Broadview
www.broadview.com.cn

R语言

数据可视化之美

专业图表绘制指南

张杰 / 著

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

内 容 简 介

本书主要介绍使用 R 中的 ggplot2 包及其拓展包绘制专业图表的方法。本书先介绍了 R 语言编程基础知识，以及使用 dplyr、tidyr、reshape2 等包的数据操作方法；再对比了 base、lattice 和 ggplot2 包的图形语法。本书首次系统性地介绍了使用 ggplot2 包及其拓展包绘制类别对比型、数据关系型、时间序列型、整体局部型等常见的二维图表的方法，以及使用 plot3D 包绘制三维图表（包括三维散点图、柱形图和曲面图等）的方法。另外，本书也首次介绍了论文中学术图表的图表配色、规范格式等相关技能与知识。

图书在版编目（CIP）数据

R 语言数据可视化之美：专业图表绘制指南 / 张杰著. —北京：电子工业出版社，2019.6
ISBN 978-7-121-36366-5

Ⅰ. ①R… Ⅱ. ①张… Ⅲ. ①统计分析－应用软件－指南 Ⅳ. ①C819-62

中国版本图书馆 CIP 数据核字（2019）第 070960 号

责任编辑：石　倩
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×980　1/16　印张：18.25　字数：461 千字
版　　次：2019 年 6 月第 1 版
印　　次：2019 年 6 月第 1 次印刷
定　　价：109.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。
质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。
本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

前　言

本书主要介绍使用 R 中的 ggplot2 包及其拓展包绘制专业图表的方法。本书先介绍了 R 语言编程基础知识，以及使用 dplyr、tidyr、reshape2 等包的数据操作方法；再对比了 base、lattice 和 ggplot2 包的图形语法。本书首次系统性地介绍了使用 ggplot2 包及其拓展包绘制类别对比型、数据关系型、时间序列型、整体局部型等常见的二维图表的方法，以及使用 plot3D 包绘制三维图表（包括三维散点图、柱形图和曲面图等）的方法。另外，本书也首次介绍了论文中学术图表的图表配色、规范格式等相关技能与知识。

本书定位

虽然现在 Python 语言越来越流行，尤其是在机器学习与深度学习等领域，但是 R 语言在数据分析与可视化方面仍然具有绝对的优势，其中 ggplot2 包及其拓展包人性化的绘图语法大受用户的喜爱，特别是生物信息与医学研究者。现在 *Nature*、*Science* 和 *Cell* 等期刊上大量的图表都是使用 R 语言绘制的，所以很有必要系统性地介绍 R 语言的绘图方法。

R ggplot2 有两本很经典的教程：*ggplot2 Elegant Graphics for Data Analysis* 和 *R Graphics Cookbook*，两书重点介绍了 ggplot2 包的绘图语法及常见图表的绘制方法，但是其介绍的图表种类并不多。所以本书基于 R 中的 ggplot2 包及其拓展包和 plot3D 包，系统性地介绍了几乎所有常见的二维和三维图表的绘制方法，包括简单的柱形图系列、条形图系列、折线图系列，以及复杂的和弦图、矩形树状图、日历图等。

读者对象

本书适用于想学习数据分析与可视化相关专业课程的高校学生，以及对数据分析与可视化感兴

趣的职场人士阅读，尤其是 R 语言用户。从软件掌握程度而言，本书同样适用于零基础学习 R 语言的用户。

阅读指南

全书内容共有 9 章，其中，第 1 章和第 2 章是后面 7 章的基础，第 3 ~ 8 章都是独立章节，可以根据实际需求有选择性地进行学习。

第 1 章　介绍 R 语言编程与数据可视化基础，对比了 base、lattice 和 ggplot2 包的图形语法，重点介绍了 ggplot2 包的图形语法；

第 2 章　介绍 R 语言数据处理基础，重点介绍了使用 dplyr、tidyr、reshape2 等包的数据操作方法；

第 3 章　介绍类别比较型图表，包括柱形图系列、条形图系列、南丁格尔玫瑰图、径向柱图等约 30 种图表；

第 4 章　介绍数据关系型图表，包括二维和三维散点图、气泡图、等高线图、三维曲面图、三元相图、二维和三维瀑布图、相关系数热力图等约 60 种图表；

第 5 章　介绍数据分布型图表，包括一维、二维和三维的统计直方图和核密度估计图、抖动散点图、点阵图、箱形图、小提琴图等约 50 种图表；

第 6 章　介绍时间序列型图表，包括折线图和面积图系列、日历图、螺旋图系列、量化波形图、地平线图等约 20 种图表；

第 7 章　介绍局部整体型图表，包括饼图、散点复合饼图系列、旭日图、矩形树状图、马赛克图、华夫饼图等约 20 种图表；

第 8 章　介绍高维数据的可视化方法，包括分面图系列、矩阵散点图、热力图、平行坐标系图、RadViz 图、图标法等约 20 种图表；

第 9 章　介绍论文中学术图表的常用技能，包括常见的截图与图片处理软件及其功能、矢量图片的修改、论文中学术图表数据的提取与重绘、论文中学术图表的规范与调整等。

应用范围

本书的图表绘制方法都是基于 R 中的 ggplot2 包及其拓展包和其他绘图包实现的，几乎适应于所有常见的二维和三维图表。但是由于依据《地图管理条例》第十五条规定：“国家实行地图审核制度。向社会公开的地图，应当报送有审核权的测绘地理信息行政主管部门审核。但是，景区图、街区图、地铁线路图等内容简单的地图除外。”本书本来有专门的章节讲解使用 R 语言如何绘制不同地理坐标

投影下，从世界到不同国家与区域的地图，但是由于地图审核周期等方面的原因忍痛移除。

适用版本

本书所用 R 版本为：3.3.3。R 作为开源免费的软件，数据分析与可视化的包更新迭代很快，这是它的优势。但是有时候有些代码运行可能会由于 R 或者 R 包版本的更新，而出现函数弃用（deprecated）的情况。此时，需要自己更新代码，使用新的函数替代原有的函数等。

源代码

本书配备有几乎所有图表的 R 语言源文件及其.csv 或.txt 格式的数据源文件。但是需要注意的是，如果运行的 R 语言版本没有安装相应的数据分析与可视化的包（package），那么请预先安装相应的包，才能成功运行代码。同时，也请注意运行 R 语言和 R 包的版本是否已经更新。

与我联系

因本人知识与能力所限，书中纰漏之处在所难免，欢迎并恳请读者朋友们给予批评与指正，可以通过邮箱联系笔者本人；如果读者有关于 R 语言学术图表或商业图表绘制的问题，可以联系笔者交流。另外，更多关于 R 语言图表绘制的教程请关注笔者的博客、专栏和微博平台。也可以重点关注我们的微信公众号：EasyCharts，也可以添加笔者微信：EasyCharts。R 语言数据分析与可视化的文章会优先发表在微信公众号平台。


邮　　箱：easycharts@qq.com


知乎专栏：https://zhuanlan.zhihu.com/EasyCharts-R（知乎账号：EasyCharts）


博　　客：http://easychart.github.io/


新浪微博：https://weibo.com/easycharts?source=blog&is_all=1（微博账号：EasyCharts）

致谢

桃李春风一杯酒，江湖夜雨十年灯。笔者的处女作《Excel 数据之美：科学图表与商业图表的绘制》也至今出版逾两年，一直想着要修订这本书的。但是旧书未翻新，新书忙于码字改稿，实在是有愧于读者。其实，在撰写这本新书的时候，数次想放弃。写书实在是一件费力劳神的事情，笔者是凭借着对数据可视化的热爱才坚持至今。

这本书从 2017 年 5 月 25 日开始动笔，断断续续居然也花费了两年的时间。与其说是花费，不如说是陪伴吧。笔者经常对朋友开玩笑说，心情不好的时候码码代码、画画图表，是一件消磨时间、

放松心情的事情。

在断断续续的写稿中，笔者也认识了很多热爱数据分析与可视化的朋友，甚是荣幸，也得益于他们的帮助。很感谢《R 语言游戏数据分析与挖掘》的作者谢佳标老师和先锋信息科技有限公司 CEO 林祯舜老师对笔者的鼓励与帮助，也因此有幸参加 2018 年的 R 语言大会；也非常感谢在码字、写代码时一起交流学习的李誉辉（四川大学高分子学院）、杜雨（美团用户平台—大数据与算法部—商业分组部）、刘钰（河南大学土木建筑学院）、厚缁（深圳中观经济咨询有限公司）等诸多技术大佬。因为有你们的帮助，所以才有今天这本书。

最后，想对大家说，也是对自己说：且将新火试新茶，诗酒趁年华！

作　者

2019 年 3 月 31 日

读 者 服 务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **下载资源**：本书如提供示例代码及资源文件，均可在 下载资源 处下载。
- **提交勘误**：您对书中内容的修改意见可在 提交勘误 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方 读者评论 处留下您的疑问或观点，与我们和其他读者一同学习交流。

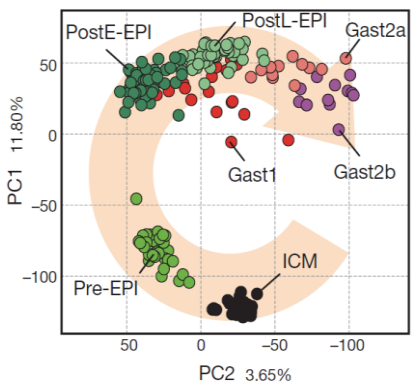
页面入口：http://www.broadview.com.cn/36366

第 1 章

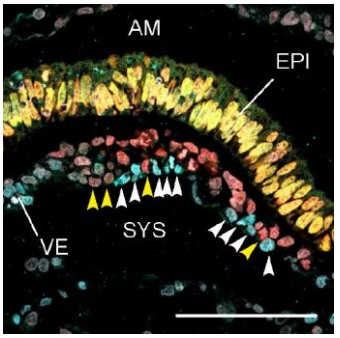
R 语言编程与绘图基础

1.1 学术图表的基本概念

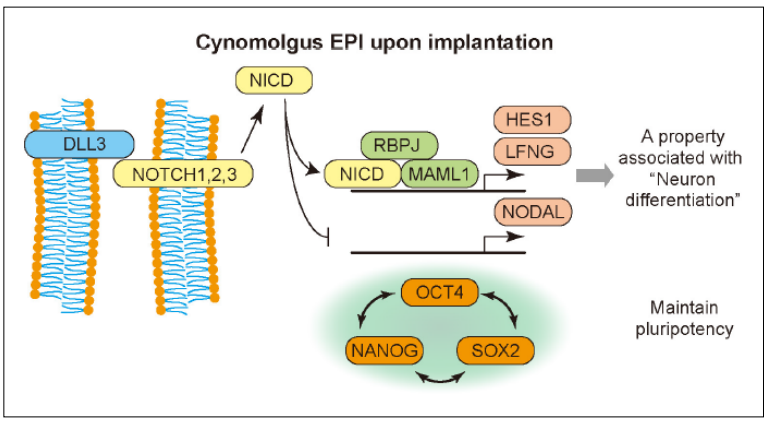
学术图表是为论文结论（conclusion）提供证据的视觉方式。所以，论文作者为了产生强烈的视觉效果，应该通过分析实验数据，精心设计可视化图表。本书开篇先跟大家讲讲学术图表的类型。通常学术论文中主要有三类图表，如图 1-1-1 所示。流程示意图和数据展示图都是非常讲究技能的图表，本书重点讲解的是数据展示图。



(a) 数据展示图



(b) 实验拍摄图



(c) 流程示意图

图 1-1-1 不同类型的图表 [1]

1. 数据展示图：先根据数据绘制成图表，再将其导出生成图片，主要包括各种点线图、柱形图、饼图等统计图表，一般使用 Excel、GraphPad Prism、SigmaPlot、Origin、MATLAB、Python、R 等专业绘图软件绘制（Excel 并非如大众所说不能导出高分辨率的图片和矢量图）。注意，保存图片时，

一定要保存成高分辨率的 TIFF 格式和 EPS 矢量格式的图片，因为矢量图片是可以使用图片处理软件进行再编辑的。由数据生成的图表是可重复修改的，因此一定要保存好原始数据，一旦发现图表有任何问题就可以进行修改。

2. **实验拍摄图**：使用设备或者仪器拍摄采集的图片，包括显微镜、扫描仪及摄像机等所拍照片。一定要在最刚开始时就拍成高清的（设置成高分辨率），也就是要保证原始图片的高分辨率，接下来处理图片就会比较方便，免得因为图片质量不佳而重复实验。必要的话，把每张图片存储成 TIFF 和 JPG 两种格式（以应对部分期刊的特殊要求）。

3. **流程示意图**：使用简明的线条、基本图形和箭头等绘制论文中的重要的实验流程或步骤，用以说明基本原理或解释文字材料，一般使用 PPT、Visio、Illustrator、CorelDRAW、3DMax 等软件绘制。

1.1.1　学术图表的基本作用

图表在学术论文中是很重要的一部分。实验结果通常是论文的核心和主要部分，而实验结果一般以图表的形式呈现。读者经常通过图表来判断这篇文章是否值得阅读，所以每个图表都应该能不依赖正文而独立存在。所谓“一图抵千言”（A picture is worth a thousand words）。图表设计是否精确且合理直接影响数据的完整与准确表达，从而影响论文的质量。图表是期刊评审过程中仅次于摘要的关键一环，准确而美观的图表能促进审稿人和读者对论文表达的快速理解。以 *Nature* 上的文章 *Cotranslational signal-independent SRP preloading during membrane targeting* [2]选取的前两页为例（见图 1-1-2），我们首先关注的是论文的标题（title），其次是第一页最开始的摘要（abstract），接下来我们就被这些包含大量实验数据与信息的图表所吸引。在每页的文章中，包含图名（figure）的图表部分几乎占据整个页面的 1/4~1/3，由此可见图表在论文中的重要性。

根据 Edward R. Tufte 的 *The Visual Display of Quantitative Information*[3]和 *Visual Explanations*[4]的阐述，图表在论文的作用主要有：

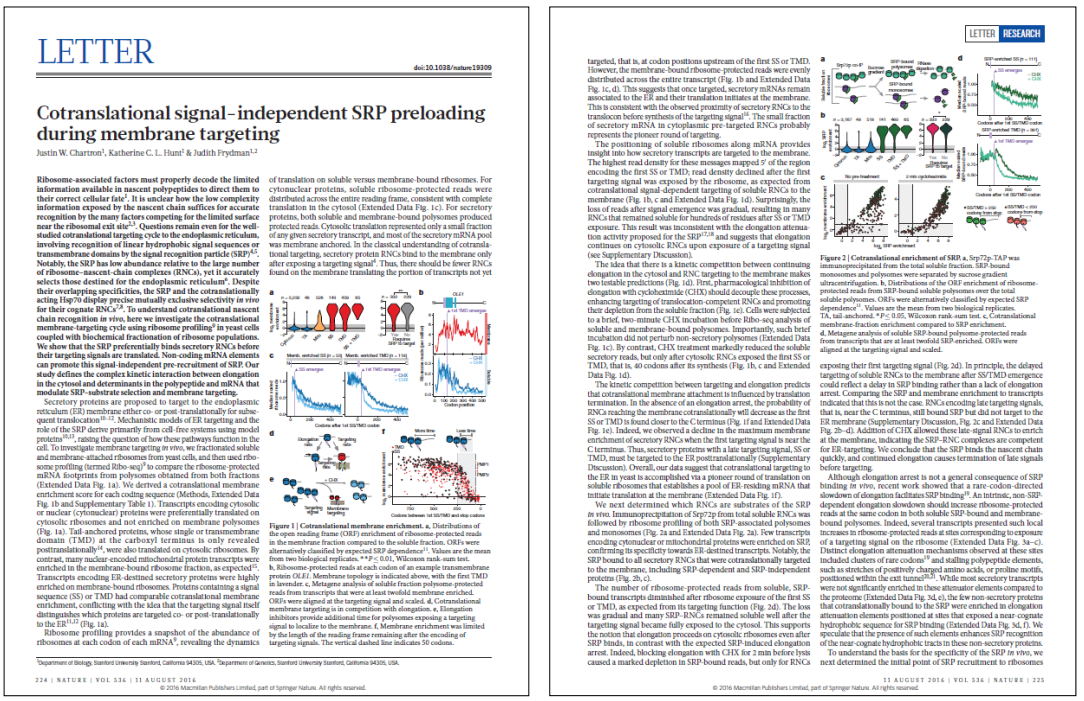
（1）真实、准确、全面地展示数据；

（2）以较小的空间承载较多的信息；

（3）揭示数据的本质、关系、规律。

第三点作用尤为重要，Matthew O. Ward 也提出，可视化的终极目标是洞悉蕴含在数据中的现象和规律，这包括多重含义：发现、决策、解释、分析、探索和学习[5]。表 1-1-1 所示的原始数据是 31 组 *x*-*y* 的二维数据。仅仅只从数据的角度去观察数据，就很难发现 *x* 与 *y* 之间的具体关系。将实际的数据分布情况使用二维可视化的方法呈现，如图 1-1-3 所示，则可以快速地从数据中发现数据内在的

模式与规律。所以，有时使用数据可视化的方法也可以很好地帮助我们去分析数据。



LETTER

doi:10.1038/nature19309

Cotranslational signal-independent SRP preloading during membrane targeting

Justin W. Chartron[1], Katherine C. L. Hunt[1] & Judith Frydman[1,2]

Ribosome-associated factors must properly decode the limited information available in nascent polypeptides to direct them to their correct cellular fate[1]. It is unclear how the low complexity information exposed by the nascent chain suffices for accurate recognition by the many factors competing for the limited surface near the ribosomal exit site[2,3]. Questions remain even for the well-studied cotranslational targeting cycle to the endoplasmic reticulum, involving recognition of linear hydrophobic signal sequences or transmembrane domains by the signal recognition particle (SRP)[4,5]. Notably, the SRP has low abundance relative to the large number of ribosome–nascent-chain complexes (RNCs), yet it accurately selects those destined for the endoplasmic reticulum[6]. Despite their overlapping specificities, the SRP and the cotranslationally acting Hsp70 display precise mutually exclusive selectivity *in vivo* for their cognate RNCs[7,8]. To understand cotranslational nascent chain recognition *in vivo*, here we investigate the cotranslational membrane-targeting cycle using ribosome profiling[9] in yeast cells coupled with biochemical fractionation of ribosome populations. We show that the SRP preferentially binds secretory RNCs before their targeting signals are translated. Non-coding mRNA elements can promote this signal-independent pre-recruitment of SRP. Our study defines the complex kinetic interaction between elongation in the cytosol and determinants in the polypeptide and mRNA that modulate SRP–substrate selection and membrane targeting.

Secretory proteins are proposed to target to the endoplasmic reticulum (ER) membrane either co- or post-translationally for subsequent translocation[10–12]. Mechanistic models of ER targeting and the role of the SRP derive primarily from cell-free systems using model proteins[10,13], raising the question of how these pathways function in the cell. To investigate membrane targeting *in vivo*, we fractionated soluble and membrane-attached ribosomes from yeast cells, and then used ribosome profiling (termed Ribo-seq)[9] to compare the ribosome-protected mRNA footprints from polysomes obtained from both fractions (Extended Data Fig. 1a). We derived a cotranslational membrane enrichment score for each coding sequence (Methods, Extended Data Fig. 1b and Supplementary Table 1). Transcripts encoding cytosolic or nuclear (cytonuclear) proteins were preferentially translated on cytosolic ribosomes and not enriched on membrane polysomes (Fig. 1a). Tail-anchored proteins, whose single or transmembrane domain (TMD) at the carboxyl terminus is only revealed posttranslationally[14], were also translated on cytosolic ribosomes. By contrast, many nuclear-encoded mitochondrial protein transcripts were enriched in the membrane-bound ribosome fraction, as expected[15]. Transcripts encoding ER-destined secretory proteins were highly enriched on membrane-bound ribosomes. Proteins containing a signal sequence (SS) or TMD had comparable cotranslational membrane enrichment, conflicting with the idea that the targeting signal itself distinguishes which proteins are targeted co- or post-translationally to the ER[11,12] (Fig. 1a).

Ribosome profiling provides a snapshot of the abundance of ribosomes at each codon of each mRNA[9], revealing the dynamics of translation on soluble versus membrane-bound ribosomes. For cytonuclear proteins, soluble ribosome-protected reads were distributed across the entire reading frame, consistent with complete translation in the cytosol (Extended Data Fig. 1c). For secretory proteins, both soluble and membrane-bound polysomes produced protected reads. Cytosolic translation represented only a small fraction of any given secretory transcript, and most of the secretory mRNA pool was membrane anchored. In the classical understanding of cotranslational targeting, secretory protein RNCs bind to the membrane only after exposing a targeting signal[6]. Thus, there should be fewer RNCs found on the membrane translating the portion of transcripts not yet

Figure 1 | Cotranslational membrane enrichment. a, Distributions of the open reading frame (ORF) enrichment of ribosome-protected reads in the membrane fraction compared to the soluble fraction. ORFs were alternatively classified by expected SRP dependence[11]. Values are the mean from two biological replicates. ***P < 0.01, Wilcoxon rank-sum test. b, Ribosome-protected reads at each codon of an example transmembrane protein *OLE1*. Membrane topology is indicated above, with the first TMD in lavender. c, Metagene analysis of soluble fraction polysome-protected reads from transcripts that were at least twofold membrane enriched. ORFs were aligned at the targeting signal and scaled. d, Cotranslational membrane targeting is in competition with elongation. e, Elongation inhibitors provide additional time for polysomes exposing a targeting signal to localize to the membrane. f, Membrane enrichment was limited by the length of the reading frame remaining after the encoding of targeting signals. The vertical dashed line indicates 50 codons.

[1]Department of Biology, Stanford University Stanford, California 94305, USA. [2]Department of Genetics, Stanford University Stanford, California 94305, USA.

224 | NATURE | VOL 536 | 11 AUGUST 2016

LETTER RESEARCH

targeted, that is, at codon positions upstream of the first SS or TMD. However, the membrane-bound ribosome-protected reads were evenly distributed across the entire transcript (Fig. 1b and Extended Data Fig. 1c, d). This suggests that once targeted, secretory mRNAs remain associated to the ER and their translation initiates at the membrane. This is consistent with the observed proximity of secretory RNCs to the translocon before synthesis of the targeting signal[16]. The small fraction of secretory mRNA in cytoplasmic pre-targeted RNCs probably represents the pioneer round of targeting.

The positioning of soluble ribosomes along mRNA provides insight into how secretory transcripts are targeted to the membrane. The highest read density for these messages mapped 5′ of the region encoding the first SS or TMD; read density declined after the first targeting signal was exposed by the ribosome, as expected from cotranslational signal-dependent targeting of soluble RNCs to the membrane (Fig. 1b, c and Extended Data Fig. 1d). Surprisingly, the loss of reads after signal emergence was gradual, resulting in many RNCs that remained soluble for hundreds of residues after SS or TMD exposure. This result was inconsistent with the elongation attenuation activity proposed for the SRP[17,18] and suggests that elongation continues on cytosolic RNCs upon exposure of a targeting signal (see Supplementary Discussion).

The idea that there is a kinetic competition between continuing elongation in the cytosol and RNC targeting to the membrane makes two testable predictions (Fig. 1d). First, pharmacological inhibition of elongation with cycloheximide (CHX) should decouple these processes, enhancing targeting of translocation-competent RNCs and promoting their depletion from the soluble fraction (Fig. 1e). Cells were subjected to a brief, two-minute CHX incubation before Ribo-seq analysis of soluble and membrane-bound polysomes. Importantly, such brief incubation did not perturb non-secretory polysomes (Extended Data Fig. 1e). By contrast, CHX treatment markedly reduced the soluble secretory reads, but only after cytosolic RNCs exposed the first SS or TMD, that is, 40 codons after its synthesis (Fig. 1b, c and Extended Data Fig. 1d).

The kinetic competition between targeting and elongation predicts that cotranslational membrane attachment is influenced by translation termination. In the absence of an elongation arrest, the probability of RNCs reaching the membrane cotranslationally will decrease as the first SS or TMD is found closer to the C terminus (Fig. 1f and Extended Data Fig. 1e). Indeed, we observed a decline in the maximum membrane enrichment of secretory RNCs when the first targeting signal is near the C terminus. Thus, secretory proteins with a late targeting signal, SS or TMD, must be targeted to the ER posttranslationally (Supplementary Discussion). Overall, our data suggest that cotranslational targeting to the ER in yeast is accomplished via a pioneer round of translation on soluble ribosomes that establishes a pool of ER-residing mRNA that initiate translation at the membrane (Extended Data Fig. 1f).

We next determined which RNCs are substrates of the SRP *in vivo*. Immunoprecipitation of Srp72p from total soluble RNCs was followed by ribosome profiling of both SRP-associated polysomes and monosomes (Fig. 2a and Extended Data Fig. 2a). Few transcripts encoding cytonuclear or mitochondrial proteins were enriched on SRP, confirming its specificity towards ER-destined transcripts. Notably, the SRP bound to all secretory RNCs that were cotranslationally targeted to the membrane, including SRP-dependent and SRP-independent proteins (Fig. 2b, c).

The number of ribosome-protected reads from soluble, SRP-bound transcripts diminished after ribosome exposure of the first SS or TMD, as expected from its targeting function (Fig. 2d). The loss was gradual and many SRP–RNCs remained soluble well after the targeting signal became fully exposed to the cytosol. This supports the notion that elongation proceeds on cytosolic ribosomes even after SRP binds, in contrast with the expected SRP-induced elongation arrest. Indeed, blocking elongation with CHX for 2 min before lysis caused a marked depletion in SRP-bound reads, but only for RNCs exposing their first targeting signal (Fig. 2d). In principle, the delayed targeting of soluble RNCs to the membrane after SS/TMD emergence could reflect a delay in SRP binding rather than a lack of elongation arrest. Comparing the SRP and membrane enrichment to transcripts indicated that this is not the case. RNCs encoding late targeting signals, that is, near the C terminus, still bound SRP but did not target to the ER membrane (Supplementary Discussion, Fig. 2c and Extended Data Fig. 2b–d). Addition of CHX allowed these late-signal RNCs to enrich at the membrane, indicating the SRP–RNC complexes are competent for ER-targeting. We conclude that the SRP binds the nascent chain quickly, and continued elongation causes termination of late signals before targeting.

Although elongation arrest is not a general consequence of SRP binding *in vivo*, recent work showed that a rare-codon-directed slowdown of elongation facilitates SRP binding[19]. An intrinsic, non-SRP-dependent elongation slowdown should increase ribosome-protected reads at the same codon in both soluble SRP-bound and membrane-bound polysomes. Indeed, several transcripts presented such local increases in ribosome-protected reads at sites corresponding to exposure of a targeting signal on the ribosome (Extended Data Fig. 3a–c). Distinct elongation attenuation mechanisms observed at these sites included clusters of rare codons[19] and stalling polypeptide elements, such as stretches of positively charged amino acids, or proline motifs, positioned within the exit tunnel[20,21]. While most secretory transcripts were not significantly enriched in these attenuator elements compared to the proteome (Extended Data Fig. 3d, e), the few non-secretory proteins that cotranslationally bound to the SRP were enriched in elongation attenuation elements positioned at sites that exposed a near-cognate hydrophobic sequence for SRP binding (Extended Data Fig. 3d, f). We speculate that the presence of such elements enhances SRP recognition of the near-cognate hydrophobic tracts in these non-secretory proteins.

To understand the basis for the specificity of the SRP *in vivo*, we next determined the initial point of SRP recruitment to ribosomes

Figure 2 | Cotranslational enrichment of SRP. a, Srp72p-TAP was immunoprecipitated from the total soluble fraction. SRP-bound monosomes and polysomes were separated by sucrose-gradient ultracentrifugation. b, Distributions of the ORF enrichment of ribosome-protected reads from SRP-bound soluble polysomes over the total soluble polysomes. ORFs were alternatively classified by expected SRP dependence[11]. Values are the mean from two biological replicates. TA, tail-anchored. *P < 0.05, Wilcoxon rank-sum test. c, Cotranslational membrane-fraction enrichment compared to SRP enrichment. d, Metagene analysis of soluble SRP-bound polysome-protected reads from transcripts that are at least twofold SRP-enriched. ORFs were aligned at the targeting signal and scaled.

11 AUGUST 2016 | VOL 536 | NATURE | 225

(a) (b)

图 1-1-2 论文摘取的页面案例[2].

表 1-1-1 四组二维数据点集（相同的 *x* 变量，不同的 *y* 变量：*y*1，*y*2，*y*3，*y*4）

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
*y*1	4.6	5.4	5.2	6.6	5.9	6.1	5.8	6.8	6.5	6.7	6.9	11.1	8.2	10.3	12.8	13
*y*2	6.1	11.6	16.6	19	22.7	31.8	34	33.7	35.6	34.5	39.6	58.3	57.7	72.9	68.4	82.6
*y*3	5.5	31.1	33.1	51.8	55.7	60.7	63.5	75.5	84.4	84.6	76.3	92.4	81.6	91	88.1	93.8
*y*4	1	3	4.9	7.9	9.8	12	18.9	24.7	28.9	28.6	39.3	33.2	42.1	54.4	43.3	90.2

x	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	-
*y*1	20.8	12.4	15.9	15.3	38.8	35.9	24.3	54.5	62.9	43.8	76.9	91	96.9	51.4	100	-
*y*2	84.5	82	89.1	102.1	68.1	96.3	108.5	76.7	107.6	103.4	116.5	106.4	142.5	115.1	110.5	-
*y*3	101.3	103	107.4	104.3	110.7	103.4	113.6	105.1	112.5	119.3	113.7	109.5	108.7	110.1	118.8	-
*y*4	81.2	90.8	70.9	66.8	67.5	88.6	116.9	141.4	104	161.4	101.8	137.1	175.3	119.5	257.3	-

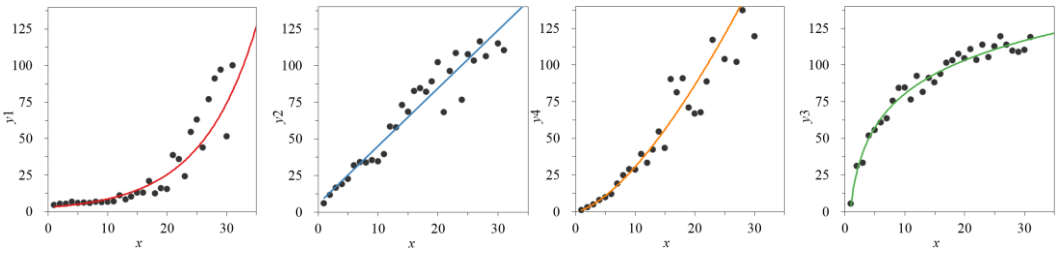
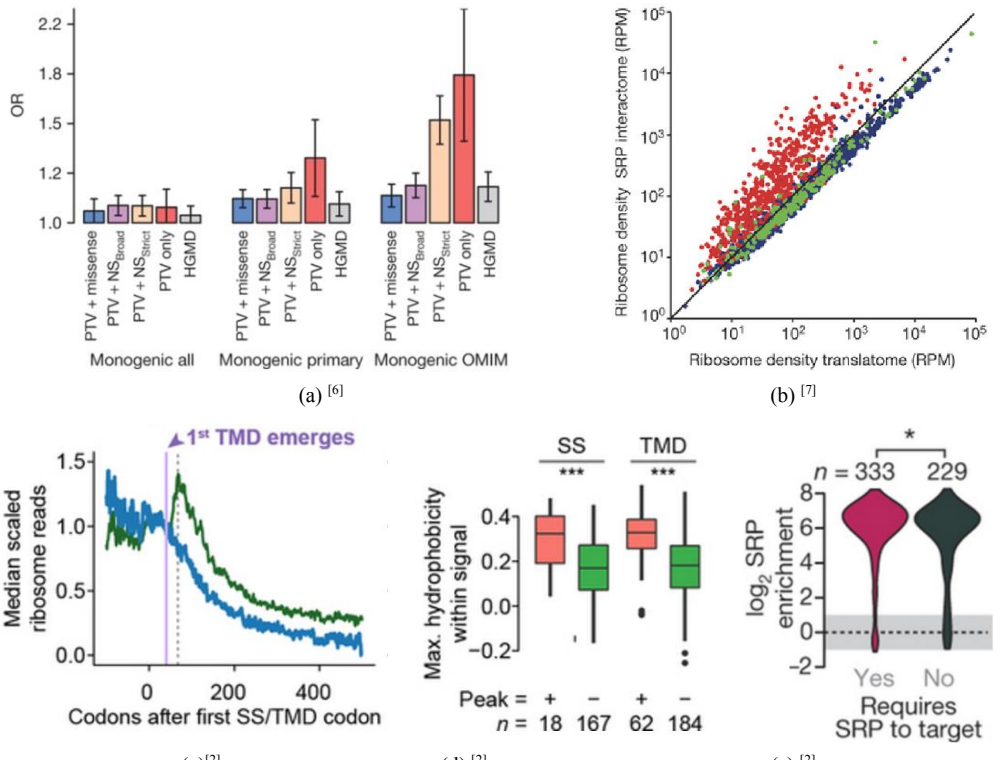


图 1-1-3　四个不同规律的二维数据点集的可视化案例

1.1.2　学术图表的基本类别

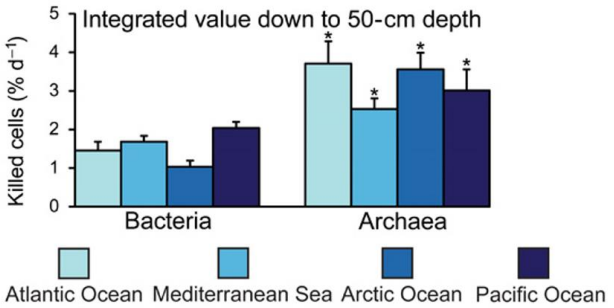
可以先通过国际顶级期刊的学术图表，如 *Science*、*Nature*、*Cell* 等（见图 1-1-4 和图 1-1-5），了解优秀学术图表的基本类型与风格。图表从色彩运用的角度可以分成两大类：彩色图表与黑白图表。



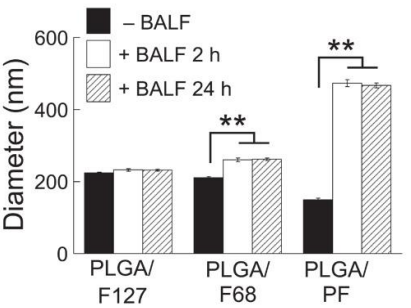
(a) [6]　(b) [7]

(c)[2]　(d) [2]　(e) [2]

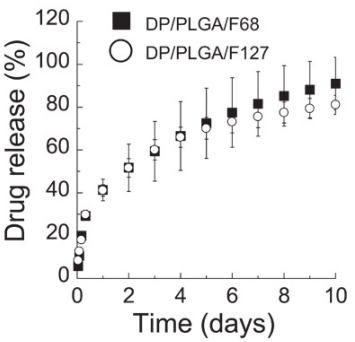
图 1-1-4　*Nature* 期刊的图表案例



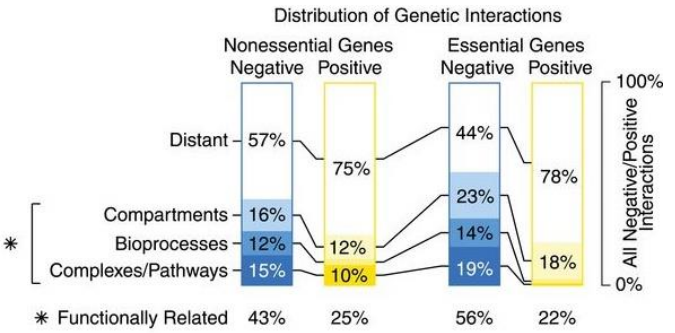
(a) [8]



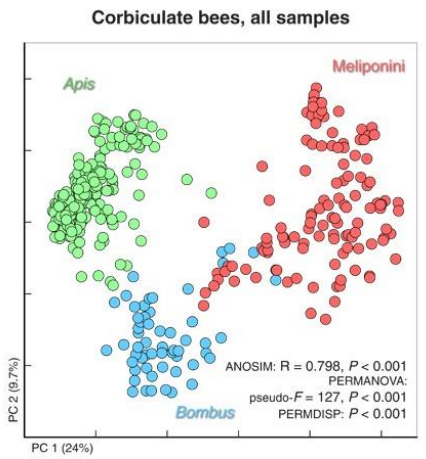
(b) [9]



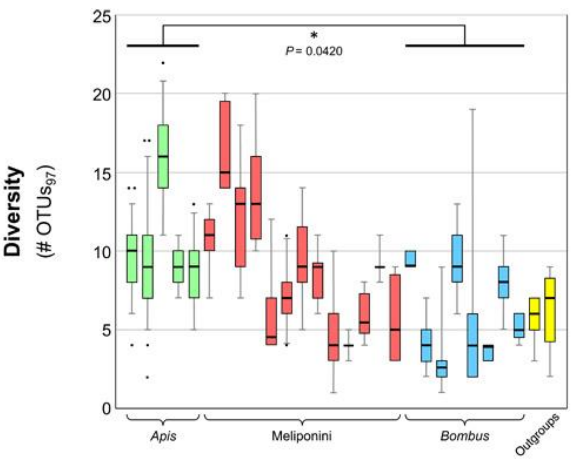
(c) [9]



(d) [10]



(e) [11]



(f) [11]

图 1-1-5 *Science* 期刊的图表案例

1. 黑白图表

由于彩色印刷的成本相对较高，所以大部分期刊是非彩色的，期刊也往往要求投稿的学术图表为黑白颜色，如图 1-1-4(b)和图 1-1-4(c)所示。如果论文中使用的都是彩色图表，有些期刊可能会在出版时向作者收取额外的彩色出版费用。在黑白图表中，数据系列的区分主要体现在数据标记上，可使用不同的填充纹理（见图 1-1-4(b)）或不同的填充颜色和标记形状（见图 1-1-4(c)）。

2. 彩色图表

随着互联网的发展，现在越来越多的文章会预先在网上发布（publish online），而且越来越多的读者与审稿人都喜欢阅读 PDF 形式的文章，这也导致越来越多的期刊接受彩色图表。彩色图表往往比黑白图表更加美观，从而更加吸引读者与审稿人。有时在只借助纹理、形状等无法准确而全面地展示数据，就只能用颜色来丰富数据的表达，如图 1-1-5(b)所示，由于不同数据系列的数据量多而密集，如果使用形状（如菱形◇、圆心○、方形□、三角形△等）区分数据系列，就很难清晰地展示数据的分布规律。

国内期刊一般以黑白印刷为主，绘图时需要注意采用不同的线型、标记等对不同曲线进行区分；国外的期刊相对而言以彩色印刷为主，但需要注意颜色的搭配。

1.1.3　学术图表的绘制原则

每个学术期刊都有自己对学术图表的基本要求，具体可以参考投稿期刊的《作者投稿指南》或 *Author Guidelines*、*Author Instructions*。以 *Nature* 期刊为例，作者的投稿主页（submit manuscript）如图 1-1-6 所示[1]，然后点击 instructions for authors，就可以进入作者的投稿指南，其中就有对图表（figure）的要求，包括基本图表要求（general figure guideline）和终稿图表要求（final figure submission guideline）两个部分。

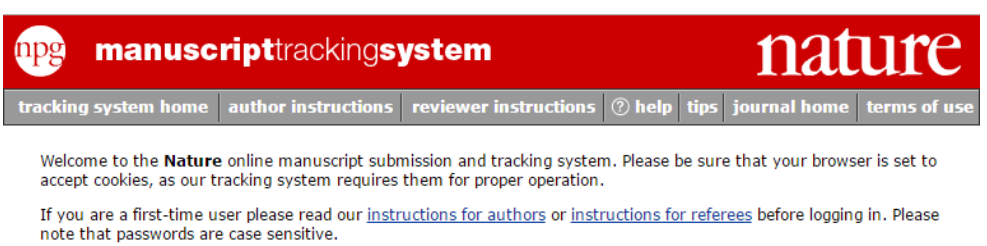


图 1-1-6　*Nature* 投稿主页页首

1 *Nature* 投稿主页: http://mts-nature.nature.com/cgi-bin/main.plex

所以，学术图表首先要规范，符合期刊的投稿要求，然后在规范的基础上实现图表的美观和专业。在当前贯彻科技论文规范化、标准化的同时，图表的设计也应规范化、标准化。总而言之，学术图表的制作原则主要是规范、简洁、专业和美观。

1. **规范**：规范就是指学术图表符合投稿期刊的图表格式和分辨率方面的要求，这是绘制图表的一个基础条件。绘图时满足投稿期刊的图表要求，这样至少能满足期刊编辑的要求，不会立即被退稿、被要求修改图表格式，例如图表的单位、字体、坐标、图例、轴名等。另外，期刊还会要求图表的分辨率和格式，一般要求 RGB 彩色图片的分辨率为 300dpi 及以上。

2. **简洁**：学术图表的关键在于清楚地表达数据信息。Robert A. Day 在 *How to write and publish a scientific paper* [12]书中指出：Combined or not, each graph should be as simple as possible（如果一张学术图表包含的数据信息太多，反而让读者难以理解自己所要表达的数据信息）。所以，学术图表应尽量简洁、清楚地表达数据信息。考虑到期刊的印刷成本，学术图表的尺寸也要尽量以较小的空间承载较多的信息，但也不要太小到无法看清图表的文字。

3. **专业**：图表类型的选择是做好图表的重要基础。专业就是指图表要能全面地反映数据的相关信息。当我们获得足够的实验数据后，需要重点思考的就是选择哪种图表能更加全面地表达数据信息。比如，同样是多次重复实验获得的数据，带误差线的散点图、带误差线的柱形图、箱形图等图表类型的选择就是我们要重点考虑的问题。

4. **美观**：图表美观是做好图表的一个重要条件。美观是指学术图表要简洁且具有美感。图表的配色、构图和比例等是影响图表美观的主要因素。但是由于大部分理工科的学生平时缺乏审美能力的训练，所以这也是许多学术图表缺乏美感的主要原因。

1.2 你为什么要选择 R

“工欲善其事，必先利其器”，学术绘图软件的选择与使用特别重要。不同学科的研究人员使用的软件有所不同，但是基础的绘图思想与理念是相通的（这部分会在后面的章节讲解）。具有工科背景的人员常使用 MATLAB，具有计算机背景的人员常使用 Python，具有统计学科背景的人员常使用 R，具有医学背景的人员常使用 GraphPad Prism 等。常用的学术图表绘制软件包括 Excel、Origin、SigmPlot、GraphPad Prism、MATLAB、Python、R 等，如图 1-2-1 所示。每个绘图软件的图表都有不同的图表风格。

图 1-2-1　绘图软件的标签云

笔者列出了常用的 7 款学术图表绘图软件，如表 1-2-1 所示。从技能要求的角度主要可以分为两大类：编程与界面操作。

表 1-2-1　常用绘图软件的性能对比

LOGO	名称	开源	付费	技能要求	官方网站
	Excel	否	是	界面操作	https://support.office.com/en-GB/Excel
	Origin	否	是	界面操作	http://originlab.com/
	SigmPlot	否	是	界面操作	https://systatsoftware.com/products/sigmaplot/
	GraphPad Prism	否	是	界面操作	http://www.graphpad.com/
	MATLAB	否	是	编程	https://www.mathworks.com/products/matlab.html
	Python	是	否	编程	https://www.python.org/
	R	是	否	编程	https://www.r-project.org/

像 Excel、Origin、SigmaPlot、GraphPad Prism 这 4 款软件，就不需要编程，只要点击界面按钮就可以绘制图表。尽管这些工具都非常容易使用，但也存在一些缺憾。只需鼠标操作无疑十分便捷，但随之而来的却是丧失一些灵活性。你可以改变颜色、字体和标题，但仅限于软件所提供的那些元素。这些软件只能由你去适应它的操作规则，让你使用现有的图表，而并不能创造新的图表。

像 MATLAB、Python 和 R 这 3 款软件，则需要编程才能实现图表的绘制。这些软件本身包含很多数据可视化的函数（function）或者包（package），供用户绘图时使用。尤其针对不同的数据集需

要重复操作的情况，如果使用界面绘图软件，则可能需要从头到尾将绘图流程重新实现一遍，而相比之下，通过代码来处理数据就会更加容易，因为针对不同的数据集只需稍微改动一下代码就可以解决。如果你充分掌握代码与算法，那你也可以自己编写函数设计新颖的图表。

1．R

相较于其他的所有软件，R 的优势之一在于，它是专为数据分析而设计的，它是主要用于统计分析、绘图的语言和操作环境。R 是属于 GNU 系统的一个自由、免费、源代码开放的软件，它是一个用于统计计算和统计制图的优秀工具。R 语言有一系列的数据可视化包，包括 ggplot2[1]及 ggplot2 拓展包[2]、lattice、leaflet、playwith、ggvis、ggmaps。

R 还提供了部分地图绘制功能，地区数据分析[3]提供了有关地区分析的综合性 R 工具包列表。另外，用户可以下载《地理统计制图实用指南》[4]——关于如何使用 R 及其他工具分析空间数据的可免费下载的电子书。

2．Python

Python 是一种面向对象的解释型计算机程序设计语言。Python 具有丰富和强大的库。它常被昵称为“胶水语言”，能够把用其他语言制作的各种模块（尤其是 C/C++）很轻松地联结在一起。程序员们戏称“人生苦短，要学 Python”，现在 Python 越来越流行，尤其应用在机器学习、机器视觉、深度学习、网络爬虫等方面。Python 语言也有一系列的数据可视化包，包括 Pandas、Plotnine[5]、matplotlib、Seaborn[6]、ggplot、Bokeh、Pygal 等[7]。其中 Plotnine 包是参考 R ggplot2 图形语法实现的可视化包。虽然 Python 越来越流行，但是在数据可视化方面与 R 还是有很大差距的。

3．MATLAB

MATLAB 是美国 MathWorks 公司出品的商业数学软件，用于算法开发、数据可视化、数据分析，以及数值计算的高级技术计算语言和交互式环境。MATLAB 可以进行矩阵运算、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序等，主要应用于工程计算、控制设计、信号处

1 ggplot2 包的官网：http://docs.ggplot2.org/current

2 ggplot2 extensions 拓展包的官网：http://www.ggplot2-exts.org/index.html

3 http://cran.r-project.org/web/views/Spatial.html

4 http://spatial-analyst.net/book/download

5 Plotnine 包的官网：https://plotnine.readthedocs.io/en/stable

6 Seaborn 包的官网：https://seaborn.pydata.org/

7 Python 语言的数据可视化包：http://pbpython.com/visualization-tools-1.html

理与通信、图像处理、信号检测、金融建模设计与分析等领域。MATLAB 软件本身就提供了很多绘图函数，可以满足数据可视化的基本需求[1]。但是还有另外两款 MATLAB 绘图包很值得推荐使用：PlotPub[2]和 Gramm[3]，其中，Gramm 包是在 MATLAB 中实现了 R ggplot2 的绘图风格，大大提高了 MATLAB 绘图的美观程度。

4．SigmaPlot

SigmaPlot 是一款最佳的科学绘图软件！使用 SigmaPlot 画出精密的图形是件极容易的事，目前已有超过十万的使用者，特别适合科学家使用。本软件允许用户自行建立任何所需的图形，可插入多条水平轴或垂直轴，指定误差棒（error bar）的方向，让你的图更光彩耀眼，只要用 SigmaPlot 将图制作完成即可动态连接给其他软件展示使用，并可输出成 EPS、TIFF、JPEG 等图形格式，或放置于网站上以供浏览。非常适合网站动态显示图形，使用场合如长时间记录的气象、温度等。

5．Origin

Origin 为 OriginLab 公司出品的较流行的专业函数绘图软件，是公认的简单易学、操作灵活、功能强大的软件，既可以满足一般用户的制图需要，也可以满足高级用户数据分析、函数拟合的需要。Origin 自 1991 年问世以来，由于其操作简便、功能开放，很快就成为国际流行的分析软件之一，是公认的快速、灵活、易学的制图软件。Origin 2017 版本增加了许多颜色主题方案，可以大大改进图表的美观程度。

6．GraphPad Prism

GraphPad Prism 是一款集数据分析和作图为一体的数据处理软件，尤其适合生物医学类，可以直接输入原始数据获得高质量的科学图表。它在统计分析上劣于 SPSS 等统计软件，但是不需要输入程序语言，只需输入原始数据，其操作容易、绘图美观。可与 PPT、Word 相连接。

7．Excel

几乎所有人都知道这款软件。Microsoft Excel 是微软公司的办公软件 Microsoft Office 的组件之一，是由 Microsoft 为 Windows 和 Apple Macintosh 操作系统的电脑而编写和运行的一款电子表格软件。Excel 是微软办公套装软件的一个重要组成部分，它可以进行各种数据的处理、统计分析和辅助决策操作，广泛地应用于管理、统计财经、金融等众多领域。Excel 能实现大部分二维图表的绘制

1 MATLAB 软件数据可视化库：https://cn.mathworks.com/products/matlab/plot-gallery.html

2 PlotPub 包的官网：https://github.com/masumhabib/PlotPub

3 Gramm 包的官网：https://github.com/piermorel/gramm

与基础的数据处理与分析，具体可以参考学习《Excel 数据之美：科学图表与商业图表的绘制》。

实例分析 为更好地了解这 7 款绘图软件的风格，现采用相同的数据集合，分别绘制了散点图、曲线图、（堆积）柱形图和箱形图 4 种图表类型，如图 1-2-2 到图 1-2-8 所示。

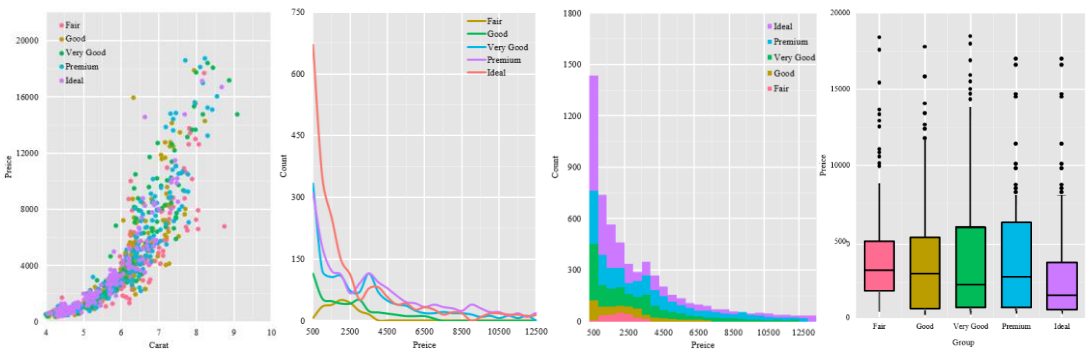
（1）图 1-2-2 由 R ggplot2 绘制，其图表风格最为独特与美观，这种图表在部分论文中也是有直接使用的。使用 R ggplot2 Set3 的颜色主题，绘图区背景填充颜色为 RGB（229, 229, 229）的灰色，以及白色的网格线[主要网格线的颜色为 RGB(255, 255, 255),次要网格线的颜色为 RGB(242, 242, 242)]。

（2）图 1-2-3 由 Python Seaborn 绘制，其图表风格也很有特色，使用 Seaborn 包的颜色主题方案，绘图区背景填充颜色为 RGB（234,234,242）的淡蓝色，以及 RGB（255,255,255）的白色的主要网格线（无次要网格线）。

（3）图 1-2-4 是使用 MATLAB 2014b 通过编程绘制的图表，使用 MATLAB 默认的颜色主题方案 Parula，网格线设定为“无”。MATLAB 通过函数（function）直接绘制的图表，可以通过图表编辑器对图表进行优化，但是也并不能实现箱形图颜色的填充。如果 MATLAB 使用 Gramm 包，则可以绘制更加美观的图表。

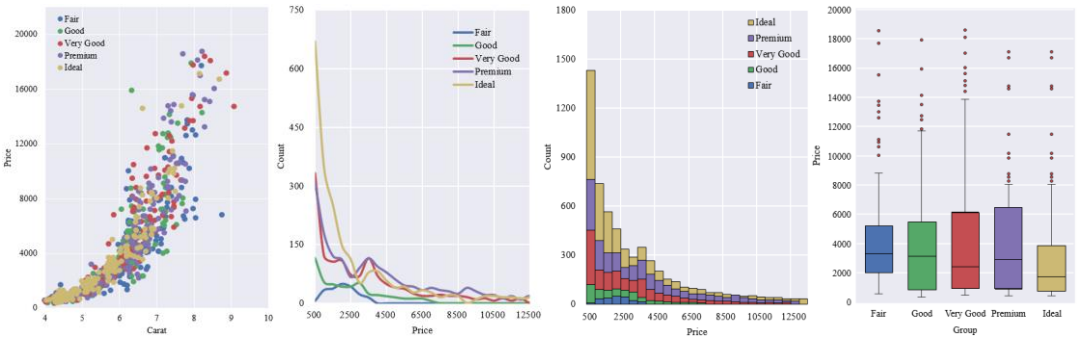
（4）图 1-2-5 到图 1-2-7 分别对应 SigmaPlot、Origin 和 GraphPad Prism 绘制的图表，这是最为常见的学术图表。它们的图表风格基本相同：绘图区背景填充颜色为 RGB（255,255,255）的白色，这样可以使背景不太复杂，尤其适应于图表尺寸较小时，可以保证数据的清晰展示；这些图表使用绘图软件的默认颜色主题，由于不同软件的颜色主题不同，即便是相同的图表样式，也会导致图表的美观存在较大的审美差异。

（5）图 1-2-8 是使用 Excel 绘制的图表，使用 Excel 默认颜色主题方案“Office 2007-2010”。Excel 2016 添加了几种新型图表类型，包括矩形树状图、箱形图等；Excel 2013 及以前版本只能通过堆积柱形图间接地实现箱形图。



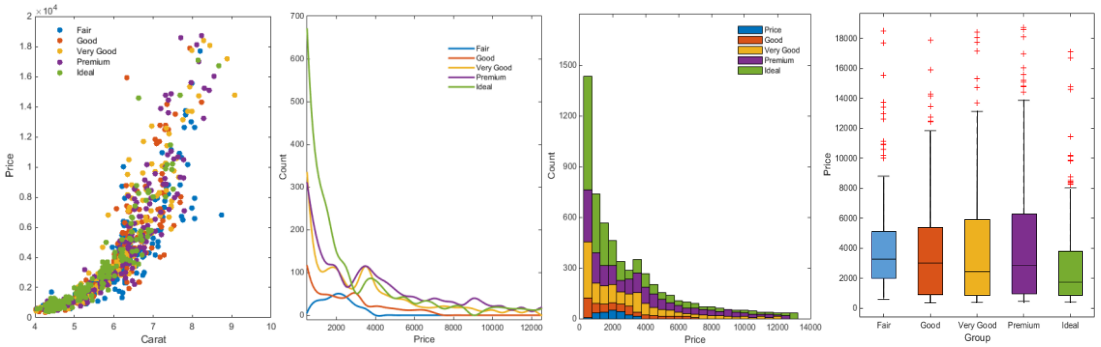
(a)　(b)　(c)　(d)

图 1-2-2　R ggplot2 图表实例



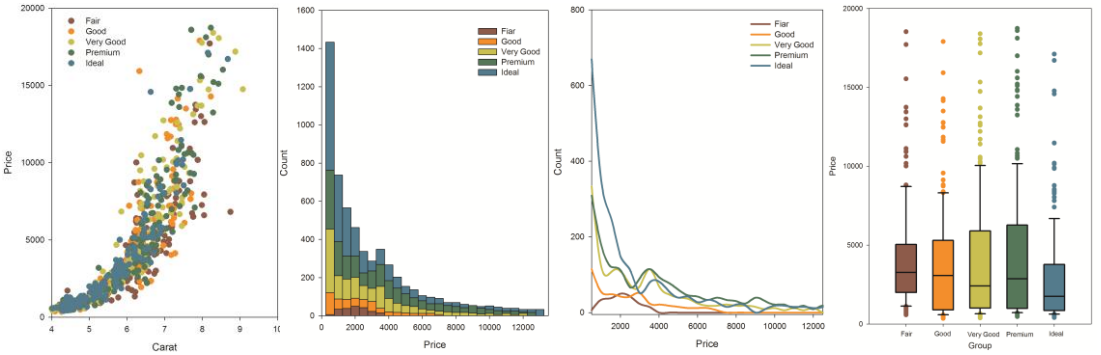
(a)　(b)　(c)　(d)

图 1-2-3　Python Seaborn 图表实例



(a)　(b)　(c)　(d)

图 1-2-4　MATLAB 图表实例



(a)　(b)　(c)　(d)

图 1-2-5　SigmaPlot 图表实例

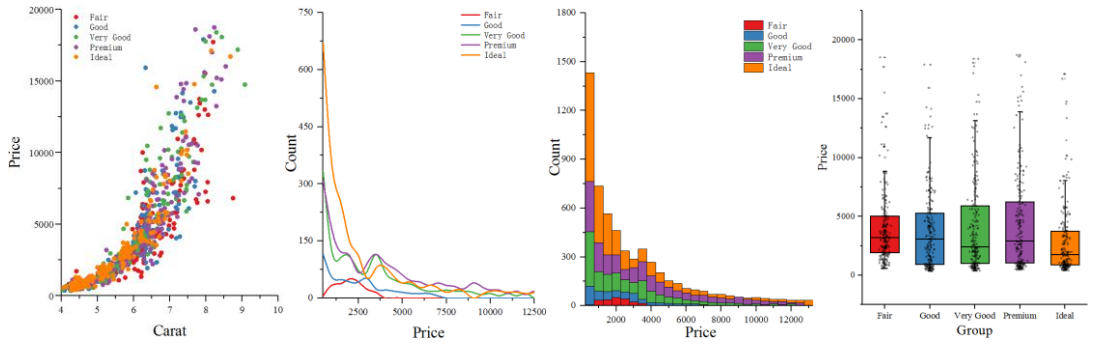


图 1-2-6　Origin 图表实例

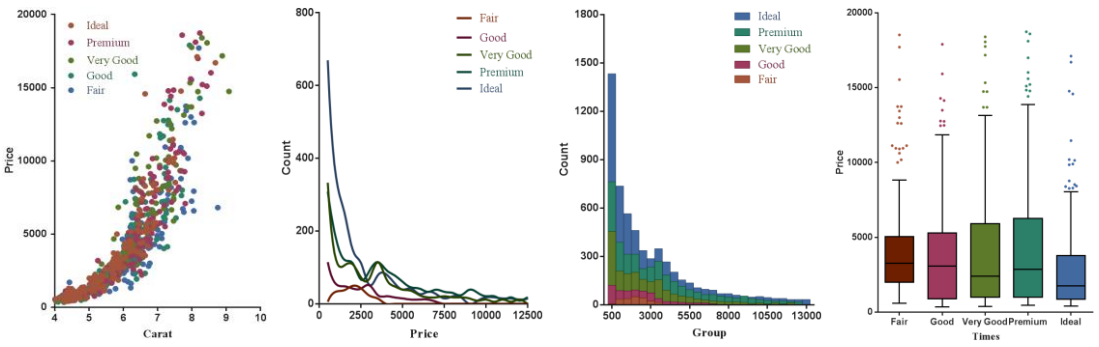


图 1-2-7　GraphPad Prism 图表实例

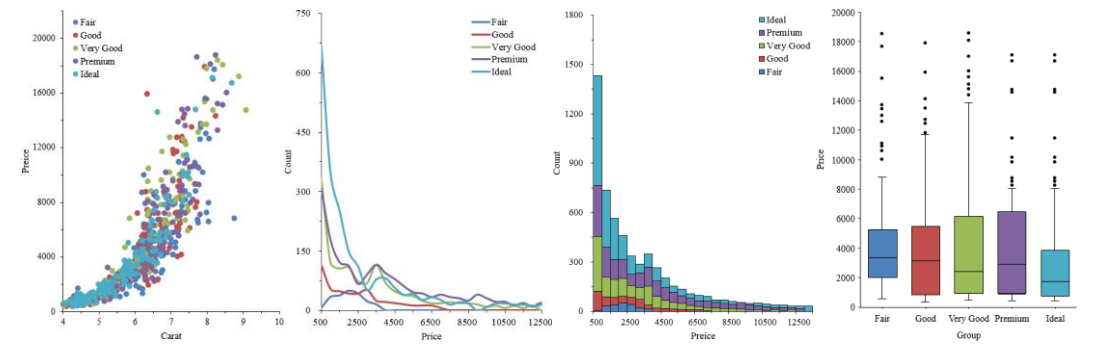


图 1-2-8　Excel 图表实例

在这么多种绘图软件中，我们为什么要选择 R 呢？首先因为 R 是开源的，可以免费使用。其次，R 是一套完整的数据处理、计算和制图软件系统。其功能包括：数据存储和处理系统、数组运算工具、完整连贯的统计分析工具、优秀的统计制图功能。尤其是 R 的 ggplot2 包及其拓展包以人性化的图形语法，可以快速帮助用户展示数据，并可以实现个性化的图表。另外，R 还有很多其他绘图包，比如可以绘制三维图表的 plot3D 包等，几乎可以帮助用户绘制所有常见的图表类型。

但是，绘图软件只是使用的一个工具而已[1]。归根结底，对数据的分析和图表的设计取决于你自己。如果你打算深入研究数据，而且日后可能（或者希望日后）还会接触大量与数据相关的项目，那么现在花些时间学习编程最终会节省其他项目的时间，并且作品也会给人留下更加深刻的印象。你的编程技巧会在每一次项目中获得提高，你会发现编程越来越容易。

心中有剑，落叶飞花，皆是兵器！

1.3　R 软件的安装与使用

1.3.1　R 与 RStudio 的安装

1．R 的获取和安装

R 可以在 CRAN（Comprehensive R Archive Network）[2]上免费下载。Linux、mac OS x 和 Windows 都有相应编译好的二进制版本，根据你所选择平台的安装说明进行安装即可。本书所用 R3.3.3 版本。有时候有些代码运行可能会由于 R 或者 R 包的版本，出现函数弃用（deprecated）的情况。此时，需要自己更新代码，使用新的函数替代原有的函数等。

2．RStudio 的获取与安装

虽然现在有很多可用的 IDE（集成开发环境），但是在这里推荐使用 JJ Allaire 小组设计的 RStudio。我们可以去 RStudio 官网下载免费版本 RStudio Desktop[3]。RStudio 的通用界面如图 1-3-1 所示。另外，在 RStudio 中可以使用 installr 包的 updateR()更新 R 版本：installr::updateR()。

1　更多绘图工具可参考：https://keshif.me/demo/VisTools

2　http://cran.r-project.org

3　https://www.rstudio.com/products/rstudio/download

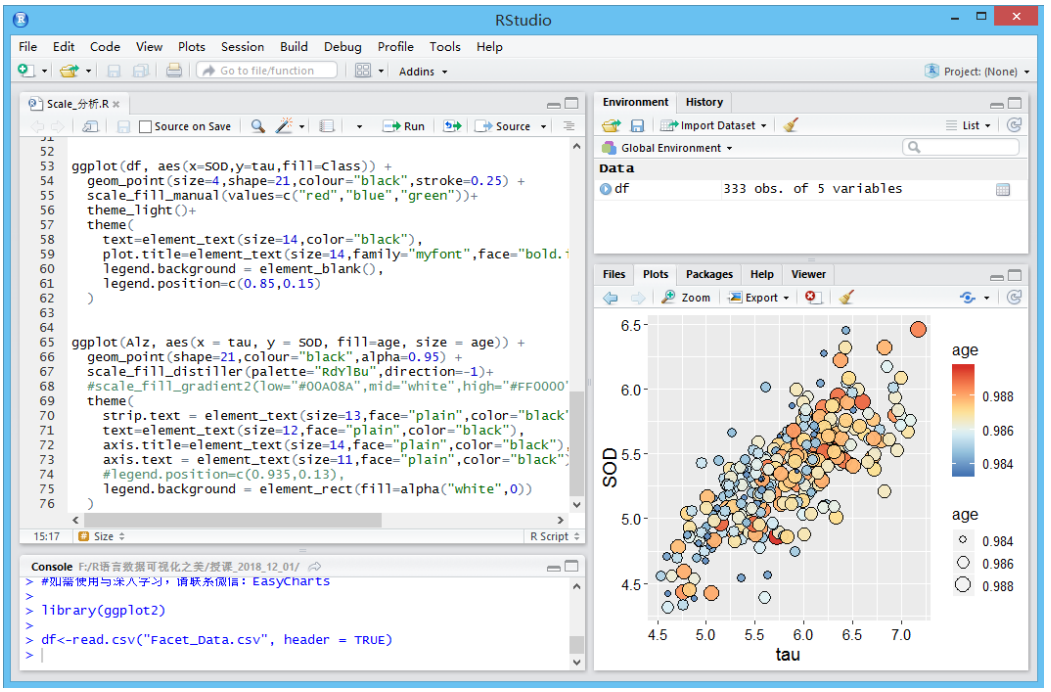


图 1-3-1　RStudio 通用界面

1.3.2　包的安装与加载

1. 包的安装

如果拥有 RStudio，那么最简单的方法是单击右下角写有“Packages”的选项卡，然后在弹出对话框中输入包的名称就可以。或者直接在左下角的“Console”控制台输入安装命令：install.packages("ggplot2")

有时候需要直接从 Github 或 BitBucket 上下载安装包，这种方法可以得到包的开发版本，但是需要使用 devtools 包来完成: devtools::install_github("tidyverse/ggplot2")。

2. 包的加载

既然包已经安装好，首先它需要被加载才能使用。现在主要有两种函数可供选择：library()或者 require()，比如：library(ggplot2)。

有时候已经加载的包可能需要被卸载。这个可以简单地通过在 RStudio 的“Packages”界面消除复选框中的选项，或使用 detach()函数：detach("package: ggplot2")

1.4　R 语言编程基础

R 是一种区分字母大小写的解释性语言，R 语句的分隔符是分号（;）或换行符。当语句结束时，可以不使用分号，R 语言会自动识别语句结束的位置。R 语言只支持单行注释，注释由#开头，当前行出现在#之后的任何文本都会被 R 解释器忽略。R 语句由函数和赋值构成。R 使用 <-，而不是传统的 = 作为赋值符号。R 语言的数学运算跟我们平时的数学运算（加+，减-，乘*，除/）基本一致。在这里，我们会重点讲解与 R 语言数据可视化相关的编程基础内容。

1.4.1　数据类型

R 语言有很多不同的数据类型，用于储存不同的数据。我们最常用到的 4 种数据类型为数值型（numeric）、字符型（character）、日期型（date）和逻辑型（logical）。变量中储存的数据类型都可以使用 class()函数查看。

①数值型（numeric）：

```
a<-1, is.numeric(a)   #输出判定 a 是否为数值型：TRUE
```

②字符型（character）：

```
b<- "peter"; nchar(b) #输出字符串的长度为：4
```

③日期型（date）：最常用的日期型数据类型是 Date（仅储存日期）和 POSIXct（同时储存日期与时间）

```
c<-as.Date（"2012-06-12"）; class(c)   #输出 c 的数据类型为："Date"
d<-as.POSIXct（"2012-06-12 17:32"）; class(d)   #输出 d 的数据类型为："POSIXct" "POSIXt"
```

④逻辑型（logical）：

```
e<-TRUE, f<-FALSE
```

其中，在处理时序数据时，我们需要处理日期型数据，往往需要使用 as.Date()函数将读入的数据从数值型转换成日期型，有时候还需要进一步提取日期型数据的年、月、周等数据信息。此时我们需要使用 as.numeric()函数或者 as.integer()函数将日期型数据转换成数值型。其中，strftime(x, format = "")函数可以定义日期型数据的格式，比如 strftime(c, '%Y')表示只显示年份。

```
c_Year<- as.integer(strftime(c, '%Y'))    #输出年份：2012
c_month <- as.integer(strftime(c, '%m')) #输出月份：6
c_week<- as.integer(strftime(c, '%W'))   #输出周数：24
```

1.4.2 数据结构

常见的数据结构包括：向量（vector）、数据框（data.frame）、矩阵（matrix）、列表（list）和数组（array）。其中，矩阵是将数据用行和列排列的长方形表格，它是二维的数组，其单元必须是相同的数据类型，通常用列来表示不同的变量，用行表示各个对象；数组可以看作是带有多个下标的类型相同的元素的集合；列表是一个对象的有序集合构成的对象，列表中包含的对象又称为它的分量（component），分量可以是不同的模式或（和）类型。我们在本书的数据可视化中，比较常用的是向量（因子属于特殊的向量）和数据框，所以我们重点介绍这两种类型的数据结构，还将介绍与数据可视化密切相关的函数。

1. 向量

向量是用于存储数值型、字符型或逻辑型数据的一维数组。执行组合功能的函数 c()可用来创建向量（c 代表合并：combine）。值得注意的是，单个向量中的数据类型是固定的，比如数值型向量中的元素就必须全为向量。量是 R 语言中最基本的数据结构，其他类型的数据结构都可以由向量构成。最常见的向量有三种类型：数值型、字符型、逻辑型。

（1）向量的创建

向量的创建有多种方法，我们既可以手动输入，使用函数 c()创建向量；也可以使用现有的函数创建向量，比如 seq()、rep()等函数，具体如表 1-4-1 所示。

表 1-4-1 向量的创建

输入	输出	描述
c(2,4,6)	2 4 6	将元素连接成向量
2:6	2 3 4 5 6	等差整数数列
seq(2, 3, by=0.5)	2.0 2.5 3.0	步长为 0.5 的等差数列
rep(1:2, times=3)	1 2 1 2 1 2	将一个向量重复 3 次
rep(1:2, each=3)	1 1 1 2 2 2	将一个向量中的每个元素重复 3 次
rnorm(3, mean = 0, sd = 3)	-2.09 -3.52 -4.25	均值为 0、标准差为 3 的正态分布
runif(3, min = 0, max = 1)	0.63 0.05 0.61	最大值为 1、最小值为 0 的均匀分布
sample(c("A","B","C"), 4, replace=TRUE)	"A" "A" "A" "B"	从一个向量中随机抽取

（2）向量的处理

- 向量的排序。向量的排序和数据框的排序有时候对数据的展示尤为重要，很多时候我们需要对数据先进行降序处理，再展示数据。sort()函数可以实现对向量的排序处理，

index.return=TRUE，表示返回排序的索引；decreasing = TRUE，表示降序处理。如下输出的结果 order 包括两部分：$x 为[5 4 3 2 1]，$ix 为[4 2 3 5 1]。

```
Vec<-c(1,4,3,5,2)
order<-sort(Vec, index.return=TRUE,decreasing = TRUE)
```

- 向量的唯一值。unique()函数主要是返回一个删除了把重复元素或行的向量、数据框或数组。在需要对数据框根据某列进行分组运算时，需要使用该函数先获取类别总数。

```
Vec<-c("peter","jack","peter","jack","eelin")
Uni<-unique(Vec)  #输出："peter", "jack", "eelin"
```

- 连续向量的离散化。在做数据挖掘模型时，我们有时会需要把连续型变量转换为离散型变量，这种转换的过程就是数据离散化，分箱就是离散化常用的一种方法。数据离散化最简单的方法就是使用 cut()函数自定义离散区间，从而对数据进行离散处理。

```
Num_Vector<- c(10, 5, 4, 7, 6, 1, 4, 8, 8, 5)
Cut_Vector<-cut(Num_Vector,breaks=c(0,3,6,9,11), labels=c("0~3", "3~6", "6~9", ">9"), right = TRUE)
# 输出结果为因子向量：>9, 3~6, 3~6, 6~9, 3~6, 0~3, 3~6, 6~9, 6~9, 3~6；其水平 Levels 为: 0~3, 3~6, 6~9, >9
```

（3）向量的索引

向量是多个元素的集合，当我们只需要指定或者说提取该向量中的某个元素时，就可以使用向量的索引（indexing）。向量元素有三种基本类型的向量索引：整数型，索引的是元素位置；字符型，索引的是名称属性；逻辑型，索引的是相同长度的逻辑向量对应的逻辑值为真的元素。

```
x<-c(1,4,3,5,2)
```

- 整数型索引，选择某个或多个元素：x[2]; x[-2]; x[2:4]; x[c(1,4)]
- 逻辑型索引，逻辑运算选择元素：x[x>2]; x[x==1]; x[x<=5]

2. 因子

因子（factor）是 R 语言中许多强大运算的基础，包括许多针对表格数据的运算，可分为类别型变量和有序型变量。因子可以看成是包含了额外信息的向量，这额外的信息就是不同的类别，称之为水平（level）。因子在 R 中非常重要，因为它决定了数据的分析方式，以及如何进行视觉呈现。

（1）因子的创建

一个因子不仅包括分类变量本身，还包括变量不同的可能水平（即使它们在数据中不出现）。因子函数 factor()用下面的选项创建一个因子。对于字符型向量，因子的水平默认依字母顺序创建:

```
(Fair,Good, Ideal, Premium, Very Good)
Cut<-c("Fair","Good","Very Good","Premium","Ideal")
```

```
Cut_Facor1<-as.factor(Cut)
```

（2）水平的更改

很多时候，按默认的字母顺序排序的因子很少能够让人满意。因此，可以指定 levels 选项来覆盖默认排序。更改因子向量的 levels 为("Good","Fair","Very Good","Ideal","Premium")，就需要使用 factor()函数更改 levels。

```
Cut_Facor2<-factor(x=c("Fair","Good","Very Good","Premium","Ideal"),
                   levels= c("Good","Fair","Very Good","Ideal","Premium"),
                   ordered=TRUE)
```

（3）类型的转换

数值型因子向量的类型变换。有时候我们需要将数值型的因子向量重新转换成数值型向量，这时，我们需要使用 as.numeric(as.character())组合函数，而不能直接使用 as.numeric()函数。其中 as.character()函数表示将向量变成字符型，as.numeric()函数表示将向量变成数值型。

```
Num _Facor<-factor(x=c(1,3,5,2), levels= c(5,3,2,1), ordered=TRUE)
Num_Vector1<-as.numeric(as.character(Num_Facor))   # 输出：1 3 5 2
Num_Vector2<-as.numeric(Num_Facor)                 # 输出：4 2 1 3
```

3. 数据框

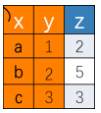
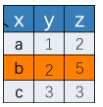
数据框是 R 语言中的一个种表格结构，对应于数据库中的表，类似 Excel 中的数据表。数据框是由多个向量构成的，每个向量的长度相同。数据框类似矩阵，也是一个二维表结构。在统计学术语中，用行来表示观测（observation），用列来表示变量（variable）。

（1）数据框的创建与查看

创建数据框，最简单的方法就是，用同名的定义函数 data.frame()，输入每个变量的名称及对应的向量，每个向量的长度相同。一个数据框可能包含多个变量（向量），有时需要单独提取某个变量，使用特殊的$符号来访问，由“数据框$变量名”构成。数据框数据的选取如表 1-4-2 所示。

表 1-4-2 数据框数据的选取

语句	示例	语句	示例
数据框的构建： df<-data.frame(x= c("a","b","c"), y=1:3,z=c(2,5,3))	x y z a 1 2 b 2 5 c 3 3	选取某一列： df[,2]，df$y，df[[2]]	x y z a 1 2 b 2 5 c 3 3

<table>
<tr><td>选取多列：
df[c("x","y")]，df[,1:2]</td><td>  </td><td>选取某一行：
df[2,]</td><td>  </td></tr>
<tr><td>选取多行：
df[1:2,]</td><td>  </td><td>选取某个元素：
df[2,2]</td><td>  </td></tr>
</table>

- 获取数据框的行数、列数和维数：nrow()、ncol()、dim()。
- 获取数据框的列名或行名 :names()、rownames()、colnames() ;重新定义列名 :names(df)<-c("X", "Y", "Z")。
- 观察数据框的内容：view(df)、head(df, n=3)、tail(df)。

（2）空数据框的创建

创建空数据框，在需要自己构造绘图的数据框数据信息时尤为重要。有时候，绘制复杂的数据图表的过程中，我们需要对现有的数据进行插值、拟合等处理时，需要使用空的数据框储存新的数据，最后使用新的数据框绘制图表。创建空的数据框主要有如下两种方法。

- 创建一个名为 Df_Empty，包括两个变量（var_a 为 numeric 类型；var_b 为 character 类型）的 data.frame。但是注意：要加上 stringsAsFactors=FALSE，否则在后面逐行输入数据时，会因为 var_b 的取值未经定义的 factor level 而报错。

```
Df_Empty1<- data.frame(var_a = numeric(),var_b = character(),stringsAsFactors=FALSE)
```

- 先使用矩阵创建空的数据框，同时通过 dimnames 设定数据框的列名。这个相比于前一种方法可以更快速地创建多列空数据框：

```
Df_Empty2 <- data.frame(matrix(ncol=2, nrow=0,dimnames=list(c(),c("var_a","var_b"))))
```

1.4.3　数据属性

数据框作为 R 语言数据分析与可视化很常用的数据结构，常由多列不同数据属性的变量组成。在我们实现数据可视化时，很有必要先了解这些变量的属性。我们平时记录的实验数据所用的表（table）就是由一系列不同属性的变量组成的。 Jiawei Han 等人的 *Data mining: concepts and techniques*[13]根据数据属性取值的集合类型，对数据属性进行了分成三类：类别型、序数型和数值型，如图 1-4-1 所示。Pang-Ning Ta 等人的 *Introduction to Data Mining*[14]，将序数型和类别型数据统称为类别型（categorical）或者定性型（qualitative），将数值型（numeric）也称为定量型（quantitative）。

类别型	序数型	数值型

图 1-4-1　不同数据类型

1. 类别型

类别型属性（categorical attribute）是用于区分不同数据对象的符号或名称，而它们是没有顺序关系的，又包含多元类别和二元类别两种类型。对于多元类别，可以理解为购买服装时的不同服装名称，如衬衫、毛衣、T 恤、夹克等；对于二元类别，可以理解为购买服装时的不同性别，只有男士和女士两种性别分类。类别型数据的可视化一般使用标尺类中的分类尺度。

2. 序数型

序数型属性（ordinal attribute）的属性值是具有顺序关系，或者存在衡量属性值顺序关系的规则。比如常见的时序数据，就一般是按时间先后排序的；还有就是平时调查问卷中经常使用的 5 个喜欢程度：非常喜欢、比较喜欢、无所谓、不太喜欢、非常不喜欢。序数型数据的可视化一般使用标尺类中的顺序尺度和时间尺度两种类型。

序数型数据的排列方向有三种，分别是单向型（sequential），有公共零点的双向型（diverging），以及环状周期型（cyclic），如图 1-4-2 所示。

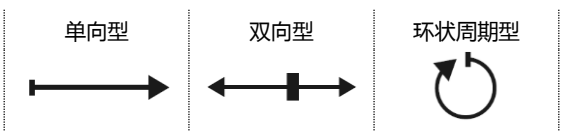


图 1-4-2　不同数据结构的序数型

3. 数值型

数值型属性（numeric attribute）使用定量方法表达属性值，如整数或者实数，包括区间型数值属性（interval-scaled attribute）和比值型数值属性（ratio-scaled attribute），如表 1-4-3 所示。区间型与比值型数值最大的区别就是有无基准点，通常为零点（internal zero-point）。

比值型数值属性的数据一般拥有基准点，比如开氏温标（K）以绝对零度（0K=–273.15°C）为其零点，以及平时通常使用的数量、重量、高度和速度等。

而区间型数值属性的数据的起始值一般是在整个实数区间上取值，可进行差异运算，但不能进行比值运算。比如摄氏温标（°C）与华氏温标（°F）下的温度、日历中的年份、经度（longitude）与纬度（latitude），它们都没有真正的零点。在日历中，0 年并不对应时间的开始，但 0°C 并不代表没有温度。所以可以说 10°C 比 5°C 温度高（差异运算），但是不能说 10°C 是 5°C 的 2 倍（比值运算）。

表 1-4-3　包含不同数据属性的变量组合表 [13]

对象标识	test-1（类别型）	test-2（序数型）	test-3（区间型数值型）	test-4（比值型数值型）
1	产品-A	非常喜欢	2002（年）	445 K
2	产品-B	比较喜欢	2003（年）	22 K
3	产品-C	无所谓	2005（年）	164 K
4	产品-A	不太喜欢	2007（年）	123 K

我们也可以用值的个数区分数据类型，可以分为离散型和连续型 [14]。离散型属性具有有限个值或者无限个值，这样的属性可以是分类的，也可以是数值型的。其中二元属性（binary attribute）是离散型属性的一种特殊情况，并只接受两个值，比如 True/False（真/假）、Yes/No（是/否）、Male/Female（男/女），以及 0/1。通常二元属性使用布尔变量表示，或者只取 0 和 1 两个值的整数变量表示。连续型属性是取实数值的属性，通常使用浮点数变量表示。理论上讲，基于数据集合类型划分的数据类型（类别型、序数型和数值型）可以与基于属性值个数的任意类型（离散型和连续型）组合，从而不同的数据可能有不同的数据属性组合。

1.4.4　数据的导入导出

1. 数据文件的导入与导出

我们常用外部保存的数据文件来绘制图表。此时，就需要借助可以导入数据的函数导入不同格式的数据，包括 CSV、TXT，以及 Excel、SQL、HTML 等数据文件。有时候，我们也需要将处理好的数据从 R 语言中导出保存。其中，我们在数据可视化中使用最多的就是前 3 种格式的数据文件。

（1）CSV 格式数据的导入与导出

使用 read.csv()函数，可以导入 CSV 格式的数据，并存储为数据框形式。需要注意是：当 stringsAsFactors=TRUE 时，R 会自动将读入的字符型变量转换成因子，但是这样很容易导致数据只按默认字母顺序展示。在导入大批量数据时，为了提高性能，尽可能分两步走：

① 显式指定“stringsAsFactors = FALSE”；

② 依次将所需要的数据列（向量）转换为因子。

```
mydata<-read.csv("Data.csv",sep=",",na.strings="NA",stringsAsFactors=FALSE)
```

使用 write.csv()函数，可以将 data.frame 的数据存储为 CSV 文件

```
write.csv(mydata,file = "File.csv")
```

CSV 文件主要有以下 3 个特点。

① 文件结构简单，基本上和 TXT 文本的差别不大；

② 可以和 Excel 进行转换，这是一个很大的优点，很容易进行查看模式转换，但是其文件的存储大小比 Excel 小。

③ 由于其简单的存储方式，一方面可以减少存储信息的容量，这样有利于网络传输以及客户端的再处理；另一方面，由于是一堆没有任何说明的数据，其具备基本的安全性。所以相比 TXT 和 Excel 数据文件，我们更加推荐使用 CSV 格式的数据文件进行导入与导出操作。

（2）TXT 格式数据的导入与导出

使用 read.table()函数不仅可以导入 CSV 格式的文件数据，还可以导入 TXT 格式的文件数据，并存储为数据框数据。

```
mydata<-read.table("Data.txt",header = TRUE)
```

使用 write.table()函数可以将 data.frame 的数据存储为 CSV 文件：

```
write.table(mydata, file = "File.txt")
```

（3）Excel 格式数据的导入与导出

使用 xlsx 包[1]的 read.xlsx()函数和 read.xlsx2()函数可以导入 XLSX 格式的数据文件。但是更推荐使用 CSV 格式导入数据文件。

```
mydata<- read.xlsx("Data.xlsx", sheetIndex=1)
```

也可以使用 write.xlsx()函将数据文件导出为 XLSX 格式：

```
write.xlsx(mydata, "Data.xlsx", sheetName="Sheet Name")
```

需要注意的是：使用 R ggplot2 绘图时，通常使用一维数据列表的数据框。但是如果有时候导入的数据表格是二维数据列表 那么我们需要使用 reshape2 包的 melt()函数或者 tidyr 包的 gather()函数，可以将二维数据列表的数据框转换成一维数据列表。

一维数据列表和二维数据列表的区别 一维数据列表就是由字段和记录组成的表格。一般来说字段在首行，下面每一行是一条记录。一维数据列表通常可以作为数据分析的数据源，每一行代表完整的一条数据记录，所以可以很方便地进行数据的录入、更新、查询、匹配等，如图 1-4-3 所示。 二维数据列表就是行和列都有字段，它们相交的位置是数值的表格。	

1 更多关于 xlsx 包的使用可参考：https://github.com/colearendt/xlsx

这类表格一般是由分类汇总得来的，既有分类，又有汇总，所以是通过一维数据列表加工处理过的，通常用于呈现展示，如图 1-4-4 所示。

一维数据列表也常被称为流水线表格，它和二维数据列表做出的数据透视表最大的区别在于“行总计”。判断数据是一维数据列表还是二维数据列表的一个最简单的办法，就是看其列的内容：每一列是否是一个独立的参数。如果每一列都是独立的参数那就是一维数据列表，如果每一列都是同类参数那就是二维数据列表。

注意　为了后期更好地创建各种类型的数据透视表，建议用户在数据录入时，采用一维数据列表形式的进行数据录入，避免采用二维数据列表的形式对数据进行录入。

Name	Subject	Grade
Peter	English	99
Peter	Math	84
Peter	Chinese	95
Jack	English	83
Jack	Math	93
Jack	Chinese	92
Jon	English	82
Jon	Math	90
Jon	Chinese	84

图 1-4-3　一维数据列表

Name	English	Math	Chinese
Peter	99	84	95
Jack	83	93	92
Jon	82	90	84

图 1-4-4　二维数据列表

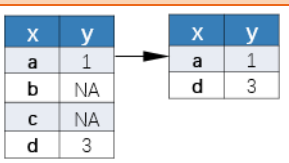
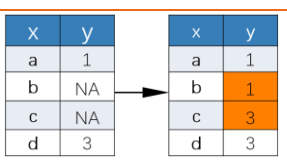
2. 缺失值的处理

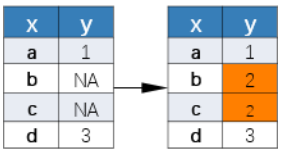
有时候，我们导入的数据存在缺失值。另外，在统计与计算中，缺失值也起着至关重要的作用。R 语言中主要有两种类型的缺失数据：NA 和 NULL。

（1）NA

在 R 中，使用 NA 代替缺失数据作为向量中的另外一种元素出现。我们可以使用 is.na()函数来检查向量或数据框中的每个元素是否缺失数据。我们先构造一个含有缺失数据的数据框，然后讲解使用 tidyr 包实现常用的缺失数据的处理方法，如表 1-4-4 所示。

表 1-4-4　缺失值的处理

ID	代码	示意
1	直接删除带 NA 的行： tidyr::drop_na(df,y)	
2	使用最邻近的元素填充 NA： tidyr::fill(df,y)	

3	使用指定的数值替代 NA: tidyr::fill(df,list(y=2))	

（2）NULL

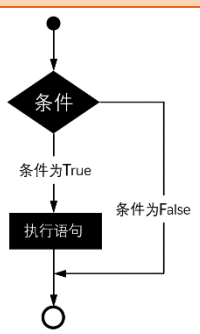
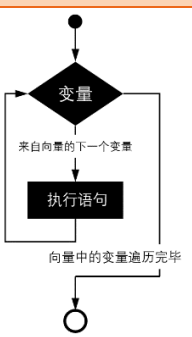
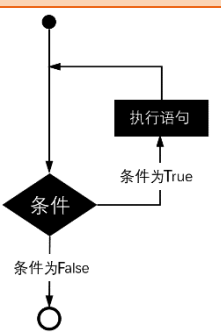
NULL 就是没有任何东西，表示数据的空白，而并非数据的缺失，也不能成为向量或者数据框的一部分。在函数中，参数有可能是 NULL，返回的结果也可能是 NULL。我们可以使用 is.null()函数判定变量是否为 NULL。

1.4.5 控制语句与函数编写

我们常用的控制语句包括 if…else、ifelse 条件语句，以及 for 和 while 循环语句。其中我们最常见的就是 if…else ,主要用于检查判定。其条件最基本的检查包括等于(=)、小于(<)、小于等于(<=)、大于 (>)、大于等于 (>=) 和不等于 (!=)。if…else 语句对数据的操作运算命令都需要放在{}里面。需要注意的是：else 跟在其左边的大括号“}”必须在同一行，否则程序无法识别，会导致代码运行错误。另外，R 语言还有一个 ifelse()语句，可以向量化 if 语句，从而加速代码的运行，如表 1-4-5 所示的 if…else 条件语句可以使用 ifelse()重写为：ifelse(i > 3, print('Yes'), print('No'))。该语句可以结合 transform()函数等对数据框的每个元素进行判别运算，从而生成新的列。

我们最常用的循环是 for 循环，for 循环的向量不一定是连续型的，也可以是其他类型的向量，如表 1-4-6 所示的 for 循环示例。其中 ,1:4 的输出起点为 1、终点为 4、步长为 1 的等差数列向量(1,2,3,4)，效果类似于 seq(1,4,1)。另外，while 循环虽然没有 for 循环用得普遍，但是更加易于操作。但对新手来说，容易由于设定的循环条件有误而导致循环不停迭代，从而陷入“死循环”。

表 1-4-5 控制语句

类别	if…else 条件语句	for 循环语句	while 循环语句
示意			

语法	if (条件){
	　执行语句
	}　else {
	　其他执行语句　}
		for (变量 in 向量){
		　执行语句
		}
			while (条件){
			　执行语句
			}

示例	i<-5
	if (i > 3){
	　print('Yes')
	} else {
	　print('No')}
		for (i in 1:4){
		　j <- i + 10
		　print(j)
		}
			i<-1
			while (i < 5){
			　print(i)
			　i <- i + 1
			}

输出	'Yes'	11,12,13,14	1,2,3,4

我们在实现数据可视化时，更多是使用现有包的函数，比如等差数列生成函数 seq()、向量排序函数 sort()、插值函数 spline()等，而很少需要自定义函数（表 1-4-6 为各种自定义函数的语法格式）。我们更加需要了解的是现有函数的输入参数与数据的结构、输出参数的数据内容等，比如 plot3D 包的 persp3D()函数和 lattice 包的 wireframe()函数都可以绘制相同的三维曲面图，但是 persp3D()函数要求输入的数据是向量与矩阵形式，而 wireframe()函数要求输入的数据是数据框。

表 1-4-6　自定义函数

ID	自定义函数的语法	示例

1	函数名<-function(参数){
	　执行语句
	　return(新数据)
	}
		square <- function(x){
		　squared <- x*x
		　return(squared)
		}
		print(square(2)) #输出结果为 4

2	函数名<-function(参数 1，参数 2){
	　执行语句
	　return(新数据)
	}
		square <- function(x,y){
		　squared <- x*y
		　return(squared)
		}
		print(square(2,3)) #输出结果为 6

3	函数名<-function(参数 1，参数 2){
	　执行语句
	　return(c(新数据 1，新数据 2))
	}
		square <- function(x,y){
		　squared1 <- x*x
		　squared2 <- y*y
		　return(c(squared1,squared2))
		}
		print(square(2,3)) #输出结果为 4,9

1.5 R 语言绘图基础

在 R 里，主要有两大底层图形系统，一是 base 图形系统，二是 grid 图形系统。lattice 包与 ggplot2 包正是基于 grid 图形系统构建的，它们都有自己独特的图形语法。

grid 图形系统可以很容易地控制图形基础单元，给予编程者创作图形极大的灵活性。grid 图形系统还可以产生可编辑的图形组件，这些图形组件可以被复用和重组，并能通过 grid.layout()等函数，把图形输出到指定的位置上。但是因为 grid 包中没有提供生成统计图形及完整绘图的函数，因此很少直接采用 grid 包来分析与展示数据。

lattice 包通过一维、二维或三维条件绘图，即所谓的栅栏（trellis）图来对多元变量关系进行直观展示。相比于 base()函数是直接在图形设备上绘图的，lattice()函数是返回 trellis 对象。在命令执行的时候，栅栏图会被自动打印，所以看起来就像是 lattice()函数直接完成了绘图[15]。更多关于 base、grid 和 lattice 的语法可以参考 Murrell 和 Paul 所撰写的书籍 *R graphics*[15]。

ggplot2 包则基于一种全面的图形语法，提供了一种全新的图形创建方式，这套图形语法把绘图过程归纳为数据（data）、转换（transformation）、度量（scale）、坐标系（coordinate）、元素（element）、指引（guide）、显示（display）等一系列独立的步骤，通过将这些步骤搭配组合，来实现个性化的统计绘图。于是，得益于该图形语法，Hadley Wickham 所开发的 ggplot2 包是如此人性化，不同于 R base 基础绘图和先前的 lattice 包那样参数繁多，而是摈弃了诸多烦琐细节，并以人性化的思维进行高质量作图。在 ggplot2 包中，加号（+）的引入是革命性的，这个神奇的符号完成了一系列图形语法叠加[16, 17]。更多 ggplot2 的使用与学习可以参考两本关于 ggplot2 的经典书籍：*ggplot2 Elegant Graphics for Data Analysis* [16]和 *R Graphics Cookbook*[17]。

R 语言基础安装中就包含 base、grid 和 lattice 三个包，无须另外下载。但是除了 base 包，其他包依旧需要使用 library()函数加载后，才能被使用。使用 base、lattice 和 ggplot2 包绘制的散点图、统计直方图和箱形图，如图 1-5-1、图 1-5-2 和图 1-5-3 所示。

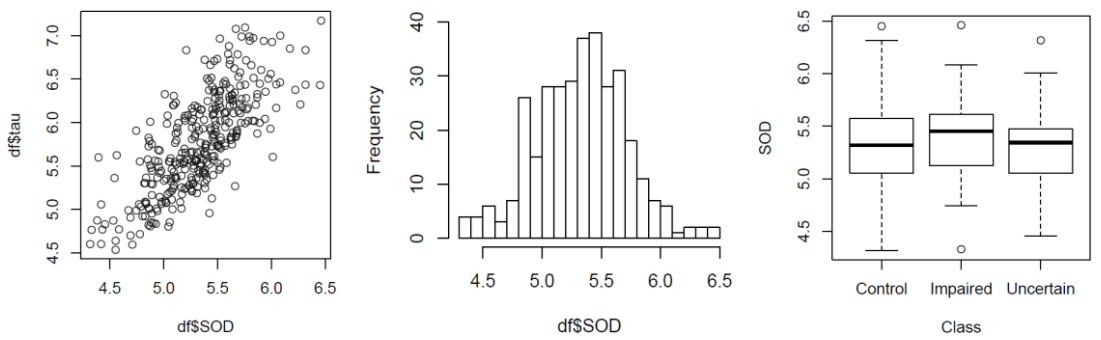


图 1-5-1　base 包绘制的图表示例

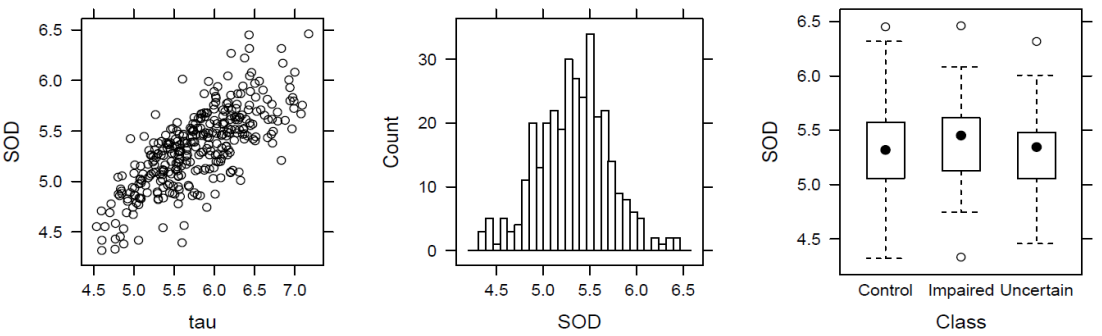


图 1-5-2　lattice 包绘制的图表示例

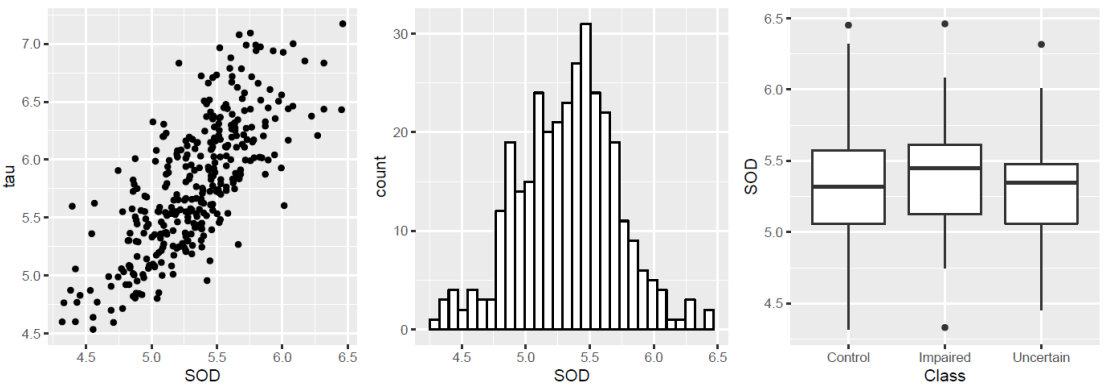


图 1-5-3　ggplot2 包绘制的图表示例

使用 base、lattice 和 ggplot2 包绘制的散点图、统计直方图和箱形图的具体代码如表 1-5-1 所示。df 是一个包含 SOD、tau 和 Class(Control、Impaired 和 Uncertain)三列的数据框。其中，base 和 lattice 语法最大的问题就是参数繁多、条理不清。而 ggplot2 语法相对来说很清晰，可以绘制很美观的个性

化图表。本书将会以图表类型为线索，详细地介绍常用图表的绘制方法，以 ggplo2 图形语法为主，但是有时候也会使用 base 和 lattice 等图形语法。

表 1-5-1　不同图形语法的代码示例

图形语法	散点图	统计直方图	箱形图
base	plot(dfSOD, dftau)	hist(df$SOD,breaks =30,ylim=c(0,40),main = "")	boxplot(SOD~Class,data=df,xlab="Class",ylab="SOD")
lattice	xyplot(SOD~tau,df,col="black")	histogram(~SOD,df,type="count",nint=30,col="white")	bwplot(SOD~Class,df,xlab="Class", par.settings = canonical.theme(color = FALSE))
ggplot2	ggplot(df, aes(x=SOD,y=tau)) + geom_point()	ggplot(df, aes(SOD)) + geom_histogram(bins=30,colour="black",fill="white")	ggplot(df, aes(x=Class,y=SOD)) + geom_boxplot()

1.6　ggplot2 图形语法

ggplot2 是一个功能强大且灵活的 R 包，由 Hadley Wickham 编写，它可以生成优雅而实用的图形。ggplot2 中的 gg 表示图形语法（grammar of graphic），这是一个通过使用“语法”来绘图的图形概念。ggplot2 主张模块间的协调与分工，整个 ggplot2 的语法框架如图 1-6-1 所示，主要包括数据绘图部分与美化细节部分。R ggplot2 图形语法的主要特点如下所示。

（1）采用图层的设计方式，有利于结构化思维实现数据可视化。有明确的起始（ggplot()开始）与终止，图层之间的叠加是靠“+”实现的，越往后，其图层越在上方。通常一条 geom_×××()函数或 stat_×××()函数可以绘制一个图层。

（2）将表征数据和图形细节分开，能快速将图形表现出来，使创造性的绘图更加容易实现。而且通过 stat_×××()函数将常见的统计变换融入绘图中。

（3）图形美观，扩展包（extension package）丰富，有专门调整颜色（color）、字体（font）和主题（theme）等辅助包。可以帮助用户快读定制个性化的图表。

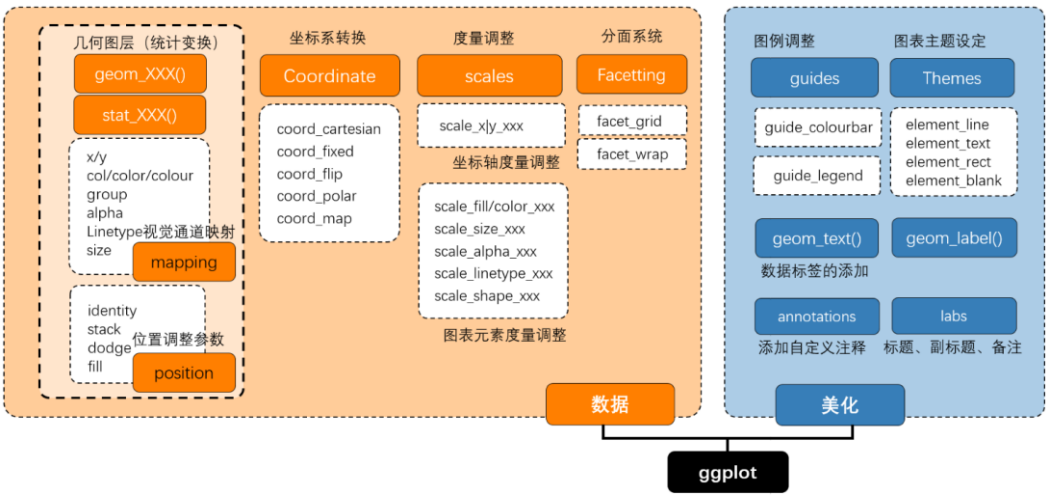


图 1-6-1　ggplot2 语法框架

ggplot2 的绘图基本语法结构如图 1-6-2 所示。其中所需的图表输入信息如下所示。

（1）ggplot()：底层绘图函数。DATA 为数据集，主要是数据框（data.frame）格式的数据集；MAPPINGS 变量的视觉通道映射，用来表示变量 x 和 y，还可以用来控制颜色（color）、大小（size）或形状（shape）等视觉通道；STAT 表示统计变换，与 stat_×××()相对应，默认为"identity"（无数据变换）；POSITION 表示绘图数据系列的位置调整，默认为"identity"（无位置调整），关于 POSITION 的具体内容可见第 3 章 3.1 节。

（2）geom_×××() | stat_×××()：几何图层或统计变换，比如常见的 geom_point()（散点图）、geom_bar()（柱形图）、geom_histogram()（统计直方图）、geom_ boxplot()（箱形图）、geom_line()（折线图）等。我们通常使用 geom_×××()函数就可以绘制大部分图表，有时候通过设定 stat 参数可以先实现统计变换。

可选的图表输入信息包括如下 5 个部分，主要是实现图表的美化与变换等。

（1）scale_×××()：度量调整，调整具体的度量，包括颜色（color）、大小（size）或形状（shape）等，跟 MAPPINGS 的映射变量相对应；

（2）coord_×××()：坐标变换，默认笛卡儿坐标系，还包括极坐标系、地理空间坐标系等；

（3）facet_×××()：分面系统，将某个变量进行分面变换，包括按行、列和网格等形式分面绘图，这部分内容具体可见第 9 章 9.4 节。

（4）guides()：图例调整，主要包括连续型和离散型两种类型的图例。

（5）theme()：主题设定，主要用于调整图表的细节，包括图表背景颜色、网格线的间隔与颜色等。

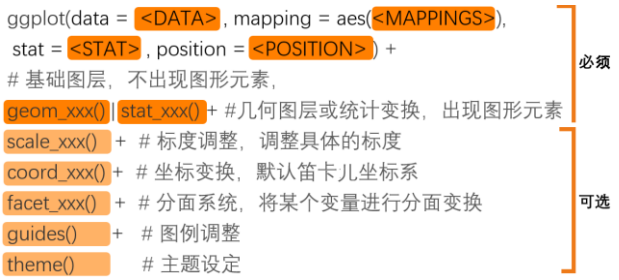


图 1-6-2　ggplot2 绘图的基本语法结构

Broadview®
www.broadview.com.cn
博文视点·IT出版旗舰品牌
技术凝聚实力·专业创新出版

"Beautiful Visualization with R" provides a comprehensive introduction to the many types of visualisation that you can create with R. As well as foundational visualisations like bar charts, scatterplots, and histogram, this book will teach you about more complex tools like 3d plots, ternary plots, polar coordinate plots and more! Read this book to learn how to create visually compelling plots that help you understand your data.

Hadley Wickham / RStudio首席科学家，莱斯大学统计系助理教授，ggplot2等软件包开发者

一图胜千言。用图表表达数据是人机和人际沟通的重要方式。然而，数据的复杂性、任务的多样性、用户的主观性，导致精确贴合用户需求的可视化工具难求。R语言是一个通用的数据分析工具。基于R语言的图表快速生成，是提升可视化的平民化和普适化的重要途经。感谢张杰，做了这样一个关键的尝试，相信会极大地促进数据科学的发展。

陈为 / 浙江大学教授，浙江大学计算机学院副院长，博士生导师

在大数据时代，数据可视化对数据分析的最终表达尤为重要。在众多数据可视化工具中，R与Python都是数据分析与可视化的常用工具，而本书系统地介绍了使用R语言绘制各种图表的方法。相信，本书能够极好地帮助学术科研人员展示数据分析的结果。

李平 / 澳门科技大学资讯科技学院助理教授

R是一门用于统计计算和作图的语言，数据科学工作者钟爱的工具。市面上不乏详细讲述R语言的书籍；而本书以R语言为背景，通俗易懂地讲述了大量数据可视化领域的专业知识，从图形语法、色彩原理、视觉通道到各类图表的应用实践。张杰的这本书让人眼前一亮。

林峰 / 数据可视化技术专家，AntV产品架构师，ECharts 创始人

博文视点Broadview
新浪微博 weibo.com
@博文视点Broadview

责任编辑：石　倩
封面设计：侯士卿

上架建议：数据可视化

ISBN 978-7-121-36366-5

定价：109.00元