  
[Search](#)

[About R](#)  
[R Home Page](#)  
[The R Journal](#)

[Software](#)  
[R Sources](#)  
[R Bitrates](#)  
[Packages](#)  
[Other](#)

[Documentation](#)  
[Manuals](#)  
[FAQs](#)  
[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages. Windows and Mac users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for Mac OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2017-11-30, Kite-Eating Tree) [R-3.4.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R\\_alpha](#) and [beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features](#) and [bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extensions [packages](#)

Questions About R

- 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.

### What are R and CRAN?

R is "GNU S", a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN [mirror](#) nearest to you to minimize network load.

### Submitting to CRAN

To "submit" a package to CRAN, check that your submission meets the [CRAN Repository Policy](#) and then use the [web form](#).

If this fails, upload to <https://CRAN.R-project.org/incoming/> and send an email to [CRAN-submissions@R-project.org](mailto:CRAN-submissions@R-project.org) following the policy. Please do not attach submissions to emails, because this will clutter up the mailboxes of half a dozen people.

图: R 的官方网站 CRAN(<https://cran.r-project.org/>)

82



# 为什么用 R

## R 的优势

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

34

- R 具备 S-Plus 几乎所有的优点，而且更加小巧轻便
- 开源项目，完全免费，这点是其他统计软件都不具备的
- 世界各地有大量研究机构和专业统计人员使用并自愿贡献代码，具有良好的生态系统



图：来自世界各地 R 的无私贡献者们

82



# 为什么用 R

## R 的优势

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

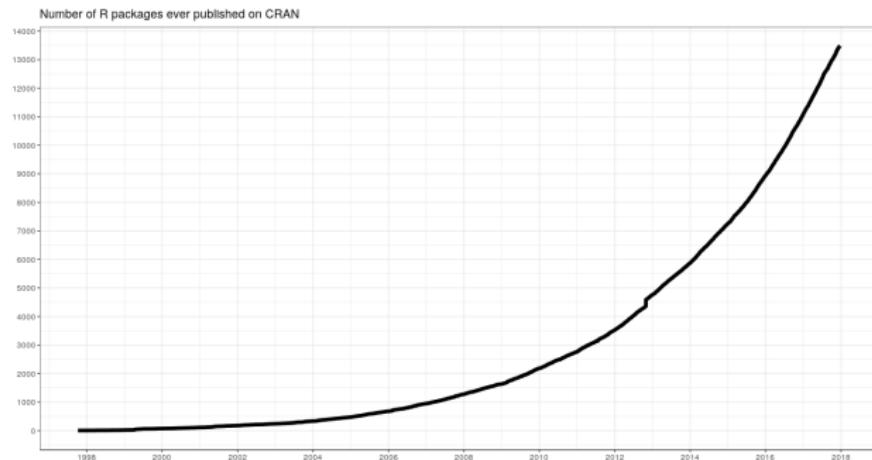
R 绘图

R 图形混合嵌入

82

35

- R 是高度模块化软件，通过各种程序包 (package) 来扩展其功能，目前 CRAN 上接收的程序包超过 12000 个，绝大多数来自自愿者的贡献
- R 语言语法简单，并不需要很高的编程技巧，而且与其他语言具有极好的兼容性
- 目前最新的统计模型和算法几乎都有 R 的实现版本，与统计研究前沿相接轨





# 为什么用 R

## R 的优势

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

35

- R 是高度模块化软件，通过各种程序包 (package) 来扩展其功能，目前 CRAN 上接收的程序包超过 12000 个，绝大多数来自自愿者的贡献
- R 语言语法简单，并不需要很高的编程技巧，而且与其他语言具有极好的兼容性
- 目前最新的统计模型和算法几乎都有 R 的实现版本，与统计研究前沿相接轨

## R 语言的兼容性

- 📎 **内部兼容：**由于 R 语言本身是解释性语言，执行效率较低，因此 R 的底层函数有很大一部分代码是 C 语言和 Fortran 语言编写的
- 📎 **外部兼容：**目前主流的编程语言，例如 JAVA、c++、python 等几乎都有相应的程序库来调用 R 语言编写的程序，来帮助这些编程语言简化统计计算和绘图相关的功能

82



# 为什么用 R

## R 的优势

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

35

- R 是高度模块化软件，通过各种程序包 (package) 来扩展其功能，目前 CRAN 上接收的程序包超过 12000 个，绝大多数来自自愿者的贡献
- R 语言语法简单，并不需要很高的编程技巧，而且与其他语言具有极好的兼容性
- 目前最新的统计模型和算法几乎都有 R 的实现版本，与统计研究前沿相接轨

<a href="#">Bayesian</a>	Bayesian Inference
<a href="#">ChemPhys</a>	Chemometrics and Computational Physics
<a href="#">ClinicalTrials</a>	Clinical Trial Design, Monitoring, and Analysis
<a href="#">Cluster</a>	Cluster Analysis & Finite Mixture Models
<a href="#">DifferentialEquations</a>	Differential Equations
<a href="#">Distributions</a>	Probability Distributions
<a href="#">Econometrics</a>	Econometrics
<a href="#">Environmetrics</a>	Analysis of Ecological and Environmental Data
<a href="#">ExperimentalDesign</a>	Design of Experiments (DoE) & Analysis of Experimental Data
<a href="#">ExtremeValue</a>	Extreme Value Analysis
<a href="#">Finance</a>	Empirical Finance
<a href="#">FunctionalData</a>	Functional Data Analysis
<a href="#">Genetics</a>	Statistical Genetics
<a href="#">Graphics</a>	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
<a href="#">HighPerformanceComputing</a>	High-Performance and Parallel Computing with R
<a href="#">MachineLearning</a>	Machine Learning & Statistical Learning
<a href="#">MedicalImaging</a>	Medical Image Analysis
<a href="#">MetaAnalysis</a>	Meta-Analysis
<a href="#">Multivariate</a>	Multivariate Statistics
<a href="#">NaturalLanguageProcessing</a>	Natural Language Processing

82



# 为什么用 R

## R 的优势

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

82

35

- R 是高度模块化软件，通过各种程序包 (package) 来扩展其功能，目前 CRAN 上接收的程序包超过 12000 个，绝大多数来自志愿者的贡献
- R 语言语法简单，并不需要很高的编程技巧，而且与其他语言具有极好的兼容性
- 目前最新的统计模型和算法几乎都有 R 的实现版本，与统计研究前沿相接轨

[NumericalMathematics](#)

[OfficialStatistics](#)

[Optimization](#)

[Pharmacokinetics](#)

[Phylogenetics](#)

[Psychometrics](#)

[ReproducibleResearch](#)

[Robust](#)

[SocialSciences](#)

[Spatial](#)

[SpatioTemporal](#)

[Survival](#)

[TimeSeries](#)

[WebTechnologies](#)

[gR](#)

Numerical Mathematics

Official Statistics & Survey Methodology

Optimization and Mathematical Programming

Analysis of Pharmacokinetic Data

Phylogenetics, Especially Comparative Methods

Psychometric Models and Methods

Reproducible Research

Robust Statistical Methods

Statistics for the Social Sciences

Analysis of Spatial Data

Handling and Analyzing Spatio-Temporal Data

Survival Analysis

Time Series Analysis

Web Technologies and Services

gRaphical Models in R

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

36

- 在 R 中进行的所有操作都是针对存储在内存中的对象
- 用户通过输入命令调用函数，分析结果可以被直接显示在屏幕上，也可以被存入某个对象或被写入硬盘(如图片对象)
- 因为分析结果本身也是对象，所以它们也能被视为数据并能像一般数据那样被处理分析
- 数据可以从本地磁盘读取，也可从远程服务器端获得

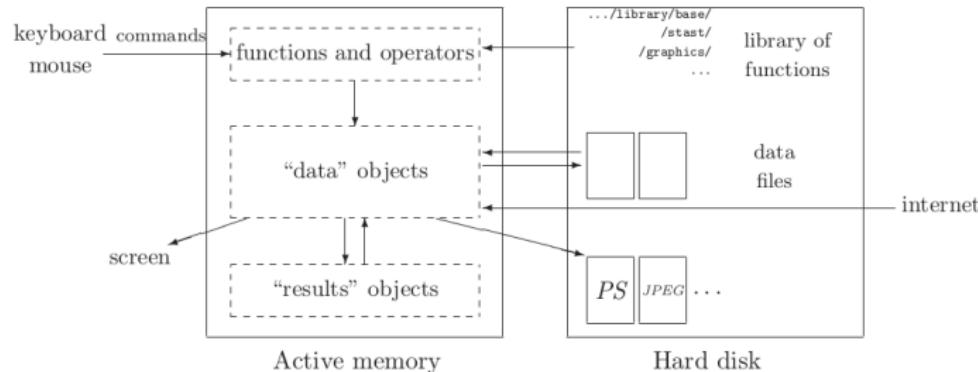


图: R 的工作原理

82



# 基础知识

## 安装运行环境

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

37

- R 的主安装程序由自愿者进行编译上传，基本上每个在 CRAN 上可以自由下载，包括 windows、linux 和 mac 三大主流平台
- 安装后的程序界面是一个交互式命令行终端，默认由一个“>”符号表示命令输入和结果输出指示符
- 在 windows 有一个简陋的 GUI，linux 和 mac 下默认只有 CLI 终端

The Comprehensive R Archive Network

**Download and Install R**

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

**Source Code for all Platforms**

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2017-11-30, Kite-Eating Tree) [R-3.4.3.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

**Questions About R**

- 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: R 在 CRAN 的下载界面

82



# 基础知识

## 安装运行环境

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

37

- R 的主安装程序由自愿者进行编译上传，基本上每个在 CRAN 上可以自由下载，包括 windows、linux 和 mac 三大主流平台
- 安装后的程序界面是一个交互式命令行终端，默认由一个“>”符号表示命令输入和结果输出指示符
- 在 windows 有一个简陋的 GUI，linux 和 mac 下默认只有 CLI 终端

82



# 基础知识

## 安装运行环境

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

37

- R 的主安装程序由自愿者进行编译上传，基本上每个在 CRAN 上可以自由下载，包括 windows、linux 和 mac 三大主流平台
- 安装后的程序界面是一个交互式命令行终端，默认由一个“>”符号表示命令输入和结果输出指示符
- 在 windows 有一个简陋的 GUI，linux 和 mac 下默认只有 CLI 终端

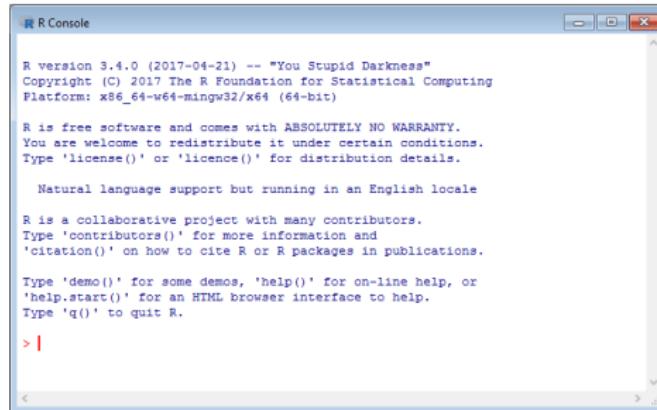


图: R 在 windows 下的 GUI

82



# 基础知识

## 安装运行环境

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

37

- R 的主安装程序由自愿者进行编译上传，基本上每个在 CRAN 上可以自由下载，包括 windows、linux 和 mac 三大主流平台
- 安装后的程序界面是一个交互式命令行终端，默认由一个“>”符号表示命令输入和结果输出指示符
- 在 windows 有一个简陋的 GUI，linux 和 mac 下默认只有 CLI 终端

```
# mono @ mono-vm in ~ [17:41:52]
$ R

R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

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

Natural language support but running in an English locale

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

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

图: R 在 linux 下的 cli 终端

82



# 基础知识

## 基本操作

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

82

38

- R 的第一种操作就是“输入命令 → 回车 → 输出结果”这种标准的交互式命令行方式

```
> x = runif(100); y = 0.2*x + 0.1*rnorm(100)
> fit = lm(y ~ x)
> summary(fit)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.30665 -0.05002 -0.01135  0.06047  0.24599 

Coefficients:
            Estimate Std. Error t value Pr(>t)    
(Intercept) 0.02052   0.01670   1.229   0.222    
x           0.17510   0.03107   5.636 1.67e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.08959 on 98 degrees of freedom
Multiple R-squared:  0.2448,    Adjusted R-squared:  0.2371 
F-statistic: 31.77 on 1 and 98 DF,  p-value: 1.671e-07
```



# 基础知识

## 基本操作

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

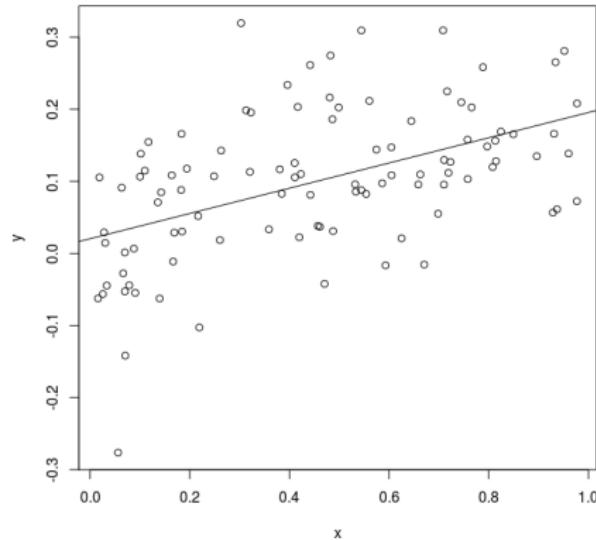
R 绘图

R 图形混合嵌入

38

- R 的第一种操作就是“输入命令 → 回车 → 输出结果”这种标准的交互式命令行方式

```
1 plot(x, y); abline(fit)
```



82





# 基础知识

## 基本操作

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

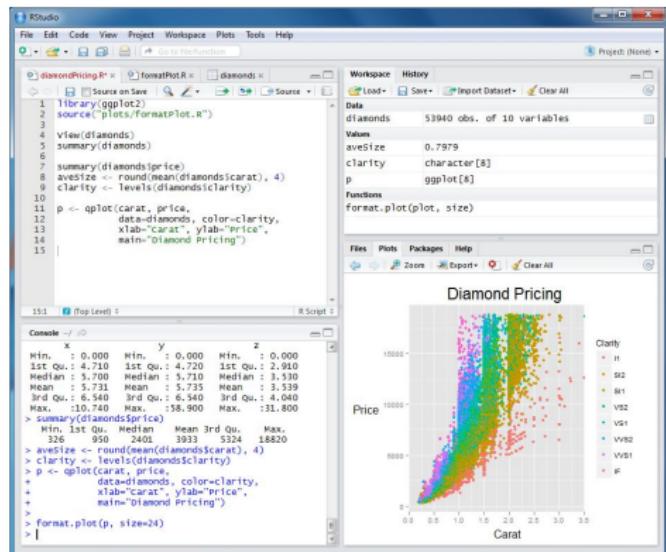
面向对象编程

R 绘图

R 图形混合嵌入

39

- 第二种方式是将脚本代码写在文件中，然后一次性运行
- 如果代码量很大建议使用第二种方式，因为文件比较容易修改和保存，而且目前有专用 IDE 可以辅助编写 R 脚本代码



图：RStudio 是目前最专业的 R IDE，具有大量针对 R 语言特点设计的功能，而且个人桌面版完全开源，可以免费使用

82



# 基础知识

## 程序包

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

40

### ● R 的程序包分为 base 包和 contrib 包

- base 包是安装 R 的时候就自带的，不需要单独安装，这类包的质量都非常高，性能稳定
- contrib 包是自愿者上传的，质量参次不齐；其中质量好且用户量大的扩展包有可能会纳入下一个版本的主程序包

[base](#)

Binaries for base distribution. This is what you want to [install R for the first time](#).

[contrib](#)

Binaries of contributed CRAN packages (for R >= 2.13 x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.

[old.contrib](#)

Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13 x; managed by Uwe Ligges).

[Rtools](#)

Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

图: CRAN 上下载 base 包和 contrib 包

82



# 基础知识

## 程序包

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

40

- R 的程序包分为 **base** 包和 **contrib** 包
- **base** 包是安装 R 的时候就自带的，不需要单独安装，这类包的质量都非常高，性能稳定
- **contrib** 包是自愿者上传的，质量参次不齐；其中质量好且用户量大的扩展包有可能会纳入下一个版本的主程序包

名称	用途
<b>base</b>	R 基础函数包
<b>methods</b>	用于 R 对象和编程工具的方法和类的定义
<b>datasets</b>	R 通用数据集
<b>graphics</b>	基础统计绘图包
<b>utils</b>	通用函数包
<b>stats</b>	基础统计计算包
<b>grDevices</b>	基础或 grid 图形设备

表：常用的 base 包

82



# 基础知识

## 程序包

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

40

- R 的程序包分为 **base** 包和 **contrib** 包
- **base** 包是安装 R 的时候就自带的，不需要单独安装，这类包的质量都非常高，性能稳定
- **contrib** 包是自愿者上传的，质量参次不齐；其中质量好且用户量大的扩展包有可能会纳入下一个版本的主程序包

名称	用途
cluster	聚类分析包
maptools	空间数据读取和处理包
spatstat	空间点数据分析包
sp	空间数据基础类包
spdep	空间自相关模型包
ggplot2	基于绘图语法的数据可视化包
knitr	R 文学编程包

表：常用的 contrib 包

82



# 基础知识

## 程序包

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

41

- base 包在 R 启动之后自动加载，可以直接使用；而 contrib 包则需要通过 library 函数调用，如果未安装相应包则会报错
- 通过 install.packages 函数安装 contrib 包

```
1 > library(sp)
2 Error in library(sp) : there is no package called 'sp'
```

82



# 基础知识

## 程序包

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

- base 包在 R 启动之后自动加载，可以直接使用；而 contrib 包则需要通过 library 函数调用，如果未安装相应包则会报错
- 通过 **install.packages** 函数安装 contrib 包

```
1 > library(sp)
2 Error in library(sp) : there is no package called 'sp'
```

```
1 > install.packages("sp")
2 Installing package into '/home/mono/Softwares/R/3.4'
3 (as 'lib' is unspecified)
4 --- Please select a CRAN mirror for use in this session ---
5 trying URL 'http://mirrors.tuna.tsinghua.edu.cn/CRAN/src/contrib/sp_1.2-6.tar.gz'
6 Content type 'application/octet-stream' length 1133739 bytes (1.1 MB)
7 =====
8 downloaded 1.1 MB
9
10 * installing *source* package 'sp' ...
```

41

82



# 基础知识

## 帮助系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

- 通过?命令或者help 函数 查看程序包中函数的本地帮助文档
- 通过 help.search 函数在整个帮助系统中进行关键字搜索
- find 函数可以根据名称精确查找对象， apropos 函数可以根据名称模糊查找对象

42

```
1 > ?lm
2 lm           package:stats          R Documentation
3
4 Fitting Linear Models
5
6 Description:
7
8   'lm' is used to fit linear models. It can be used to carry out regression, single
   ↪ stratum analysis of variance and analysis of covariance (although 'aov' may
   ↪ provide a more convenient interface for these).
9
10 Usage:
11
12 lm(formula, data, subset, weights, na.action, method = "qr", model = TRUE, x =
   ↪ FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL,
   ↪ offset, ...)
```

82



# 基础知识

## 帮助系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

- 通过**? 命令**或者**help 函数** 查看程序包中函数的本地帮助文档
- 通过**help.search 函数**在整个帮助系统中进行关键字搜索
- **find** 函数可以根据名称精确查找对象, **apropos** 函数可以根据名称模糊查找对象

```
1 > help.search("data input")
2 Help files with alias or concept or title matching 'data input' using fuzzy matching:
3 
4 
5 utils::read.DIF      Data Input from Spreadsheet
6 utils::read.table    Data Input
7 
8 
9 Type '?PKG::FOO' to inspect entries 'PKG::FOO', or 'TYPE?PKG::FOO' for entries
  ↪ like 'PKG::FOO-TYPE'.
```

42

82



# 基础知识 帮助系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

- 通过**? 命令**或者**help 函数** 查看程序包中函数的本地帮助文档
- 通过**help.search 函数**在整个帮助系统中进行关键字搜索
- **find 函数**可以根据名称精确查找对象, **apropos 函数**可以根据名称模糊查找对象

```
1 > find("lm")
2 [1] "package:stats"
3 > apropos("lm")
4 [1] "colMeans"           ".colMeans"        "confint.lm"      "contr.helmert"
5 [5] "dummy.coef.lm"     "getAllMethods"   "glm"          "glm.control"
6 [9] "glm.fit"            "KalmanForecast" "KalmanLike"    "KalmanRun"
7 [13] "KalmanSmooth"     "kappa.lm"       "lm"           ".lm.fit"
8 [17] "lm.fit"             "lm.influence"  "Im.wfit"       "model.matrix.lm"
9 [21] "nlm"                "nlminb"        "predict.glm"  "predict.lm"
10 [25] "residuals.glm"    "residuals.lm"  "summary.glm"  "summary.lm"
```

42

82



# 数据操作

## 对象 (object)

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

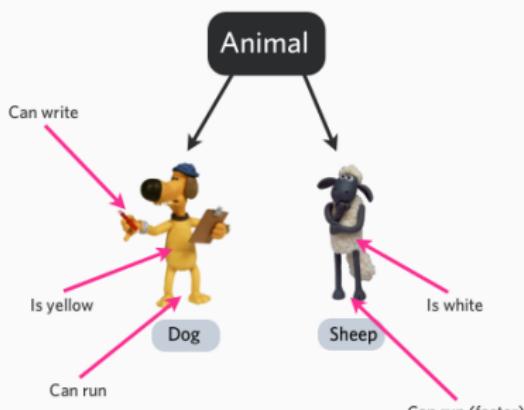
公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

43



在 R 语言中，几乎任何东西都是对象

82



# 数据操作

## 数据类型 (mode)

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

44

- 实数型 (real): 整数 (integer)、单精度 (single)、双精度 (double)
- 虚数型 (complex): 如  $10+21i$
- 字符型 (character, string): 如 "hello world"
- 逻辑型 (logical): TRUE(可以简写成 T), FALSE(可以简写成 F)
- 函数 (function)
- 表达式 (expression)

```
1 > x <- 1
2 > mode(x)
3 [1] "numeric"
4 > length(x)
5 [1] 1
6 > A <- "Gomphotherium"; compar <- TRUE; z <- 1i
7 > mode(A); mode(compar); mode(z)
8 [1] "character"
9 [1] "logical"
10 [1] "complex"
11 # 表达式
12 > x <- 3; y <- 2.5; z <- 1
13 > exp1 <- expression(x / (y + exp(z)))
14 > exp1
15 expression(x/(y + exp(z)))
16 > eval(exp1)
17 [1] 0.5749019
```

82



# 数据操作

## 数据结构

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

45

- R 语言中为了提高数据的使用效率，预定义了专门用于表示数据的对象，也就是数据结构，这些数据结构支撑了 R 强大的统计分析能力

数据结构	类型
向量 (vector)	数值型, 字符型, 复数型, 逻辑型
因子 (factor)	数值型, 字符型
数组 (array)	数值型, 字符型, 复数型, 逻辑型
矩阵 (matrix)	数值型, 字符型, 复数型, 逻辑型
数据框 (data.frame)	数值型, 字符型, 复数型, 逻辑型
列表 (list)	任意其他类型
时间序列 (ts)	数值型, 字符型, 复数型, 逻辑型

表：R 基础数据结构

82



# 数据操作

## 数据结构

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

46

## 向量 (vector)

- 向量是 R 中最基本的数据单元，向量中的对象类型必须相同
- 构建向量常用的函数: `rep()`、`c()`、`seq()`、`cbind()`、`rbind()` 等
- 向量的下标从 1 开始，这和其他计算机高级编程语言是不一样的！

```
1 > x <- 1:10; x
2 [1] 1 2 3 4 5 6 7 8 9 10
3 > x[3]
4 [1] 3
5 > c(7.11, 9.11, 9.19, 1.23)
6 [1] 7.11 9.11 9.19 1.23
7 > c("B", "A")
8 [1] "B" "A"
9 > rep(1, 10)
10 [1] 1 1 1 1 1 1 1 1 1 1
11 > seq(1, 5, 0.5)
12 [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
13 > cbind(0, rbind(1, 1:3))
14      [,1] [,2] [,3] [,4]
15 [1,]    0     1     1     1
16 [2,]    0     1     2     3
```

82



# 数据操作

## 数据结构

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

47

## 因子 (factor)

- 因子是对应统计学中的分类数据 (categorical data) 而设计的
- 因子形式上是一个对等长的向量元素进行分类 (分组) 的向量对象
- 因子数据具有水平 (level) 和标签 (label)，前者即分类变量的不同取值，后者即各类取值的名称

```
1 > factor(1:3, labels=c("A", "B", "C"))
2 [1] A B C
3 Levels: A B C
4 > (x = factor(c(1, 2, 3, 1, 1, 3, 2, 3, 3), levels = 1:3,
5 +   labels = c("g1", "g2", "g3")))
6 [1] g1 g2 g3 g1 g1 g3 g2 g3 g3
7 Levels: g1 g2 g3
```

82



# 数据操作

## 数据结构

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

48

## 数组 (array) 和矩阵 (matrix)

- 数组和矩阵是具有维度属性 (dimension) 的数据结构，实质是有附加属性 (维数 dim) 的向量
- 矩阵是数组的特例，它的维度为 2，用来指定行数和列数

```
1 # 二维矩阵
2 > matrix(data=5, nr=2, nc=2)
3   [,1] [,2]
4 [1,]    5    5
5 [2,]    5    5
6 # 三维数组
7 > array(1:24, c(3, 4, 2))
8   , , 1
9     [,1] [,2] [,3] [,4]
10    [1,]    1    4    7   10
11    [2,]    2    5    8   11
12    [3,]    3    6    9   12
13   , , 2
14     [,1] [,2] [,3] [,4]
15    [1,]   13   16   19   22
16    [2,]   14   17   20   23
17    [3,]   15   18   21   24
```

82



# 数据操作

## 数据结构

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

49

## 列表 (list)

- 列表是一种灵活的数据结构，用于生成包含不同类型对象的集合

```
1 # 创建 list
2 > x <- 1:4; y <- 2:4; L1 <- list(A=x, B=y); L1
3 $A
4 [1] 1 2 3 4
5
6 $B
7 [1] 2 3 4
8 # list 元素的引用
9 > L1[[1]]
10 [1] 1 2 3 4
11 > L1[["A"]]
12 $A
13 [1] 1 2 3 4
14 > L1[["A"]]
15 [1] 1 2 3 4
16 > L1$B
17 [1] 2 3 4
```

82



# 数据操作

## 数据结构

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

50

## 数据框 (data.frame)

- 数据框是由许多向量组成的一个二维的对象，主要用于保存建模所需要的数据
- 数据框的实质是一个“整齐的”列表，它只要求各列内的数据类型相同，而列之间的可以不同
- **数据框是 R 中最重要的一种数据结构，大多数数据都是以数据框形式输入到 R 中的**

```
1 # 创建数据框
2 # 数据框中的向量必须有相同的长度，如果其中有一个比其它的短，它将“循环”整数次（以使得其长度与其它向量相
3   同）
4 > x <- 1:4; M <- c(10, 35); y <- 2:4
5 > data.frame(x, M)
6   x   M
7 1 1 10
8 2 2 35
9 3 3 10
10 4 4 35
11 > data.frame(x, y)
12 Error in data.frame(x, y) :
  arguments imply differing number of rows: 4, 3
```

82



# 数据操作

## 数据输入输出

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

51

- R 的数据可以按照数据结构手动输入，也可以通过读取外部文件自动输入
- R base 包提供了文本文件 (ASCII) 的读取函数：read.table、scan 和 read.fwf 等
- R base 包也提供了文本文件 (ASCII) 的输出函数：write.table
- 另外，图形作为对象也可以输出，这将在后面会专门介绍

82



# 数据操作

## 数据输入输出

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

51

- R 的数据可以按照数据结构手动输入，也可以通过读取外部文件自动输入
- R base 包提供了文本文件 (ASCII) 的读取函数：`read.table`、`scan` 和 `read.fwf` 等
- R base 包也提供了文本文件 (ASCII) 的输出函数：`write.table`
- 另外，图形作为对象也可以输出，这将在后面会专门介绍

```
1 # 从外部读取 data.dat 文件，并且将数据赋给一个名为 mydata 的对象，这里 mydata 是一个 data.frame
  ↪ 数据结构
2 > mydata <- read.table("data.dat")
3
4 # read.table 的参数
5 read.table(file, header = FALSE, sep = "", quote = "\"\"", dec = ".", row.names,
  ↪ col.names, as.is = FALSE, na.strings = "NA", colClasses = NA, nrow = -1, skip
  ↪ = 0, check.names = TRUE, fill = !blank.lines.skip, strip.white = FALSE,
  ↪ blank.lines.skip = TRUE, comment.char = "#")
```

82



# 数据操作

## 数据输入输出

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

51

- R 的数据可以按照数据结构手动输入，也可以通过读取外部文件自动输入
- R base 包提供了文本文件 (ASCII) 的读取函数：**read.table**、**scan** 和 **read.fwf** 等
- R base 包也提供了文本文件 (ASCII) 的输出函数：**write.table**
- 另外，图形作为对象也可以输出，这将在后面会专门介绍

```
1 # 函数 write.table 可以在文件中写入一个对象，一般是写一个数据框，也可以是其它类型的对象
2 write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ", eol = "\n", na =
  ↪ "NA", dec = ".", row.names = TRUE, col.names = TRUE, qmethod =
  ↪ c("escape", "double"))
```

82



# 数据操作

## 数据输入输出

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

51

- R 的数据可以按照数据结构手动输入，也可以通过读取外部文件自动输入
- R base 包提供了文本文件 (ASCII) 的读取函数：**read.table**、**scan** 和 **read.fwf** 等
- R base 包也提供了文本文件 (ASCII) 的输出函数：**write.table**
- 另外，**图形作为对象也可以输出**，这将在后面会专门介绍

82



# 程序控制

## 控制语句

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

52

- if (条件) 表达式
- if (条件) 表达式 else 表达式
- if...else if...else if... else...
- ifelse (条件, yes, no)
- for (变量 in 向量) 表达式
- while (条件) 表达式

循环的编程模式在 R 中效率很低，尽量避免使用！

82



# 程序控制

## 控制语句

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

52

- if (条件) 表达式
- if (条件) 表达式 else 表达式
- if...else if...else if... else...
- ifelse (条件, yes, no)
- for (变量 in 向量) 表达式
- while (条件) 表达式

循环的编程模式在 R 中效率很低，尽量避免使用！

82



# 程序控制

## 控制语句

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

52

- if (条件) 表达式
- if (条件) 表达式 else 表达式
- if...else if...else if... else...
- ifelse (条件, yes, no)
- for (变量 in 向量) 表达式
- while (条件) 表达式

循环的编程模式在 R 中效率很低，尽量避免使用！

82



# 函数

P&LRC

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

53

- R 语言中可以定义函数，以便编程中可以以不同的参数值重复使用同一段代码
- 定义函数的方式为: `function(arglist) expr return(value)`; 其中 arglist 是参数列表,expr 是函数的主体,return() 用来返回函数值
- 函数是 R 中用于黑箱操作的重要实体

```
1 # 定义峰度函数 kurtosis, 该函数有两个参数, 数据向量 x 和是否删除缺失值 na.rm, 后者有默认值 FALSE
2 > kurtosis = function(x, na.rm = FALSE) {
3 +   if (na.rm)
4 +     x = x[!is.na(x)]
5 +   return(sum((x - mean(x))^4)/(length(x) * var(x)^2) - 3)
6 + }
7 > # 引用函数
8 > kurtosis(runif(100))
9 [1] -1.36086
```

82



# 公式和表达式

P&LRC

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

54

82



# 面向对象编程

P&LRC

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

55

R 绘图

R 图形混合嵌入

- R 语言中支持面向对象 (Object-Oriented, OO) 编程来提高代码的使用率，从而实现具体功能的扩展和模块化
- R 语言作为一种统计编程语言，需要用的 OO 的场景主要有以下两类：
  - 当需要用一种新的类型来表示数据，该类型与已有的数据类型有区别的时候
  - 当需要一个新的函数，该函数可以根据不同的参数类型做出不同的反应的时候
- R 语言中有四种 OO 的实现系统：S3、S4、RC(R5) 和 R6

82



# 面向对象编程

## S3 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

56

- S3 是 R 语言的第一种也是最简单的一种 OO 系统，也是 CRAN 程序包中最常用的一种 OO 系统
- S3 系统中方法 (method) 是属于函数而不是属于类，这种函数称为泛型函数(generic function)
- 泛型函数的形式是 generic.class(), 其实质是根据传入函数的第一个参数的类去调用相应的“子函数”

```
1 > library(pryr) # 调用 pryr 程序包检测某个方法是否是 S3 系统
2 > df <- data.frame(x = 1:10, y = letters[1:10])
3 > otype(df) # data.frame 是一个 S3 方法
4 [1] "S3"
5 # 调用 methods() 来查看属于某个泛型的所有方法
6 > methods("mean")
7 [1] mean.Date   mean.default   mean.difftime  mean.POSIXct   mean.POSIXlt
8 # 根据传入函数的第一个参数的类去调用相应的“子函数”
9 > (today <- Sys.Date())
10 [1] "2018-01-18"
11 > tenweeks <- seq(today, length.out=10, by="1 week")
12 > class(tenweeks)
13 [1] "Date"
14 > mean(tenweeks) # 这里调用的实际上是 mean.Date
15 [1] "2018-02-18"
```

82



# 面向对象编程

## S3 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

57

R 绘图

R 图形混合嵌入

- 创建 S3 对象最简单的方法是给一个变量增加 **class 属性**；也可以通过 **structure()** 函数创建
- 使用 **UseMethod()** 函数定义 S3 型泛型函数
- 通过 **NextMethod()** 函数实现继承

```
1 # 通过给变量增加 class 属性来创建 S3 型对象
2 > x <- 1; attr(x,'class') <- 'foo'
3 > x
4 [1] 1
5 attr("class")
6 [1] "foo"
7 > otype(x)
8 [1] "S3"
9 # 通过 structure 函数来创建 S3 型对象
10 > y <- structure(2, class='foo')
11 > y
12 [1] 2
13 attr("class")
14 [1] "foo"
15 > otype(y)
16 [1] "S3"
```

82



# 面向对象编程

## S3 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

57

- 创建 S3 对象最简单的方法是给一个变量增加 **class 属性**；也可以通过 **structure()** 函数创建
- 使用 **UseMethod()** 函数定义 S3 型泛型函数
- 通过 **NextMethod()** 函数实现继承

```
1 # 用 UseMethod 定义 S3 泛型函数
2 > teacher <- function(x,...) UseMethod("teacher")
3 # 定义了三个 teacher 的内部函数
4 > teacher.lecture <- function(x) print("上课")
5 > teacher.assignment <- function(x) print("布置作业")
6 > teacher.default <- function(x) print("你不是老师")
7 # 定义一个 S3 对象 a, 其 class 是 lecture
8 > a <- structure('A', class='lecture')
9 # teacher 泛型函数根据传入的类调取 teacher.lecture() 函数
10 > teacher(a)
11 [1] "上课"
12 # 默认泛型函数
13 > teacher()
14 [1] "你不是老师"
```

82



# 面向对象编程

## S3 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

57

R 绘图

R 图形混合嵌入

- 创建 S3 对象最简单的方法是给一个变量增加 **class 属性**；也可以通过 **structure()** 函数创建
- 使用 **UseMethod()** 函数定义 S3 型泛型函数
- 通过 **NextMethod()** 函数实现继承

```
1 # 定义一个 S3 泛型函数 node
2 > node <- function(x) UseMethod('node', x)
3 # 定义 node 的内部函数，其中 son 函数通过 NextMethod() 指向 father
4 > node.default <- function(x) "Default node"
5 > node.father <- function(x) c("father")
6 > node.son <- function(x) c("son", NextMethod())
7 # 定义对象 n 有两个 class，调用 node 函数会先执行 son 函数，再执行 father 函数，模拟了子函数调用父函
8   ↪ 数的过程
9 > n <- structure(1, class = c("son", "father"))
10 > node(n)
[1] "son"   "father"
```

82



# 面向对象编程

## S3 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

- 创建 S3 对象最简单的方法是给一个变量增加 **class 属性**；也可以通过 **structure()** 函数创建
- 使用 **UseMethod()** 函数定义 S3 型泛型函数
- 通过 **NextMethod()** 函数实现继承

### S3 系统的缺点

- ✎ S3 系统并不是真正的 OO，只是通过函数来模拟 OO
- ✎ S3 系统使用简单，但是很难处理复杂的对象关系
- ✎ S3 系统的内部函数并没有真正封装，可以绕过泛型函数检查直接被调用
- ✎ S3 系统的 class 属性可以被任意设置，没有检查机制

57

82



# 面向对象编程

## S4 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

- 相比 S3 系统，S4 系统有更明确和严谨的 OO 系统特征
- S4 有专门的类定义函数 `setClass()` 和类的实例化函数 `new()`
- 对象类型检查函数 `isValidity()`
- S4 泛型函数实现了方法定义和实现的分离，通过 `setGeneric()` 函数定义接口，`setMethod()` 函数定义实现方法；`standardGeneric` 定义泛型函数，类似 S3 中的 `UseMethod`

58

82



# 面向对象编程

## S4 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

58

- 相比 S3 系统，S4 系统有更明确和严谨的 OO 系统特征
- S4 有专门的类定义函数 `setClass()` 和类的实例化函数 `new()`
- 对象类型检查函数 `isValid()`
- S4 泛型函数实现了方法定义和实现的分离，通过 `setGeneric()` 函数定义接口，`setMethod()` 函数定义实现方法；`standardGeneric` 定义泛型函数，类似 S3 中的 `UseMethod`

```
1 # 定义 S4 类 person 以及 person 的子类 son
2 > setClass('person',slots=list(name="character",age="numeric"))
3 > setClass("son", slots=list(father="person",mother="person"),contains="person")
4 # 实例化对象
5 > father <- new("person",name="F",age=44)
6 > mother <- new("person",name="M",age=42)
7 > son <- new("son",name="S",age=16,father=father,mother=mother)
8 # 查看 son 对象的属性
9 > son@father
10 An object of class "person"
11 Slot "name":
12 [1] "F"
13
14 Slot "age":
15 [1] 44
```

82



# 面向对象编程

## S4 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

58

- 相比 S3 系统，S4 系统有更明确和严谨的 OO 系统特征
- S4 有专门的类定义函数 `setClass()` 和类的实例化函数 `new()`
- 对象类型检查函数 `isValidity()`
- S4 泛型函数实现了方法定义和实现的分离，通过 `setGeneric()` 函数定义接口，`setMethod()` 函数定义实现方法；`standardGeneric` 定义泛型函数，类似 S3 中的 `UseMethod`

```
1 # 定义 S4 类 person
2 > setClass('person',slots=list(name="character",age="numeric"))
3 # 传入错误的 age 类型
4 > bad <- new("person",name="bad",age="aaa")
5 Error in validObject(.Object) :
6   invalid class "person" object: invalid object for slot "age" in class "person": got class
7   ↪ "character", should be or extend class "numeric"
8 # 设置 age 属性的非负检查
9 > setValidity("person",function(object){
10 +   if(object@age <=0) stop("Age is negative."))
11 + })
12 # 传入 age 属性小于 0 时会报错
13 > bad2 <- new("person",name="bad",age=-1)
Error in validityMethod(object) : Age is negative.
```

82



# 面向对象编程

## S4 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

58

- 相比 S3 系统，S4 系统有更明确和严谨的 OO 系统特征
- S4 有专门的类定义函数 `setClass()` 和类的实例化函数 `new()`
- 对象类型检查函数 `isValidity()`
- S4 泛型函数实现了方法定义和实现的分离，通过 `setGeneric()` 函数定义接口，`setMethod()` 函数定义实现方法；`standardGeneric` 定义泛型函数，类似 S3 中的 `UseMethod`

```
1 # 定义 person 类,slots 参数定义类的属性
2 > setClass('person',slots=list(name="character",age="numeric"))
3 # 定义泛型函数 work, 即接口
4 > setGeneric("work",function(obj) standardGeneric("work"))
5 [1] "work"
6 # 定义 work 的实现函数，并指定参数类型为 person
7 > setMethod("work",signature(obj="person"),function(obj) cat(obj@name, " is "
8   ↪   "working"))
9 [1] "work"
10 # 创建 person 类型对象 a, 并将其传入 work 函数
11 > a <- new("person",name="Conan",age=16)
12 > work(a)
Conan is working
```

82



# 面向对象编程

## RC 和 R6 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

59

- RC(Reference classes), 又称为 R5, 是从 2.12 版本引入的新一代 OO 系统
- RC 系统的方法是在类中定义的, 而不是通过泛型函数
- RC 系统依赖于 S4 系统, 是对 OO 的进一步封装, 已经非常趋于主流编程语言中的 OO 实现
- R6 是一个完全独立的 R 程序包, 类似 RC 但不依赖于 S4 系统

82



# 面向对象编程

## RC 和 R6 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

59

R 绘图

R 图形混合嵌入

- RC(Reference classes), 又称为 R5, 是从 2.12 版本引入的新一代 OO 系统
- **RC 系统的方法是在类中定义的, 而不是通过泛型函数**
- RC 系统依赖于 S4 系统, 是对 OO 的进一步封装, 已经非常趋于主流编程语言中的 OO 实现
- R6 是一个完全独立的 R 程序包, 类似 RC 但不依赖于 S4 系统

```
1 # 定义一个 RC 类, 方法包括在定义中
2 > user<-setRefClass("user",
3 +   fields=list(name="character",favorite="vector"),
4 +   methods=list(
5 +     addFavorite=function(x){
6 +       favorite<-c(favorite,x)},
7 +     delFavorite=function(x){
8 +       favorite<-favorite[-which(favorite==x)]}))
9 # 实例化一个 u 对象
10 > u <- user$new(name="u",favorite=c('movie','football'))
11 # 操作方法
12 > u$addFavorite('shopping')
13 > u$favorite
14 [1] "movie"    "football" "shopping"
```

82



# 面向对象编程

## RC 和 R6 系统

统计图形和 R

统计图形

统计绘图工具

R 简介

为什么用 R

基础知识

数据操作

程序控制

函数

公式和表达式

面向对象编程

R 绘图

R 图形混合嵌入

59

- RC(Reference classes), 又称为 R5, 是从 2.12 版本引入的新一代 OO 系统
- **RC 系统的方法是在类中定义的, 而不是通过泛型函数**
- RC 系统依赖于 S4 系统, 是对 OO 的进一步封装, 已经非常趋于主流编程语言中的 OO 实现
- **R6 是一个完全独立的 R 程序包, 类似 RC 但不依赖于 S4 系统**

82



# 目录

P&LRC

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

60

## 1 统计图形

## 2 统计绘图工具

## 3 R 简介

## 4 R 绘图

## 5 R 图形混合嵌入

82



# 绘图基础知识

## 绘图设备

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

61

- 绘图设备用于管理 R 的图形输出，分为**窗口设备**和**图形设备**
- 窗口设备通过一个依赖于操作系统底层窗口来输出图形，在 Unix/Linux 系统中称为 X11，在 Windows 系统中称为 windows
- 图形设备用于将 R 对象输出到文件，例如 jpeg、pdf、png、bmp、metafile 等
- `layout()` 函数用于将绘图设备进行分割，图形将一次显示在不同的部分中；不再使用的绘图设备用 `dev.off()` 函数关闭

82



# 绘图基础知识

## 绘图设备

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

61

- 绘图设备用于管理 R 的图形输出，分为**窗口设备**和**图形设备**
- 窗口设备通过一个依赖于**操作系统底层窗口**来输出图形，在 Unix/Linux 系统中称为 X11，在 Windows 系统中称为 windows
- 图形设备用于将**R 对象输出到文件**，例如 jpeg、pdf、png、bmp、metafile 等
- `layout()` 函数用于将绘图设备进行分割，图形将一次显示在不同的部分中；不再使用的绘图设备用 `dev.off()` 函数关闭

```
1 # 开启三个绘图设备, X11 函数用于开启窗口设备, pdf 和 jpeg 函数用于输出图形到文件
2 > x11();pdf();jpeg()
3 > dev.list()
4 X11cairo      pdf      jpeg
5          2        3        4
6 # 关闭 2 号绘图设备
7 > dev.off(2)
```

82



# 绘图基础知识

## 绘图设备

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

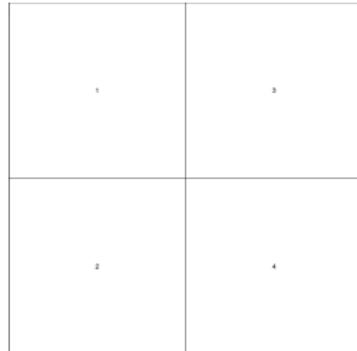
lattice 程序包

ggplot 程序包

R 图形混合嵌入

61

- 绘图设备用于管理 R 的图形输出，分为窗口设备和图形设备
- 窗口设备通过一个依赖于操作系统底层窗口来输出图形，在 Unix/Linux 系统中称为 X11，在 Windows 系统中称为 windows
- 图形设备用于将 R 对象输出到文件，例如 jpeg、pdf、png、bmp、metafile 等
- **layout()** 函数用于将绘图设备进行分割，图形将一次显示在不同的部分中；不再使用的绘图设备用**dev.off()**函数关闭



```
1 > mat <- matrix(1:4, 2, 2)
2 > layout(mat)
3 > layout.show(4)
```

82



# 绘图基础知识

## 绘图设备

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

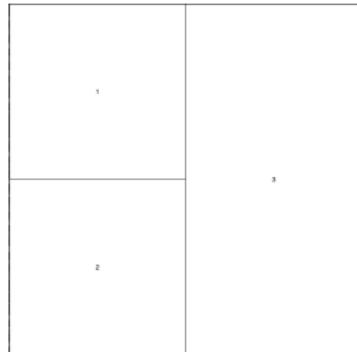
lattice 程序包

ggplot 程序包

R 图形混合嵌入

61

- 绘图设备用于管理 R 的图形输出，分为窗口设备和图形设备
- 窗口设备通过一个依赖于操作系统底层窗口来输出图形，在 Unix/Linux 系统中称为 X11，在 Windows 系统中称为 windows
- 图形设备用于将 R 对象输出到文件，例如 jpeg、pdf、png、bmp、metafile 等
- **layout()** 函数用于将绘图设备进行分割，图形将一次显示在不同的部分中；不再使用的绘图设备用**dev.off()**函数关闭



```
1 > mat <- matrix(c(1:3, 3), 2, 2)
2 > layout(mat)
3 > layout.show(3)
```

82



# 绘图基础知识

## 绘图函数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

62

- R 中的统计图形都是由相应的绘图函数生成，其包含了统计图形中各种细节的设置
- 绘图函数分为**高级绘图函数**和**低级绘图函数**
- 高级绘图函数用于快速绘制常见的统计图形，低级绘图函数用于在高级绘图函数绘制的统计图形基础上进行个性化的定制

82



# 绘图基础知识

## 绘图函数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

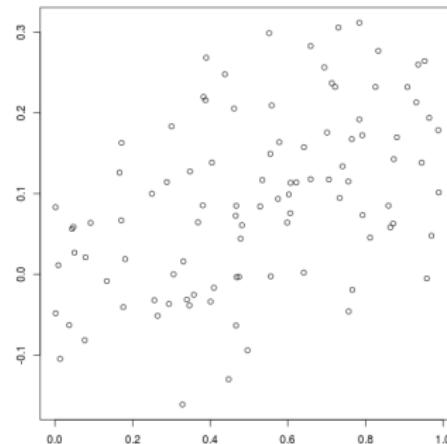
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

62



- R 中的统计图形都是由相应的绘图函数生成，其包含了统计图形中各种细节的设置
- 绘图函数分为高级绘图函数和低级绘图函数
- 高级绘图函数用于快速绘制常见的统计图形，低级绘图函数用于在高级绘图函数绘制的统计图形基础上进行个性化的定制

```
1 > x = runif(100); y = 0.2*x + 0.1*rnorm(100)
2 # 高级绘图函数 plot 绘制散点图
3 > plot(x, y)
```

82



# 绘图基础知识

## 绘图函数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

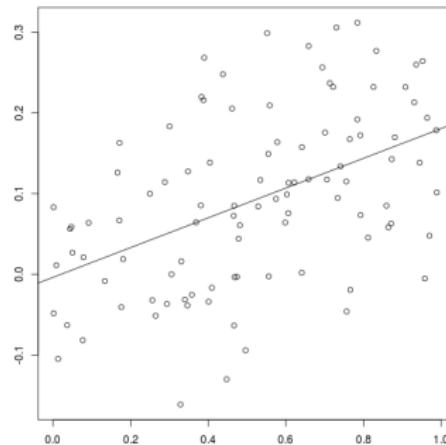
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

62



- R 中的统计图形都是由相应的绘图函数生成，其包含了统计图形中各种细节的设置
- 绘图函数分为高级绘图函数和低级绘图函数
- 高级绘图函数用于快速绘制常见的统计图形，低级绘图函数用于在高级绘图函数绘制的统计图形基础上进行个性化的定制

```
1 > x = runif(100); y = 0.2*x + 0.1*rnorm(100)
2 # 高级绘图函数 plot 绘制散点图
3 > plot(x, y)
4 # 回归模型拟合散点数据
5 > fit = lm(y ~ x)
6 # 低级绘图函数 abline 在原散点图基础上增加拟合直线
7 > abline(fit)
```

82



# 绘图基础知识

## 绘图参数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

63

- 除了低级绘图函数之外，图形的显示也可以用绘图参数来定制
- 绘图函数里面可以临时设置参数，不会影响后面其他绘图函数的效果
- `par()` 函数可以设置全局参数，全局参数只要绘图设备不关闭就会一直起作用

82



# 绘图基础知识

## 绘图参数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

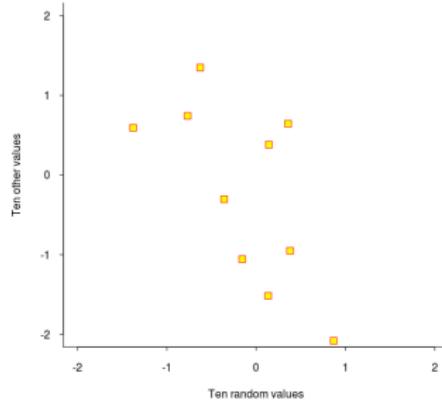
ggplot 程序包

R 图形混合嵌入

63

- 除了低级绘图函数之外，图形的显示也可以用绘图参数来定制
- 绘图函数里面可以临时设置参数，不会影响后面其他绘图函数的效果
- par() 函数可以设置全局参数，全局参数只要绘图设备不关闭就会一直起作用

通过 plot 函数的参数来定制散点图



```
1 > x<-rnorm(10); y<-rnorm(10)
2 # 通过设置 plot 函数的参数实现临时效果
3 > plot(x, y, xlab="Ten random values",
        <- ylab="Ten other values", xlim=c(-2, 2),
        <- ylim=c(-2, 2), pch=22, col="red",
        <- bg="yellow", bty="l", tcl=-.25, las=1,
        <- cex=1.5, main="通过 plot 函数的参数来定制散
        <- 点图")
```

82



# 绘图基础知识

## 绘图参数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

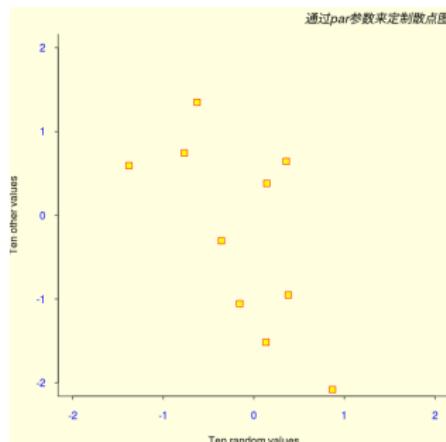
ggplot 程序包

R 图形混合嵌入

82

63

- 除了低级绘图函数之外，图形的显示也可以用绘图参数来定制
- 绘图函数里面可以临时设置参数，不会影响后面其他绘图函数的效果
- par() 函数可以设置全局参数，全局参数只要绘图设备不关闭就会一直起作用



```
1 # 缺省绘图参数被复制到 opar 对象
2 > opar <- par()
3 # 通过 par() 函数定制图形
4 > par(bg="lightyellow", col.axis="blue",
   +     mar=c(4, 4, 2.5, 0.25))
5 > plot(x, y, xlab="Ten random values",
   +       ylab="Ten other values",
6 +     xlim=c(-2, 2), ylim=c(-2, 2), pch=22,
   +     col="red", bg="yellow",
7 +     bty="1", tcl=-.25, las=1, cex=1.5)
8 # 通过低级绘图函数 title 为上图添加定制标题
9 > title("通过 par 参数来定制散点图", font.main=3,
10 +        adj=1)
11 # 恢复缺省绘图参数
12 > par(opar)
```



# 绘图基础知识

## 绘图参数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

64

参数名称	作用
adj	调整图中文字的相对位置
bg,fg	背景颜色和前景颜色
bty	设置图形边框样式
cex	图上元素(文本、符号等)的缩放倍数
col	图中符号的颜色
family,font	设置文本的字体族和字体样式
lab,mgp	设置坐标轴刻度数目和边界宽度
lend,ljoin	线条末端样式和线条相交处的样式
lheight	图中文本行高
lty,lwd	线条样式和宽度
mar,oma,pty	图形区域设置
pch	点符号样式
srt	字符串旋转角度
tck,tcl	坐标轴刻度线高度

表：par 函数的部分参数

82



# 绘图基础知识

## 绘图参数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

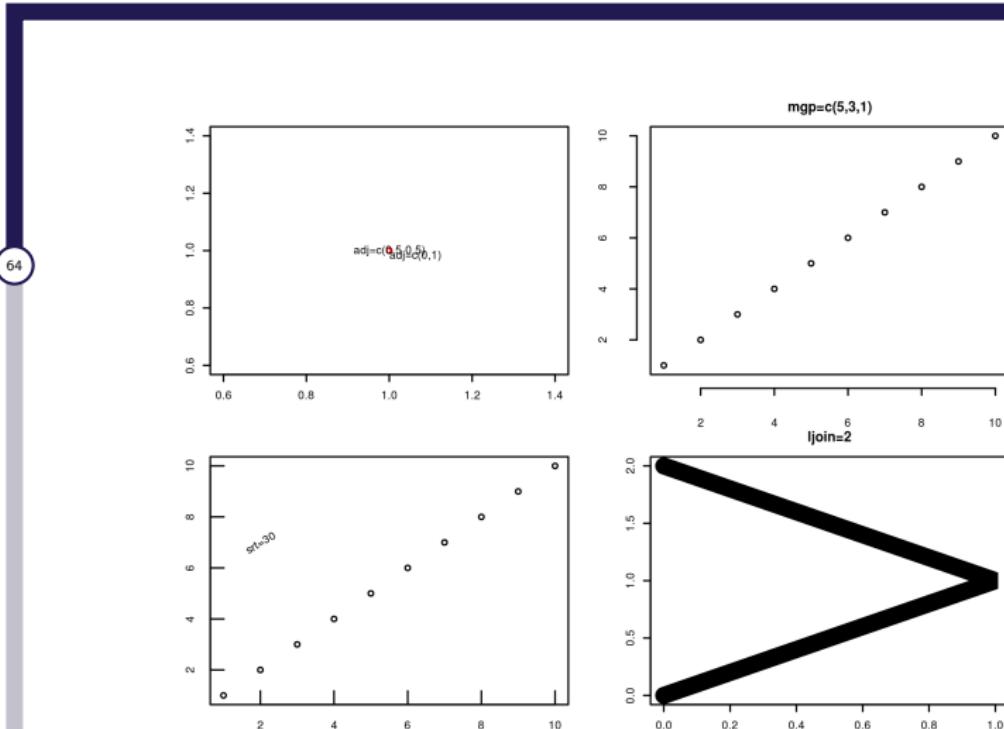


图: `par` 函数参数效果示例



# 绘图基础知识

## 绘图参数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

65

参数名称	作用
<code>type</code>	图形样式类型
<code>main,sub</code>	主标题和副标题
<code>xlab,ylab</code>	坐标轴标题
<code>asp</code>	图形横轴比
<code>x,y</code>	散点图的两个向量
<code>xlim,ylim</code>	坐标系界限
<code>axes</code>	是否画坐标轴
<code>frame.plot</code>	是否给图形加框
<code>panel.first</code>	作图前完成的工作
<code>panel.last</code>	作图后要完成的工作

表: plot 函数的部分参数

82



# 绘图基础知识

## 绘图参数

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

65

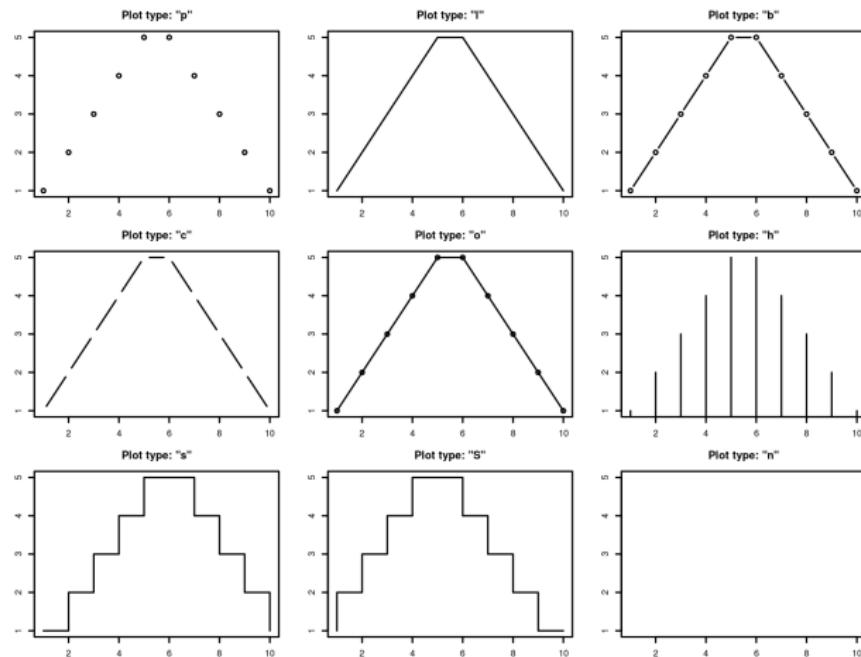


图: `plot` 函数参数 `type` 的九种效果示例

82



# 绘图元素拆解

P&LRC

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

66



82

庖丁为文惠君解牛，手之所触，肩之所倚，足之所履，膝之所踦，砉然向然，奏刀騞然，莫不中音。

—《庄子·养生主》



# 绘图元素拆解

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

67

- 统计图形的所有元素都可以在 R 语言中通过低级绘图函数和 `par()` 函数中的绘图参数实现高度定制化，使得统计图形的绘制在 R 中非常灵活

统计图形元素	常用函数
区域	<code>par</code>
颜色	<code>colors,palette,rgb,rainbow</code>
点	<code>points</code>
线	<code>lines,abline,arrows,segments,xspline</code>
面	<code>polygon,rect,box</code>
网格线	<code>grid</code>
文本	<code>text,title,mtext</code>
图例	<code>legend</code>
坐标轴	<code>axis</code>

表：统计图形的要素

82

# 绘图元素拆解

## 区域

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

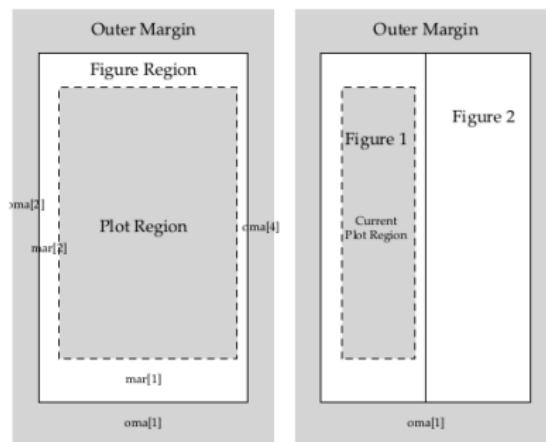
lattice 程序包

ggplot 程序包

R 图形混合嵌入

68

- R 的绘图设备分为三块区域：绘图区域 (Plot Region)、图形区域 (Figure Region) 和设备区域 (Device Region)
- 这三个区域对应两个边界：图形边界 (Figure Margin) 和外边界 (Outer Margin)；图形边界由 par 函数的 mar 参数设置，外边界由 oma 参数设置



图：最大的灰色区域是设备区域，设备区域内的白色实框区域是图形区域，最里面的灰色虚框区域是作图区域，所有的统计图形都是在作图区域内绘制

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

69

- 颜色元素由 `grDevices` 包支持，其内部编写了大量颜色函数
- 颜色函数分为固定颜色选择函数、颜色生成函数和特定颜色主题调色板

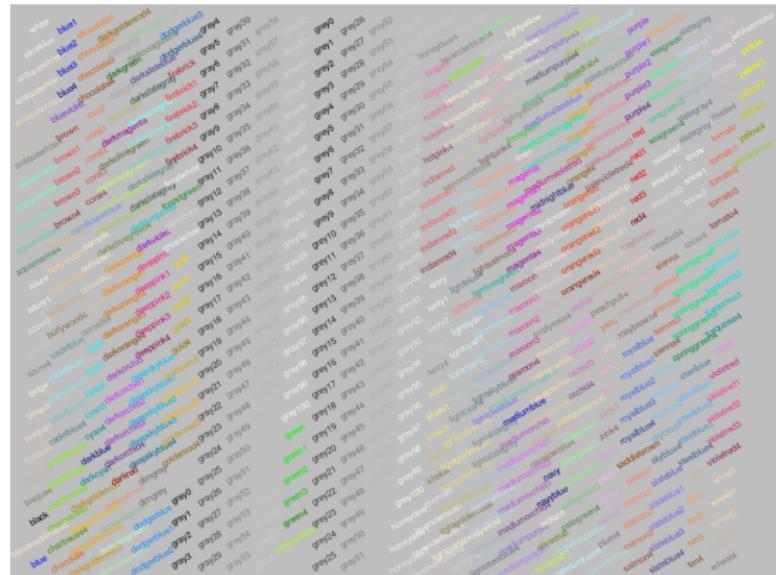


图: R 中的部分颜色

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

69

- 颜色元素由 `grDevices` 包支持，其内部编写了大量颜色函数
- 颜色函数分为固定颜色选择函数、颜色生成函数和特定颜色主题调色板

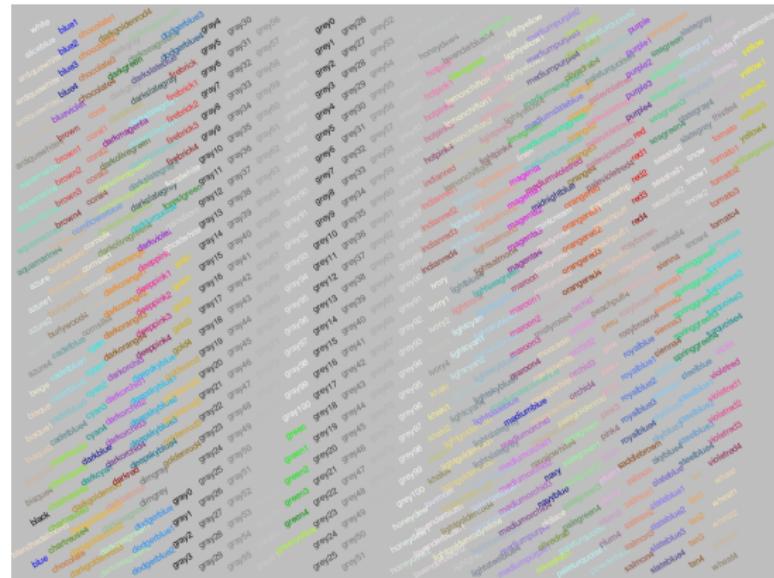


图: R 中的部分颜色

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

70

- 固定颜色选择函数: `colors()`、`palette()`

- `colors()` 函数直接通过英文名称来调取预设颜色
- `palette()` 函数用来设置调色板或者获得调色板颜色值。和 `colors` 不同的是 `palette` 函数结果并不是固定颜色；但是只要一旦设置了调试板，它的取值在下一次设置前会一直保存

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

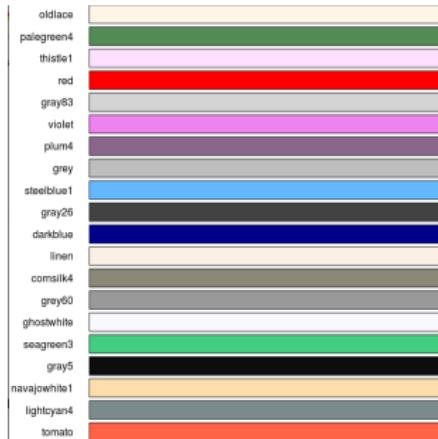
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

70



```
1 # 通过 colors 函数随机生成 20 种预设颜色
2 > sample(colors(), 20)
3 [1] "tomato"      "lightcyan4"   "navajowhite1"
4   ↪ "gray5"       "seagreen3"
5 [6] "ghostwhite"  "grey60"      "cornsilk4"
6   ↪ "linen"       "darkblue"
7 [11] "gray26"     "steelblue1"  "grey"
8   ↪ "plum4"       "violet"
9 [16] "gray83"     "red"        "thistle1"
10  ↪ "palegreen4" "oldlace"
```

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

70

- 固定颜色选择函数：`colors()`、`palette()`
- `colors()` 函数直接通过英文名称来调取预设颜色
- `palette()` 函数用来设置调色板或者获得调色板颜色值。和 `colors` 不同的是 `palette` 函数结果并不是固定颜色；但是只要一旦设置了调试板，它的取值在下一次设置前会一直保存

```
1 # palette 默认颜色
2 > palette()
3 [1] "black"    "red"      "green3"   "blue"     "cyan"     "magenta" "yellow"
4 [8] "gray"
5
6 # 更改后的调色板颜色
7 > palette(colors()[1:10])
8 > palette()
9 [1] "white"      "aliceblue"   "antiquewhite" "antiquewhite1"
10 [5] "antiquewhite2" "antiquewhite3" "antiquewhite4" "aquamarine"
11 [9] "aquamarine"  "aquamarine2"
12
13 > # 恢复默认调色板
14 > palette("default")
15 > palette()
16 [1] "black"    "red"      "green3"   "blue"     "cyan"     "magenta" "yellow"
17 [8] "gray"
```

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

71

- R 中还提供利用颜色生成原理构造颜色，如 RGB 模型（红绿蓝三原色）、HSV 色彩模型（色调、饱和度和纯度）、HCL 色彩模型（色调、色度和亮度）等
- 对应的颜色生成函数：`rgb()`、`hsv()`、`hcl()`、`gray()`等；相比固定颜色函数，颜色生成函数更加灵活，可以调配出任意颜色

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

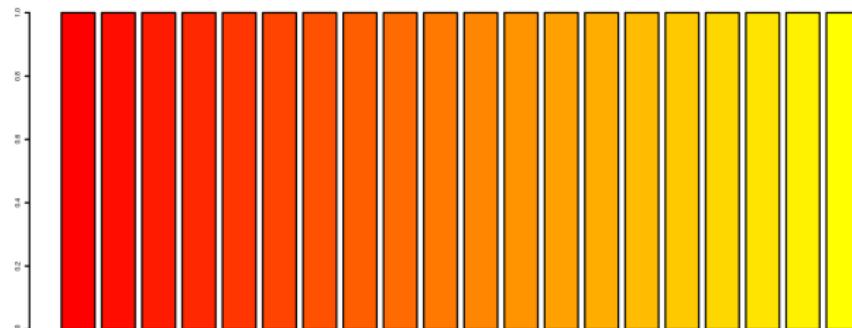
ggplot 程序包

R 图形混合嵌入

71

- R 中还提供利用颜色生成原理构造颜色，如 RGB 模型（红绿蓝三原色）、HSV 色彩模型（色调、饱和度和纯度）、HCL 色彩模型（色调、色度和亮度）等
- 对应的颜色生成函数：**rgb()**、**hsv()**、**hcl()**、**gray()**等；相比固定颜色函数，颜色生成函数更加灵活，可以调配出任意颜色

```
1 # 在 rgb() 函数中用一元线性函数控制绿色在 [0, 1] 上的取值，同时将红色和蓝色分别控制为 1 和 0，那么我们将
2   ↪ 得到从纯红色到黄色的一个颜色渐变
3 > (x = rgbd(1, seq(0, 1, length = 20), 0))
4 [1] "#FF0000" "#FF0D00" "#FF1B00" "#FF2800" "#FF3600" "#FF4300"
5 [7] "#FF5100" "#FF5E00" "#FF6B00" "#FF7900" "#FF8600" "#FF9400"
6 [13] "#FFA100" "#FFAE00" "#FFBC00" "#FFC900" "#FFD700" "#FFE400"
7 [19] "#FFF200" "#FFFF00"
> barplot(rep(1, 20), col = x)
```



82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

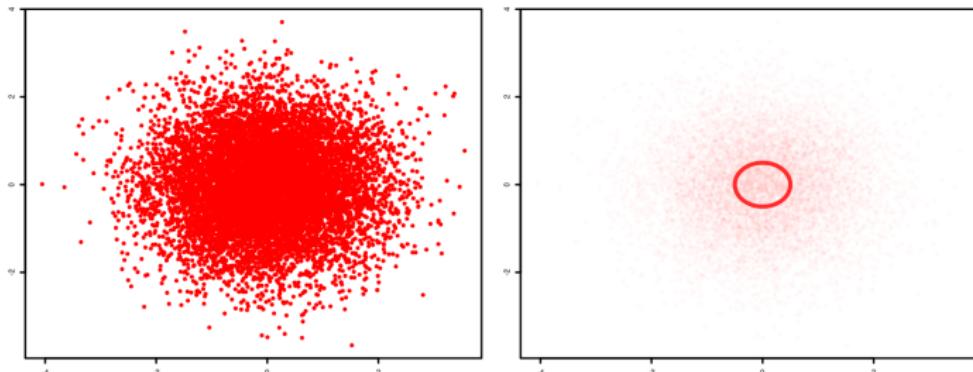
lattice 程序包

ggplot 程序包

R 图形混合嵌入

- R 中还提供利用颜色生成原理构造颜色，如 RGB 模型（红绿蓝三原色）、HSV 色彩模型（色调、饱和度和纯度）、HCL 色彩模型（色调、色度和亮度）等
- 对应的颜色生成函数：`rgb()`、`hsv()`、`hcl()`、`gray()`等；相比固定颜色函数，颜色生成函数更加灵活，可以调配出任意颜色

```
1 # 利用 rgb() 函数的透明度参数 alpha 生成半透明的显示效果
2 > library(MSG)
3 > data(BinormCircle)
4 > par(mfrow = c(1, 2), pch = 20, ann = FALSE, mar = c(2, 2 + 2, 0.5, 0.2))
5 > plot(BinormCircle, col = rgb(1, 0, 0))
6 > plot(BinormCircle, col = rgb(1, 0, 0, alpha = 0.01))
```



71

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

72

- 由于基于颜色生成原理构造颜色过于专业，因此 R 中还提供了一种比较简单的**特定颜色主题调试板**
- 特定颜色主题调试板用**渐变的颜色**来表现特定的主题

函数名称	效果
rainbow()	彩虹颜色 (红橙黄绿青蓝紫)
heat.colors()	从红色渐变到黄色再变到白色，适合表示“高温”、“白热化”
terrain.colors()	从绿色渐变到黄色再到棕色最后到白色，适合表示地理地形
topo.colors()	从蓝色渐变到青色再到黄色最后到棕色
cm.colors()	从青色渐变到白色再到粉红色
gray()	灰度 (由黑渐变到白)

表：常用的特定颜色主题调色板

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

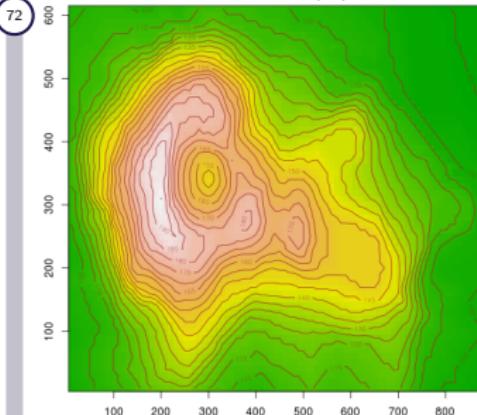
绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入



```
1 > x <- 10*(1:nrow(volcano))
2 > x.at <- seq(100, 800, by=100)
3 > y <- 10*(1:ncol(volcano))
4 > y.at <- seq(100, 600, by=100)
5 # 使用 Terrain Colors
6 > image(x, y, volcano, axes=FALSE,
7   ↪ xlab="", ylab="", col=terrain.colors(100))
8 > contour(x, y, volcano, levels=seq(90, 200,
9   ↪ by=5), add=TRUE, col="brown")
10 > axis(1, at=x.at)
11 > axis(2, at=y.at)
12 > box()
```



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

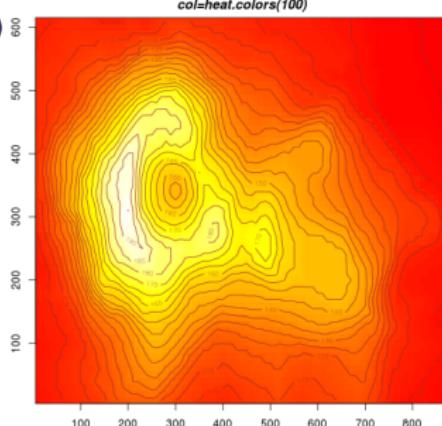
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

72



82

```
1 > x <- 10*(1:nrow(volcano))
2 > x.at <- seq(100, 800, by=100)
3 > y <- 10*(1:ncol(volcano))
4 > y.at <- seq(100, 600, by=100)
5 # 使用 Heat Colors
6 > image(x, y, volcano, axes=FALSE,
7   ↪ xlab="", ylab="", col=heat.colors(100))
8 > contour(x, y, volcano, levels=seq(90, 200,
9   ↪ by=5), add=TRUE, col="brown")
10 > axis(1, at=x.at)
11 > axis(2, at=y.at)
12 > box()
```



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

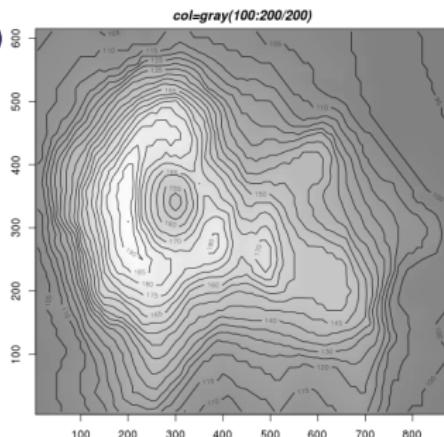
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

72



82

```
1 > x <- 10*(1:nrow(volcano))
2 > x.at <- seq(100, 800, by=100)
3 > y <- 10*(1:ncol(volcano))
4 > y.at <- seq(100, 600, by=100)
5 # 使用 Gray Colors
6 > image(x, y, volcano, axes=FALSE,
7   ↪ xlab="", ylab="", col=gray(100:200/200))
8 > contour(x, y, volcano, levels=seq(90, 200,
9   ↪ by=5), add=TRUE, col="brown")
10 > axis(1, at=x.at)
11 > axis(2, at=y.at)
12 > box()
```



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

73

- 另外, **RColorBrewer**包还提供了更简化的颜色生成函数, 只需要指定调色板名称, 再通过 `brewer.pal()` 函数就可以自动生成符合色彩科学的预设组合

调色板	作用
连续型调色板 (Sequential palettes)	生成一系列连续渐变的颜色, 通常用来标记连续型数值的大小
极端化调色板 (Diverging palettes)	生成用深色强调两端、浅色标示中部的系列颜色, 可用来标记数据中的离群点
离散型调色板 (Qualitative palettes)	生成一系列彼此差异比较明显的颜色, 通常用来标记分类数据

表: RColorBrewer 提供的调色板

82



# 绘图元素拆解

## 颜色

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

73



图：RColorBrewer 包中所有调色板颜色的演示：从上至下依次是连续型（18 种）、极端型（9 种）和离散型（8 种）调色板

82



# 绘图元素拆解

## 点

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

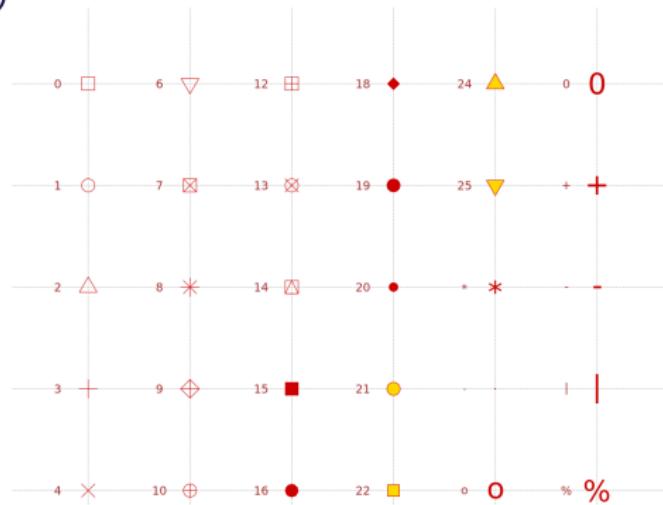
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

74



图：pch 参数不同取值的点类型，其中 col = “red3”, pch=21-25 的 bg = "gold"

82



# 绘图元素拆解

## 点

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

74

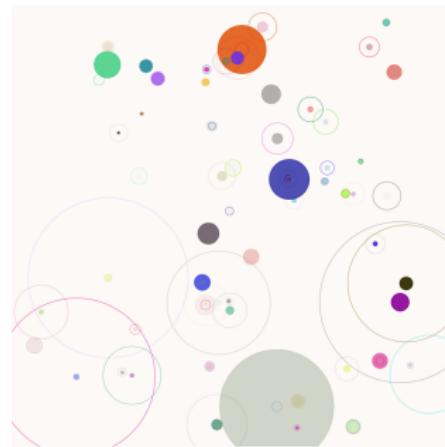


图: 点的艺术

- 点元素可以通过部分高级绘图函数中的 `pch` 参数绘制，也可以通过低级绘图函数 `points()` 绘制
- `points` 函数有两个重要参数 `pch` 和 `col`，前者用于设置点的样式，后者用于设置点的颜色；另外，`bg` 可以设置部分点类型的背景色，`lwd` 可以设置点边缘的宽度

```
1 # 利用随机数在随机位置生成随机样式的点
2 > set.seed(711)
3 > plot.new()
4 > size = c(replicate(n, 1/rbeta(2, 1.5, 4)))
5 > center = t(replicate(n, runif(2)))
6 > center = center[rep(1:n, each = 2), ]
7 > color = apply(replicate(2 * n, sample(c(0:9,
8   ↪ LETTERS[1:6]), 8, replace = TRUE)),
9   ↪ 2, function(x) sprintf("#%s", paste(x,
8   ↪ collapse = "")))
9 > points(center, cex = size, pch = rep(20:21, n),
8   ↪ col = color)
```

82



# 绘图元素拆解

## 点

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

74



图：点的艺术

- 点元素可以通过部分高级绘图函数中的 `pch` 参数绘制，也可以通过低级绘图函数 `points()` 绘制
- `points` 函数有两个重要参数 `pch` 和 `col`，前者用于设置点的样式，后者用于设置点的颜色；另外，`bg` 可以设置部分点类型的背景色，`lwd` 可以设置点边缘的宽度

```
1 # 利用随机数在随机位置生成随机样式的点
2 > set.seed(711)
3 > plot.new()
4 > size = c(replicate(n, 1/rbeta(2, 1.5, 4)))
5 > center = t(replicate(n, runif(2)))
6 > center = center[rep(1:n, each = 2), ]
7 > color = apply(replicate(2 * n, sample(c(0:9,
8   ↪ LETTERS[1:6]), 8, replace = TRUE)),
9   ↪ 2, function(x) sprintf("#%s", paste(x,
   ↪ collapse = "")))
> points(center, cex = size, pch = rep(20:21, n),
   ↪ col = color)
```

82



# 绘图元素拆解

## 点

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

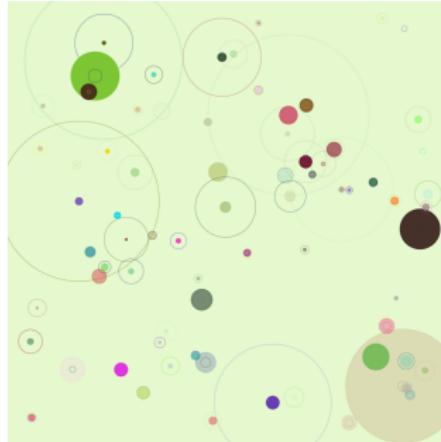
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

74



图：点的艺术

- 点元素可以通过部分高级绘图函数中的 `pch` 参数绘制，也可以通过低级绘图函数 `points()` 绘制
- `points` 函数有两个重要参数 `pch` 和 `col`，前者用于设置点的样式，后者用于设置点的颜色；另外，`bg` 可以设置部分点类型的背景色，`lwd` 可以设置点边缘的宽度

```
1 # 利用随机数在随机位置生成随机样式的点
2 > set.seed(711)
3 > plot.new()
4 > size = c(replicate(n, 1/rbeta(2, 1.5, 4)))
5 > center = t(replicate(n, runif(2)))
6 > center = center[rep(1:n, each = 2), ]
7 > color = apply(replicate(2 * n, sample(c(0:9,
8   ↪ LETTERS[1:6]), 8, replace = TRUE)),
9   ↪ 2, function(x) sprintf("#%s", paste(x,
   ↪ collapse = "")))
> points(center, cex = size, pch = rep(20:21, n),
   ↪ col = color)
```

82



# 绘图元素拆解

## 点

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

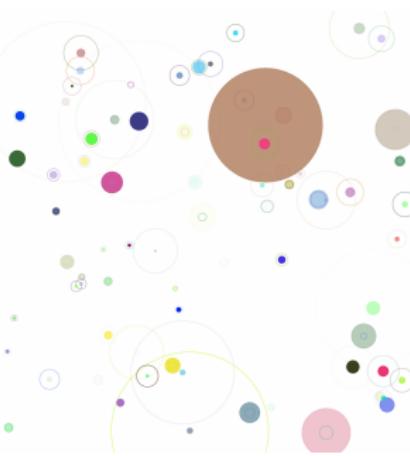
lattice 程序包

ggplot 程序包

R 图形混合嵌入

74

- 点元素可以通过部分高级绘图函数中的 `pch` 参数绘制，也可以通过低级绘图函数 `points()` 绘制
- `points` 函数有两个重要参数 `pch` 和 `col`，前者用于设置点的样式，后者用于设置点的颜色；另外，`bg` 可以设置部分点类型的背景色，`lwd` 可以设置点边缘的宽度



图：点的艺术

82

```
1 # 利用随机数在随机位置生成随机样式的点
2 > set.seed(711)
3 > plot.new()
4 > size = c(replicate(n, 1/rbeta(2, 1.5, 4)))
5 > center = t(replicate(n, runif(2)))
6 > center = center[rep(1:n, each = 2), ]
7 > color = apply(replicate(2 * n, sample(c(0:9,
8   ↪ LETTERS[1:6]), 8, replace = TRUE)),
9   ↪ 2, function(x) sprintf("#%s", paste(x,
10    ↪ collapse = "")))
11 > points(center, cex = size, pch = rep(20:21, n),
12   ↪ col = color)
```



# 绘图元素拆解

## 线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

75

- R 中的线元素包括直线、线段、多段线、箭头和样条曲线
  - 直线: `abline()` 函数
  - 线段: `segment()` 函数
  - 多段线: `lines()` 函数
  - 箭头: `arrows()` 函数
  - 样条曲线: `xspline()` 函数

82



# 绘图元素拆解

## 线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

75

- R 中的线元素包括直线、线段、多段线、箭头和样条曲线
- 直线: `abline()` 函数
- 线段: `segment()` 函数
- 多段线: `lines()` 函数
- 箭头: `arrows()` 函数
- 样条曲线: `xspline()` 函数

```
1 # 主要参数: a 是截距, b 是斜率, h 是画水平线时的纵轴值, v 是画垂直线时的横轴值, reg 是一个能用函数
  ↪ coef() 提取系数
2 (包含斜率和截距) 的 R 对象, 典型的就是用线性模型 (回归) 生成的对象, 系数是一个长度为2的向量, 分别为截距和
  ↪ 斜率
3 abline(a = NULL, b = NULL, h = NULL, v = NULL, reg = NULL, coef = NULL, untf
  ↪ = FALSE, ...)
```

82



# 绘图元素拆解

## 线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

75

- R 中的线元素包括直线、线段、多段线、箭头和样条曲线
- 直线: `abline()` 函数
- 线段: `segment()` 函数
- 多段线: `lines()` 函数
- 箭头: `arrows()` 函数
- 样条曲线: `xspline()` 函数

```
1 # 主要参数: 前四个参数表示线段的起点和终点坐标
2 segments(x0, y0, x1 = x0, y1 = y0, col = par("fg"), lty = par("lty"), lwd =
   ↪   par("lwd"), ...)
```

82



# 绘图元素拆解

## 线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

75

- R 中的线元素包括直线、线段、多段线、箭头和样条曲线
- 直线: `abline()` 函数
- 线段: `segment()` 函数
- 多段线: `lines()` 函数
- 箭头: `arrows()` 函数
- 样条曲线: `xspline()` 函数

```
1 # 主要参数: x 和 y 是元素个数相同的向量, 表示 n 组多段线的节点
2 lines(x, y = NULL, type = "1", ...)
```

82



# 绘图元素拆解

## 线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

75

- R 中的线元素包括直线、线段、多段线、箭头和样条曲线
- 直线: `abline()` 函数
- 线段: `segment()` 函数
- 多段线: `lines()` 函数
- 箭头: `arrows()` 函数
- 样条曲线: `xspline()` 函数

1 # 主要参数: 前四个参数表示箭头的起点和终点坐标, length 表示箭头上短线的长度 (单位: 英寸), angle 表示  
  → 箭头尖短线的角度, code 表示箭头的样式 (整数 1~3 分别表示尾部箭头、首部箭头和两端都带箭头)  
2 arrows(x0, y0, x1 = x0, y1 = y0, length = 0.25, angle = 30, code = 2, col = par("fg"),  
  → lty = par("lty"), lwd = par("lwd"), ...)

82



# 绘图元素拆解

## 线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

75

- R 中的线元素包括直线、线段、多段线、箭头和样条曲线
- 直线: `abline()` 函数
- 线段: `segment()` 函数
- 多段线: `lines()` 函数
- 箭头: `arrows()` 函数
- 样条曲线: `xspline()` 函数

```
1 # 主要参数: 前两个参数给定点的位置, shape 为样条的形状, 取值在 [-1, 1] 之间, 当取值为负数时, 曲线穿过给定的点, 负值绝对值越小则曲线的角度越尖锐, 反之角度越圆滑, shape 取值为正数时, 曲线脱离给定的点,  
→ 正值越小越靠近给定点; open 决定是否样条曲线封闭; repEnds 为逻辑值, 当样条曲线不封闭时, 该参数  
→ 决定是否重复使用端点上的点; draw 决定是否画线, 若为 FALSE, 则仅仅计算曲线的坐标位置而不画线;  
→ border 为曲线的颜色; col 为封闭曲线的填充颜色  
2 xspline(x, y = NULL, shape = 0, open = TRUE, repEnds = TRUE, draw = TRUE,  
→ border = par("fg"), col = NA, ...)
```

82



# 绘图元素拆解

## 线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

75

- R 中的线元素包括直线、线段、多段线、箭头和样条曲线
- 直线: `abline()` 函数
- 线段: `segment()` 函数
- 多段线: `lines()` 函数
- 箭头: `arrows()` 函数
- 样条曲线: `xspline()` 函数

### 特殊参数

线的 `lty` 参数相当灵活，除了取值 0-6 之外，还可以根据一个十六进制的数字串来设定线条的虚实，具体原理是：奇数位上的数字表示画相应长度的实线，偶数位上的数字则表示空缺相应的长度，这样可以实现几乎无数种线条样式；例如，'711911' 表示：7 单位长实线、1 单位长空白、1 单位长实线、9 单位长空白、1 单位长实线、1 单位长空白。

当设定 `type='h'` 时，`col` 参数可以使用向量，此时各条竖线都将使用不同的颜色；除此之外，若其它参数使用了向量，那么只有向量的第一个元素会被使用，其它元素都将被忽略掉。

82



# 绘图元素拆解

## 线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

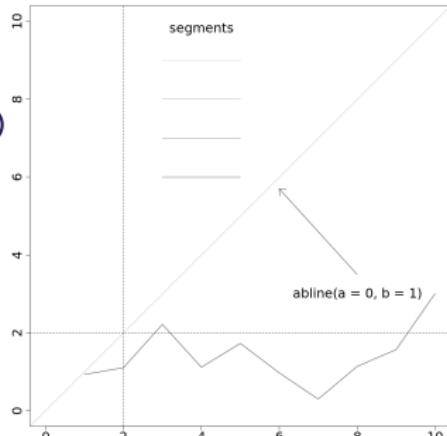
绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入



```
1 # 不作图, 只画出框架, 且指定坐标轴范围
2 > plot(1:10, type = "n", xlim = c(0, 10), ylim =
   +       c(0,10))
3 # 10 个正态随机数绝对值的波动线
4 > lines(1:10, abs(rnorm(10)))
5 # 不同的直线
6 > abline(a = 0, b = 1, col = "gray")
7 > abline(v = 2, lty = 2)
8 > abline(h = 2, lty = 2)
9 # 添加文本
10 > text(8, 3, "abline(a = 0, b = 1)")
11 # 添加箭头
12 > arrows(8, 3.5, 6, 5.7, angle = 40)
13 # 参数用了向量: 不同灰度的线段
14 > segments(rep(3, 4), 6:9, rep(5, 4), 6:9, col =
   +           gray(seq(0.2,0.8, length = 4)))
15 > text(4, 9.8, "segments")
```

图：直线、曲线、线段和箭头示例



# 绘图元素拆解

## 线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

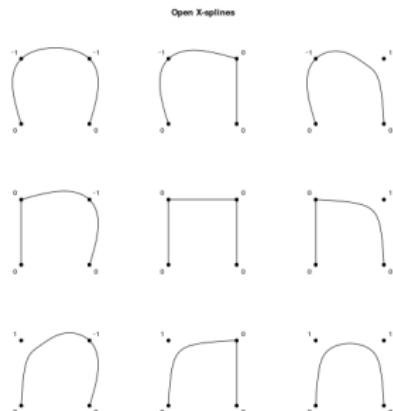
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

76



```
> xsplineTest <- function(s, open = TRUE, x =  
+   c(1,1,3,3)/4, y = c(1,3,3,1)/4, ...){  
+   plot(c(0,1), c(0,1), type = "n", axes = FALSE,  
+     xlab = "", ylab = "")  
+   points(x, y, pch = 19)  
+   xspline(x, y, s, open, ...)  
+   text(x+0.05*c(-1,-1,1,1), y+0.05*c(-1,1,1,-1),  
+     s)}  
> xsplineTest(c(0, -1, -1, 0))  
> xsplineTest(c(0, -1, 0, 0))  
> xsplineTest(c(0, -1, 1, 0))  
> xsplineTest(c(0, 0, -1, 0))  
> xsplineTest(c(0, 0, 0, 0))  
> xsplineTest(c(0, 0, 1, 0))  
> xsplineTest(c(0, 1, -1, 0))  
> xsplineTest(c(0, 1, 0, 0))  
> xsplineTest(c(0, 1, 1, 0))  
> title("Open X-splines", outer = TRUE)
```

图：样条曲线示例

82



# 绘图元素拆解

面

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

77

- **polygon()** 函数用于绘制多边形
- 矩形是多边形的特例，有专门绘制矩形的函数**rect()**
- 整幅图形的边框也是一种特殊的矩形，用**box()**函数绘制

```
1 # rect 函数用于绘制矩形
2 # 主要参数: 前四个参数分别绘制左下角和右上角的坐标; angle 参数设置填充线条的角度; col 设置填充颜色;
  ↪   border 设置边框颜色
3 rect(xleft, ybottom, xright, ytop, density = NULL, angle = 45, col = NA, border =
  ↪   NULL, lty = par("lty"), lwd = par("lwd"), ...)
```

```
1 # polygon 函数用于绘制多边形
2 # 主要参数: density 参数设置阴影线的填充密度; angle 参数设置填充线条的角度; col 设置填充颜色; border
  ↪   设置边框颜色
3 polygon(x, y = NULL, density = NULL, angle = 45, border = NULL, col = NA, lty =
  ↪   par("lty"), ..., fillOddEven = FALSE)
```

82



# 绘图元素拆解

面

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

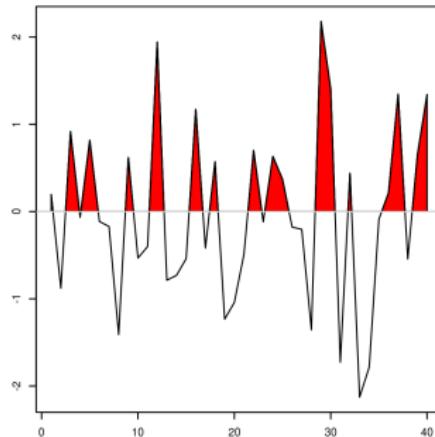
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

77



```
1 # 产生 40 个正态随机数
2 > x = rnorm(40)
3 # 画线图
4 > plot(x, xlab = "", type = "l")
5 # 绘制多边形的连线路径, 用红色填充
6 > polygon(c(1, 1:40, 40), c(0, x, 0), col = "red")
7 # 获取当前图形区域坐标范围, 以便下用
8 > xy = par("usr")
9 # 用红色矩形挡住了 0 以下的部分
10 > rect(xy[1], xy[3], xy[2], 0, col = "red")
11 # 重画一遍 x 的线条
12 > lines(x)
13 # 添加水平线
14 > abline(h = 0, col = "lightgray")
```

图：多边形和矩形函数示例 (0 上下数值分别用不同颜色填充)

82



# 绘图元素拆解

## 网格线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

78

- 网格线用对齐坐标轴的横纵直线来辅助获取更精确的元素位置
- `grid()` 函数是专门用于绘制网格线的函数
- `par()` 函数的 `tcl` 参数也可以用于绘制网格线；或者是通过直线函数 `abline()` 的 `h` 和 `v` 参数绘制横纵直线来表示网格线

```
1 # 主要参数：网格线默认颜色 col 为浅灰色，线条样式 lty 为点线，这是一种比较美观的设置；参数 nx 和 ny 分别
   ↪ 表示横纵轴上网格线的条数；equilogs 参数意思是，当坐标取了对数之后，是依然使用等距的网格线
   ↪ (TRUE) 还是根据对数函数使用不等距的网格线 (FALSE)
2 grid(nx = NULL, ny = nx, col = "lightgray", lty = "dotted", lwd = par("lwd"),
   ↪ equilogs = TRUE)
```

82



# 绘图元素拆解

## 网格线

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

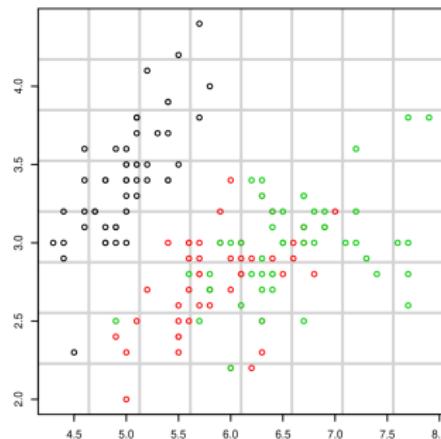
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

78



82

- 网格线用对齐坐标轴的横纵直线来辅助获取更精确的元素位置
- `grid()`函数是专门用于绘制网格线的函数
- `par()`函数的`tcl`参数也可以用于绘制网格线；或者是通过直线函数`abline()`的`h`和`v`参数绘制横纵直线来表示网格线

```
1 > with(iris,
2 { 
3   plot(Sepal.Length, Sepal.Width, col =
4     ↪ as.integer(Species), panel.first =
5     ↪ grid(8, lty = 1, lwd = 2))
6 })
```

图: `grid` 函数示例



# 绘图元素拆解

## 文本

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

79

- R 中把统计图形中的文本分为三类：**标题、任意文本和图形周边文本**
- 对应的函数分别为：**title()、text()和mtext()**

```
1 # 主要参数: 前四个参数分别是主标题、副标题、x 轴和 y 轴标题; line 设置一个距离图形边缘的距离 (lineGE 行
  ↵   高); outer 表示是否将文本放在外边界中
2 title(main = NULL, sub = NULL, xlab = NULL, ylab = N, line = NA, outer=FALSE,
  ↵   ...
3
4 # 主要参数: label 是添加的文本; pos 取值 1-4, 表示文本位置在坐标点的下、左、上、右方; offset 在会 pos
  ↵   基础上向相应方向再偏移一定比例的距离; vfont 是用 Hershey 矢量字体设置文本的字体式样
5 text(x, y = NULL, label= seq_along(x), adj = NULL, pos = NULL, offset = 0.5, vfont
  ↵   = NULL, cex = 1, col = NULL font = NULL, ...)
6
7 # 主要参数: side 取值为 1-4, 表示周边文本绘制在图形的下、左、上、右边
8 mtext(text, side = 3, line = 0, outer = FALSE, at = NA, adj = NA, padj = NA, cex =
  ↵   NA, col = NA, font = NA, ...)
```

82



# 绘图元素拆解

## 文本

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

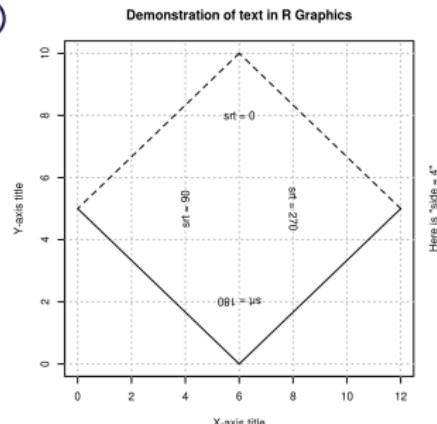
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

79



```
1 > par(mar = c(4, 4, 4, 3))
2 > plot(0:10, type = "n", xlab = "", ylab = "",
3       xlim = c(0,12))
4 > grid(col = "gray")
5 > title(main = "Demonstration of text in R
6       Graphics", xlab = "X-axis title", ylab =
7       "Y-axis title")
8 > mtext("Here is \\"side = 4\\\"", side = 4, line =
9       1)
10 > x = c(6, 4, 6, 8)
11 > y = c(8, 5, 2, 5)
12 > s = c(0, 90, 180, 270)
13 > for (i in 1:4) text(x[i], y[i], sprintf("srt =
14       %d", s[i]), srt = s[i])
15 > segments(c(6, 0, 6, 12), c(10, 5, 0, 5), c(0,
16       6,12, 6), c(5, 0, 5, 10), lty = c(2, 1, 1, 2))
```

图：文本函数示例

82



# 绘图元素拆解

## 图例

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

80

- 图例是统计图形中很重要的辅助解释信息，其作用将不同的对象分组为不同的样式
- R 中绘制图例的函数是 **legend()**

```
1 # 主要参数：前两个参数 x 和 y 表示图例的坐标位置（左上角顶点的坐标）；legend 通常为一个字符向量，表示图
  ↪ 例中的文字；fill 指定一个与图例字符向量对应的颜色向量用以在文本左边绘制一个颜色填充方块；col 设置
  ↪ 图例中点和线的颜色；lty、lwd 和 pch 指定图例中点线的样式；angle 和 density 效果类似于 fill 参
  ↪ 数，只是换成指定角度和密度的阴影线填充方块；bty 数设置图例框的样式；title 设定图例的标题
  ↪
2 legend(x, y = NULL, legend, fill = NULL, col = par("col"), border = "black", lty, lwd,
  ↪ pch, angle = 45, density = NULL, bty = "o", bg = par("bg"), box.lwd =
  ↪ par("lwd"), box.lty = par("lty"), box.col = par("fg"), pt.bg = NA, cex = 1,
  ↪ pt.cex = cex, pt.lwd = lwd, xjust = 0, yjust = 1, x.intersp = 1, y.intersp = 1,
  ↪ adj = c(0, 0.5), text.width = NULL, text.col = par("col"), merge = do.lies &&
  ↪ has.pch, trace = FALSE, plot = TRUE, ncol = 1, horiz = FALSE, title = NULL,
  ↪ inset = 0, xpd, title.col = text.col)
```

82



# 绘图元素拆解

## 图例

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

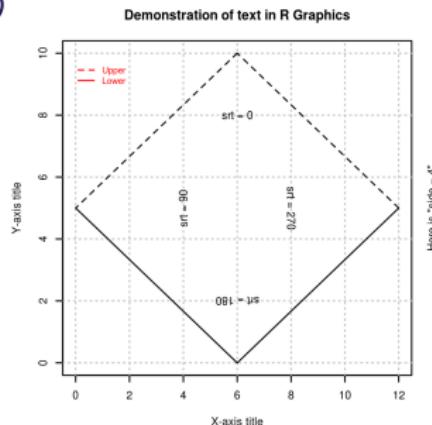
标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

80



82

```
1 > par(mar = c(4, 4, 4, 3))
2 > plot(0:10, type = "n", xlab = "", ylab = "",
3   +       xlim = c(0,12))
4 > grid(col = "gray")
5 > title(main = "Demonstration of text in R
6   +       Graphics",
7   +       xlab = "X-axis title", ylab = "Y-axis title")
8 > mtext("Here is `side = 4`", side = 4, line =
9   +       1)
10 > x = c(6, 4, 6, 8)
11 > y = c(8, 5, 2, 5)
12 > s = c(0, 90, 180, 270)
13 > for (i in 1:4) text(x[i], y[i], sprintf("srt =
14   +       %d",s[i]), srt = s[i])
15 > segments(c(6, 0, 6, 12), c(10, 5, 0, 5), c(0,
16   +       6,12, 6), c(5, 0, 5, 10), lty = c(2, 1, 1, 2))
17 > legend(-0.2, 9.8, c("Upper", "Lower"), lty =
18   +       2:1, cex = 0.8, bty = "n", text.col="red",
19   +       col="red")
```

图：文本函数示例



# 绘图元素拆解

## 坐标轴

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入

81

- 坐标轴是统计图形中元素数值大小的参照物
- R 中绘制图例的函数是 **axis()**

```
1 # 主要参数: side 参数与 mtext() 函数中的参数类似, 表示将坐标轴画在哪条边上; at 参数表示在什么位置画坐
  ↵ 标轴标记线; labels 参数指定坐标轴刻度标记的字符; tick 参数表示是否绘制刻度线
2 axis(side, at = NULL, labels = TRUE, tick = TRUE, line = NA, pos = NA, outer =
  ↵ FALSE, font = NA, lty = "solid", lwd = 1, lwd.ticks = lwd, col = NULL,
  ↵ col.ticks = NULL, hadj = NA, padj = NA, ...)
```

82



# 绘图元素拆解

## 坐标轴

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

绘图基础知识

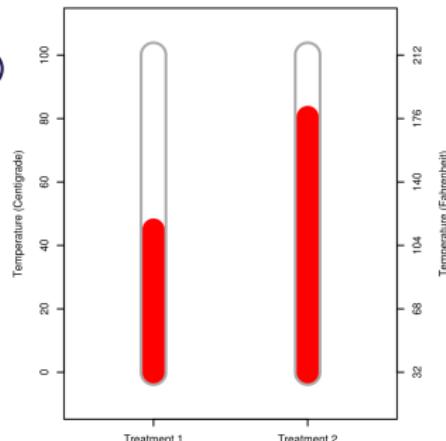
绘图元素拆解

标准绘图程序包

lattice 程序包

ggplot 程序包

R 图形混合嵌入



```
> x <- 1:2; y <- runif(2, 0, 100)
2 > plot(x, y, type="n", xlim=c(0.5, 2.5),
4   ylim=c(-10, 110), axes=FALSE,
4   ann=FALSE)
3 > axis(2, at=seq(0, 100, 20))
4 > mtext("Temperature (Centigrade)", side=2,
4   line=3)
# 绘制双坐标轴
6 > axis(1, at=1:2, labels=c("Treatment 1",
7   "Treatment 2"))
7 > axis(4, at=seq(0, 100, 20), labels=seq(0, 100,
8   20)*9/5 + 32)
9 > mtext("Temperature (Fahrenheit)", side=4,
9   line=3)
10 > box()
11 > segments(x, 0, x, 100, lwd=20, col="dark
12   grey")
12 > segments(x, 0, x, 100, lwd=16, col="white")
12 > segments(x, 0, x, y, lwd=16, col="red")
```

图：坐标轴函数示例（摄氏度双坐标轴）



# 目录

P&LRC

统计图形和 R

统计图形

统计绘图工具

R 简介

R 绘图

R 图形混合嵌入

82

① 统计图形

② 统计绘图工具

③ R 简介

④ R 绘图

⑤ R 图形混合嵌入

82

汇报完毕  
谢谢！

