© 黄湘云

现代应用统计与 R 语言

Modern Applied Statistics with R

黄湘云

2022-04-13

© 黄湘云

目录

欢迎 1
- 本书风格 1
- 本书定位 2
- 内容概要 2
- 致谢名单 3
- 授权说明 3
- 运行信息 3

第一章 前言 4
- 1.1 语言抉择 4
- 1.2 数据科学 6
- 1.3 获取帮助 7
- 1.4 写作环境 7
- 1.5 记号约定 8
- 1.6 复现环境 9
- 1.7 如何发问 9
- 1.8 作者简介 10

第一部分 数据整理 11

介绍 12

第二章 数据结构 13
- 2.1 类型 13
- 2.2 字符 15
- 2.3 向量 15
- 2.4 矩阵 15
- 2.5 数组 15
- 2.6 表达式 15
- 2.7 列表 15
- 2.8 日期 16
- 2.9 空值 21

第三章 数据搬运 22
- 3.1 导入数据 22
  - 3.1.1 `scan` 23
  - 3.1.2 `read.table` 24
  - 3.1.3 `readLines` 26
  - 3.1.4 `readRDS` 27
- 3.2 其它数据格式 28
- 3.3 导入大数据集 31
- 3.4 从数据库导入 31
  - 3.4.1 PostgreSQL 31
  - 3.4.2 MySQL 34
  - 3.4.3 Spark 35
- 3.5 批量导入数据 40
- 3.6 批量导出数据 41
- 3.7 导出数据 42
  - 3.7.1 导出运行结果 42
  - 3.7.2 导出数据对象 43
- 3.8 Spark 与 R 语言 46
  - 3.8.1 sparklyr 46
  - 3.8.2 SparkR 51
- 3.9 数据库与 R 语言 51
- 3.10 批量读取 csv 文件 52
- 3.11 批量导出 xlsx 文件 54
- 3.12 运行环境 54

第四章 字符串操作 56
- 4.1 字符数统计 56
- 4.2 字符串翻译 57
- 4.3 字符串连接 58
- 4.4 字符串拆分 58
- 4.5 字符串匹配 60
- 4.6 字符串查询 63
- 4.7 字符串替换 70
- 4.8 字符串提取 70
- 4.9 命名捕捉 72
- 4.10 精确匹配 74
- 4.11 模糊匹配 74
- 4.12 高级的替换 76
- 4.13 高级的提取 77
- 4.14 其它操作 78

© 黄湘云

4.14.1 strwrap 78
4.14.2 strtrim 84
4.14.3 strrep 85
4.14.4 trimws 85
4.14.5 tolower 86
4.15 字符串加密 86
4.16 处理性能 87
4.17 网络爬虫 87
4.18 文本挖掘 88
4.19 运行环境 88

第五章 正则表达式 90
5.1 字符常量 91
5.2 软件环境 92
5.3 基本概念 95
5.4 字符串匹配 96
5.5 级联表达式 96
5.6 反向引用 96
5.7 命名捕捉 97
5.8 表达式注释 98

第六章 数据操作 100
6.1 查看数据 101
6.2 提取子集 103
6.3 数据重塑 106
6.4 数据转换 109
6.5 按列排序 109
6.6 数据拆分 111
6.7 数据合并 113
6.8 数据去重 116
6.9 数据缺失 118
6.10 数据聚合 120
6.11 表格统计 128
6.12 索引访问 131
6.13 多维数组 131
6.14 其它操作 133
6.14.1 列表属性 133
6.14.2 堆叠向量 134
6.14.3 属性转化 135
6.14.4 绑定环境 136
6.14.5 数据环境 136
6.15 apply 族 140
6.16 with 选项 144
6.17 分组聚合 145
6.18 合并操作 148
6.19 长宽转换 149
6.20 对符合条件的列操作 150
6.21 `CASE WHEN` 和 `fcase` 151
6.22 数据操作实战 152
6.23 高频数据操作 152
6.23.1 循环合并 153
6.23.2 分组计数 153
6.23.3 分组抽样 153
6.23.4 分组排序 154

第七章 高级数据操作 158
7.1 基础介绍 158
7.1.1 过滤 161
7.1.2 变换 162
7.1.3 聚合 163
7.1.4 命名 164
7.1.5 排序 165
7.1.6 变形 166
7.1.7 分组 168
7.1.8 合并 169
7.2 高频操作 170
7.2.1 选择多列 171
7.2.2 过滤多行 172
7.2.3 去重多行 173
7.2.4 合并操作 174
7.2.5 新添多列 174
7.2.6 删除多列 174
7.2.7 修改多列类型 175
7.2.8 取每组第一行 175
7.2.9 计算环比同比 176
7.2.10 合并多个数据框 177
7.2.11 分组聚合多个指标 180
7.2.12 重命名多个列 181
7.2.13 对多个列依次排序 182
7.2.14 重排多个列的位置 182
7.2.15 整理回归结果 182
7.2.16 `:=` 和 `.()` 184
7.2.17 去掉含有缺失值的记录 . . . 185
7.2.18 集合操作 186
7.3 运行环境 186

第八章 并行化操作 188
8.1 apply 188
8.2 MapReduce 188
8.3 parallel 189
8.4 Rmpi 189
8.5 gpuR 190

© 黄湘云

8.6 运行环境 191

第九章 净土化操作 193

9.1 常用操作 194
9.1.1 查看 194
9.1.2 筛选 195
9.1.3 排序 196
9.1.4 聚合 197
9.1.5 合并 197
9.1.6 变换 198
9.1.7 去重 199
9.2 高频问题 200
9.2.1 初始化数据框 200
9.2.2 移除缺失记录 201
9.2.3 数据类型转化 204
9.2.4 跨列分组求和 204
9.3 管道操作 205
9.4 运行环境 206

第二部分 统计图形 208

介绍 209

第十章 图形基础 210

10.1 绘图基本要素 210
10.1.1 点线 210
10.1.2 区域 216
10.1.3 参考线 222
10.1.4 坐标轴 223
10.1.5 刻度线 233
10.1.6 标题 235
10.1.7 注释 235
10.1.8 图例 239
10.1.9 边空 244
10.1.10 图层 252
10.1.11 布局 254
10.1.12 组合 255
10.1.13 分屏 258
10.1.14 交互 259
10.2 基础统计图形 259
10.2.1 条形图 259
10.2.2 直方图 266
10.2.3 密度图 270
10.2.4 经验图 274
10.2.5 QQ 图 274
10.2.6 时序图 276
10.2.7 饼图 276
10.2.8 茎叶图 278
10.2.9 散点图 278
10.2.10 抖动图 286
10.2.11 箱线图 288
10.2.12 残差图 292
10.2.13 提琴图 292
10.2.14 轮廓图 292
10.2.15 折线图 292
10.2.16 函数图 293
10.2.17 马赛克图 295
10.2.18 点图 295
10.2.19 矩阵图 295
10.2.20 雷达图 297
10.2.21 玫瑰图 297
10.2.22 透视图 299
10.3 栅格统计图形 300
10.3.1 箱线图 300
10.3.2 折线图 301
10.3.3 平滑图 309
10.3.4 点图 312
10.3.5 阶梯图 313
10.3.6 分面图 314
10.3.7 等高线图 315
10.3.8 透视图 316
10.3.9 聚类图 316
10.4 运行环境 319

第十一章 数据可视化 321

11.1 元素 323
11.1.1 图层 323
11.1.2 标签 324
11.1.3 注释 325
11.1.4 刻度 328
11.1.5 图例 329
11.1.6 坐标系 329
11.1.7 坐标轴 329
11.1.8 配色 330
11.1.9 主题 331
11.1.10 布局 332
11.2 字体 333
11.2.1 系统字体 334
11.2.2 思源字体 339
11.2.3 数学字体 341
11.2.4 TikZ 设备 344

© 黄湘云

11.2.5 漫画字体 346
11.2.6 表情字体 346
11.3 配色 348
11.3.1 调色板 351
11.3.2 颜色模式 364
11.3.3 LaTeX 配色 370
11.3.4 ggplot2 配色 371
11.4 图库 371
11.4.1 饼图 371
11.4.2 地图 374
11.4.3 热图 376
11.4.4 散点图 379
11.4.5 条形图 390
11.4.6 直方图 404
11.4.7 箱线图 406
11.4.8 函数图 411
11.4.9 密度图 411
11.4.10 提琴图 419
11.4.11 抖动图 421
11.4.12 蜂群图 430
11.4.13 玫瑰图 430
11.4.14 瓦片图 433
11.4.15 日历图 434
11.4.16 岭线图 437
11.4.17 椭圆图 438
11.4.18 Q-Q 图 440
11.4.19 包络图 440
11.4.20 拟合图 443
11.4.21 地形图 444
11.4.22 树状图 445
11.4.23 留存图 448
11.4.24 瀑布图 449
11.4.25 桑基图 450
11.4.26 词云图 451
11.4.27 甘特图 452
11.4.28 马赛克图 452
11.4.29 凹凸图 452
11.4.30 水流图 454
11.4.31 时间线 455
11.4.32 三元图 457
11.4.33 向量场图 457
11.4.34 四象限图 457
11.4.35 韦恩图 458
11.4.36 龙卷风图 458
11.4.37 聚类图 459
11.4.38 主成分图 460
11.4.39 组合图 461
11.4.40 动态图 463

第十二章 交互图形 467
12.1 散点图 470
12.2 条形图 470
12.3 折线图 471
12.4 双轴图 471
12.5 直方图 473
12.6 箱线图 473
12.7 提琴图 473
12.8 气泡图 473
12.9 曲线图 475
12.10 堆积图 475
12.11 热力图 475
12.12 地图 I 475
12.13 拟合图 477
12.14 轨迹图 478
12.15 三维图 (plotly) 479
12.16 甘特图 480
12.17 帕雷托图 481
12.18 时间线 482
12.19 漏斗图 483
12.20 雷达图 484
12.21 瀑布图 484
12.22 树状图 485
12.23 旭日图 485
12.24 调色板 485
12.25 时序图 486
12.26 导出静态图形 487
12.27 静态图形转交互图形 487
12.28 地图 II 488
12.29 动画 491
12.30 三维图 (rgl) 493
12.31 网络图 494
12.31.1 networkD3 494
12.31.2 visNetwork 495
12.31.3 r2d3 495
12.32 Python 交互图形 496
12.33 运行环境 497

第三部分 动态文档 499

介绍 500

© 黄湘云

第十三章 文档元素 **502**
13.1 控制选项 502
13.2 Markdown 504
13.2.1 列表 504
13.2.2 引用 506
13.2.3 表格 506
13.2.4 图片 507
13.2.5 公式 508
13.3 表格 512
13.4 流程图 512
13.5 编程语言引擎 513
13.6 快速创建书籍项目 514
13.7 Markdown 生态系统 514
13.8 R Markdown 生态系统 515
13.9 支持网页图形 516
13.10 支持 Shiny App 516

第十四章 便携式文档 **518**
14.1 文档汉化 518
14.2 添加水印 518
14.3 双栏排版 518
14.4 参数化报告 518
14.5 学术幻灯片 519
14.6 文档模版 519
14.7 引用文献 519
14.8 自定义块 520

第十五章 网页文档 **522**
15.1 幻灯片 522
15.2 电子邮件 522

第十六章 办公文档 **525**

第十七章 工作流 **526**

第十八章 高级文档 **527**
18.1 编写书籍 527
18.2 个人网站 527
18.3 R 包文档 527
18.4 课程网站 527

第四部分 数据产品 **528**

介绍 **529**

第十九章 交互表格 **530**
19.1 DT 和 reactable 530
19.2 gt 和 kableExtra 535
19.3 运行环境 537

第二十章 交互报表 **539**
20.1 开发流程 539
20.2 开发工具 542
20.3 基础知识 542
20.4 基础组件 542
20.4.1 书签 542
20.4.2 表格 543
20.5 高级主题 547
20.6 部署应用 548
20.7 最佳实践 548
20.8 仪表盘 548
20.9 交互式数据报表 dash 553
20.10 运行环境 553

第五部分 统计基础 **555**

介绍 **556**

第二十一章 抽样分布 **557**
21.1 正态分布 557
21.2 指数族 557

第二十二章 参数估计 **561**
22.1 点估计 561
22.1.1 矩估计 562
22.1.2 最小二乘估计 562
22.1.3 极大似然估计 565
22.2 区间估计 566
22.2.1 正态分布 566
22.2.2 0-1 分布 567
22.2.3 置信区间和信仰区间 569
22.3 最小角回归 577
22.4 刀切法 577
22.5 重抽样 577
22.6 Delta 方法 577

第二十三章 假设检验 **578**
23.1 Ansari-Bradley 检验 `ansari.test` . 581
23.2 Bartlett 检验 `bartlett.test` 581
23.3 二项检验 `binom.test` 582
23.4 时间序列独立性检验 `Box.test` . . . 583
23.5 皮尔逊卡方检验 `chisq.test` 583
23.6 费舍尔精确检验 `fisher.test` 583

© 黄湘云

23.7 方差齐性检验 fligner.test 584
23.8 Friedman 秩和检验 friedman.test 584
23.9 Kruskal-Wallis 秩和检验 kruskal.test 584
23.10 同分布检验 ks.test 585
23.11 Cochran-Mantel-Haenszel 卡方检验 mantelhaen.test 585
23.12 Mauchly 球形检验 mauchly.test . 585
23.13 McNemar 卡方检验 mcnemar.test . 586
23.14 Mood 方差检验 mood.test 586
23.15 单因素多重比较 oneway.test . . . 586
23.16 配对样本的检验 587
23.16.1 配对比例检验 pairwise.prop.test 587
23.16.2 配对 t 检验 pairwise.t.test 589
23.16.3 配对 Wilcoxon 检验 pairwise.wilcox.test 589
23.16.4 配对样本相关性检验 cor.test 589
23.17 精确泊松检验 poisson.test 590
23.18 单位根检验 PP.test 590
23.19 比例检验 prop.test 590
23.19.1 两个独立二项总体等价性检验 591
23.19.2 不同页面的点击率问题 . . . 592
23.19.3 比例齐性检验 593
23.20 比例趋势检验 prop.trend.test . . 595
23.21 Quade 检验 quade.test 595
23.22 正态性检验 shapiro.test 595
23.23 正态性检验 Epps-Pully 检验 596
23.24 学生 t 检验 t.test 596
23.24.1 正态总体两样本的均值之差的检验 596
23.24.2 办公软件里的 T 检验 601
23.25 方差比检验 var.test 602
23.26 Wilcoxon 秩和检验 wilcox.test . 603
23.26.1 ROC 曲线和 wilcox.test 检验的关系 603
23.27 3 + 1 统计检验 605
23.28 经典案例 606
23.28.1 1973 年加州大学伯克利分校的学生招生 606
23.28.2 1976~1977 年美国佛罗里达州的凶杀案件中被告肤色和死刑判决的关系 606
23.28.3 统计专业学生的头发和眼睛的颜色 607
23.29 运行环境 607

第二十四章 功效分析 610
24.1 方差分析检验的功效 610
24.2 比例检验的功效 611
24.3 t 检验的功效 613
24.4 运行环境 616

第二十五章 试验设计 618
25.1 学生睡眠质量 618
25.2 驱虫喷雾的效果 619
25.3 重复数不等的多重比较 626
25.4 不同地区的草类植物吸收二氧化碳的情况 627
25.5 果园喷雾剂的效力 627
25.6 验证孟德尔的豌豆实验结果 627

第六部分 统计模型 629

介绍 630

第二十六章 线性模型 631
26.1 方差分析 631
26.2 单因素方差分析 631
26.3 双因素方差分析 636
26.4 多因素方差分析 636
26.5 核学习 636
26.6 通用机器学习 636
26.7 理论基础 637
26.8 多重多元线性回归 637
26.9 回归诊断 638
26.10 1977 年美国人口普查 639
26.11 石油岩石样品的测量 641
26.12 1888 年瑞士生育率分析 641
26.13 Intercountry Life-Cycle Savings Data 1960-1970 645
26.14 Longley's Economic Regression Data 1947-1962 645
26.15 甲醛的测定 645
26.16 迈克尔逊光速数据分析 646
26.17 不同喂食方式对小鸡体重的影响 I . 647
26.18 不同喂食方式对小鸡体重的影响 II . 647
26.19 酶的酶联免疫吸附测定 650
26.20 婴儿的体重随年龄的变化情况 . . . 651
26.21 火炬松树的生长情况 656
26.22 酶促反应的反应速率 658
26.23 茶碱的药代动力学 659
26.24 本章总结 662

© 黄湘云

26.25 运行环境 662

第二十七章 广义线性模型 664
27.1 介绍 664
27.2 理论基础 665
27.2.1 岭回归 665
27.2.2 Lasso 665
27.2.3 最优子集回归 665
27.2.4 偏最小二乘回归 665
27.3 吸烟喝酒和食道癌的关系 666
27.4 自然流产和人工流产后的不育 . . . 667
27.5 细菌数据集 670
27.6 研究婴儿出生体重低的相关危险因素 672
27.7 哥本哈根住房状况调查 676
27.8 癫痫病发作次数 678
27.9 对数线性模型 679
27.10 泊松回归模型 679

第七部分 数据建模 687

介绍 688

第二十八章 文本分析 689

第二十九章 生存分析 690
29.1 急性粒细胞白血病生存数据 690

第三十章 时序分析 692
30.1 时序数据 693
30.2 时序图 695
30.3 基本概念 697
30.4 时序检验 700
30.5 指数平滑 701
30.6 Holt-Winters 701
30.7 1749-2013 年太阳黑子数据 702
30.8 1991-1998 年欧洲主要股票市场日闭市价格指数 704
30.9 自回归模型 707
30.10 移动平均模型 707
30.11 自回归移动平均模型 707
30.12 自回归条件异方差模型 707
30.13 广义自回归条件异方差模型 708
30.14 其它特征的时间序列 708
30.15 港股走势 708
30.16 美股走势 709
30.17 51Talk 股价走势 709
30.18 运行环境 710

第八部分 时空数据 713

介绍 714

第三十一章 空间数据分析 715

第三十二章 空间数据可视化 717
32.1 空间数据 717
32.1.1 raster 717
32.2 可视化 720
32.2.1 斐济地震带分布 720
32.3 美国各个城镇的失业率分布 721
32.3.1 maps 721
32.3.2 latticeExtra 723
32.3.3 ggplot2 723
32.3.4 sf 725
32.3.5 mapsf 727

第三十三章 案例研究 728
33.1 统计学家生平 729
33.2 R 语言发展历史 729
33.3 不同实验条件下植物生长情况 . . . 729
33.4 橘树生长情况 737
33.5 R 包网络分析 740
33.5.1 R 核心团队 740
33.5.2 高产的开发者 744
33.5.3 社区开发者 746
33.5.4 首次贡献 R 包 750
33.5.5 贡献关系网络 752
33.5.6 更新知多少 755
33.5.7 使用许可证 756

第三十四章 数据探索 759

第九部分 机器学习 760

介绍 761

第三十五章 梯度提升机 762
35.1 XGBoost 762

第三十六章 数值优化 763
36.1 线性规划 764
36.2 整数规划 766
36.2.1 一般整数规划 766

© 黄湘云

36.2.2 0-1 整数规划 766
36.2.3 混合整数规划 767
36.3 二次规划 769
36.3.1 凸二次规划 769
36.3.2 半正定二次优化 772
36.4 非线性规划 773
36.4.1 一元非线性优化 773
36.4.2 多元非线性无约束优化 . . . 774
36.4.3 多元非线性约束优化 793
36.5 非线性方程 809
36.5.1 一元非线性方程 809
36.5.2 非线性方程组 810
36.6 多目标规划 814
36.7 经典优化问题 814
36.8 回归与优化 814
36.9 对数似然 816
36.10 微分方程 818
36.10.1 常微分方程 818
36.10.2 偏微分方程 819
36.10.3 延迟微分方程 824
36.10.4 随机微分方程 824
36.11 运行环境 824

附录 A 命令行操作 827
A.1 查看文件 827
A.2 创建文件夹 828
A.3 移动文件 828
A.4 查看文件大小 829
A.5 终端模拟器 829
A.6 压缩和解压缩 831
A.7 从仓库安装 R 831
A.7.1 Ubuntu 831
A.7.2 CentOS 832
A.8 源码安装 832
A.8.1 Ubuntu 832
A.8.2 CentOS 832
A.9 忍者安装 835
A.10 配置 836
A.10.1 初始会话 `.Rprofile` 836
A.10.2 环境变量 `.Renviron` 836
A.10.3 编译选项 `Makevars` 836
A.11 命令行参数 836
A.12 从源码安装 R 837
A.13 安装软件 838
A.14 安装 R 包 839
A.15 软件包管理器 842
A.15.1 dnf 842
A.15.2 apt 843

附录 B 矩阵运算 845
B.1 矩阵乘法 846
B.2 Hadamard 积 847
B.3 矩阵转置 848
B.4 矩阵外积 848
B.5 矩阵乘方 849
B.6 矩阵求幂 850
B.7 矩阵交叉积 850
B.8 矩阵行列式 851
B.9 矩阵条件数 851
B.10 矩阵求逆 851
B.11 矩阵伴随 853
B.12 矩阵范数 853
B.13 矩阵求秩 854
B.14 矩阵求迹 854
B.15 单位矩阵 854
B.16 对角矩阵 855
B.17 上/下三角矩阵 855
B.18 稀疏矩阵 857
B.19 三对角矩阵 857
B.20 LU 分解 857
B.21 Schur 分解 857
B.22 Cholesky 分解 857
B.23 特征值分解 858
B.24 SVD 分解 858
B.25 QR 分解 859
B.26 Jordan 分解 861
B.27 Givens 旋转 861
B.28 特殊函数 861
B.28.1 阶乘 861
B.28.2 伽马函数 862
B.28.3 贝塔函数 863
B.28.4 贝塞尔函数 863

附录 C 符号计算 865

附录 D 混合编程 869
D.1 函数源码 869
D.2 命名约定 871
D.3 R 与 JavaScripts 871
D.4 R 与 Python 871
D.5 R 与 C 872

© 黄湘云

D.6 R 与 C++ 872
D.7 R 与 LaTeX 873
D.8 运行环境 874

附录 E 面向对象编程 876
E.1 环境 876
E.2 引用 877
E.3 调用栈 877
E.4 闭包 877
E.5 递归 878
E.6 异常 878
E.7 对象 878
E.8 泛型 878
E.9 除虫 879
E.10 性能 879
E.11 质量 879

附录 F 文件操作 881
F.1 查看文件 881
F.2 操作文件 885
F.3 压缩文件 886
F.4 路径操作 887
F.5 查找文件 888
F.6 文件权限 890
F.7 区域设置 890
F.8 进程管理 892
F.9 系统命令 893
F.10 时间管理 893
F.11 R 包管理 896
F.12 查找函数 900
F.13 运行环境 901

附录 G 其它软件 903
G.1 文本编辑器 903
G.2 代码编辑器 904
G.3 集成开发环境 904
G.3.1 RStudio 桌面版 904
G.3.2 RStudio 服务器版 905
G.3.3 Shiny 服务器版 907
G.3.4 Eclipse + StatET 907
G.3.5 Emacs + ESS 907
G.3.6 Nvim-R 908
G.4 Git 版本控制 908
G.4.1 安装配置 909
G.4.2 追踪文件 910
G.4.3 合并上流 911
G.4.4 大文件支持 911
G.4.5 新建分支 912
G.4.6 创建 Github Pages 站点 . . . 912
G.4.7 博客主题 912
G.4.8 修改远程仓库的位置 913
G.4.9 统计代码仓库的提交量 . . . 913
G.4.10 账户共存 914
G.4.11 回车换行 915
G.4.12 子模块 915
G.4.13 克隆项目 915
G.4.14 创建 PR 915
G.4.15 修改 PR 916
G.5 Pandoc 文档处理 916
G.6 Calibre 书籍管理 917
G.7 ImageMagick 图像处理 917
G.8 OptiPNG 图片优化 918
G.9 PDFCrop 裁剪边空 919
G.10 PhantomJS 网页截图 919
G.11 Inkscape 矢量绘图 920
G.12 QPDF PDF 文件操作 921
G.13 UML 标准建模图 921
G.14 Graphviz 流程图 922
G.15 LaTeX 排版工具 924
G.15.1 TinyTeX 发行版 924
G.15.2 安装和更新 925
G.15.3 查询和搜索 925
G.15.4 TikZ 绘图工具 926
G.16 Octave 科学计算 926
G.17 Python 环境配置 927
G.18 Python 基础绘图 928
G.19 Python 基础操作 930
G.20 VBox 虚拟机 938
G.20.1 从命令行启动虚拟机 938
G.21 Docker 虚拟环境 939
G.22 安装的 R 包 943

附录 H 符号说明 960

© 黄湘云

插图

© 黄湘石

表格

© 黄湘云

欢迎

警告

Book in early development. Planned release in 202X.

本书风格

可以说，点估计、区间估计、假设检验、统计功效是每一个学数理统计的学生都绕不过去的坎，离开学校从事数据相关的工作，它们仍然是必备的工具。所以，本书会覆盖相关内容，但是和高校的教材最大的区别是更加注重它们之间的区别和联系，毕竟每一个统计概念都是经过了千锤百炼，而我们的主流教材始终如一地遵循的一个基本套路，就是突然给出一大堆定义、命题或定理，紧接着冗长的证明过程，然后给出一些难以找到实际应用背景的例子。三板斧抡完后就是给学生布置大量的习题，这种教学方式无论对于立志从事理论工作的还是将来投身于工业界的学生都是不合适的。

> 极大似然估计最初由德国数据学家 Gauss 于 1821 年提出，但未得到重视，后来，R. A. Fisher 在 1922 年再次提出极大似然的思想，探讨了它的性质，使它得到广泛的研究和应用。[茆诗松 et al., 2006]

这是国内某著名数理统计教材在极大似然估计开篇第一段的内容，后面是各种定义、定理、公式推导。教材简短一句话，这里面有很多信息值得发散，一个数学家提出了统计学领域极其重要的一个核心思想，他是在研究什么的时候提出了这个想法，为什么后来没有得到重视，整整 100 年以后，Fisher 又是怎么提出这一思想的呢？他做了什么使得这个思想被广泛接受和应用？虽然这可能有点离题，但是读者可以获得很多别的启迪，要知道统计领域核心概念的形成绝不是一蹴而就的，这一点也绝不局限于统计科学，任何一门科学都是这样的，比如物理学之于光的波粒二象性。历史上，各门各派的学者历经多年的思想碰撞才最终沉淀出现在的结晶。笔者认为，学校要想培养出有原创理论创新的人才，在对待前辈的成果上，我们要不吝笔墨和口水，传道不等于满堂灌和刷分机，用寥寥数节课或者数页纸来梳理学者们几十年乃至上百年的智慧结晶是非常值得的，我们甚至可以从当时的社会、人文去剖析。非常欣赏有人在收集关于统计学历史的材料，读者不妨去看看 https://github.com/sctyner/history_of_statistics。另一个不得不提的人是 Allison Horst，她以风趣幽默的漫画形式，以画龙点睛之手法勾勒出基本的统计概念和思想，详见 https://github.com/allisonhorst/stats-illustrations，是我见过最好的科普读物。

Bradley Efron 在他的课程中谈及现代统计的研究层次，第一层次是基于正态分布假设的，这种类型已经研究的很清楚了，往往可以得到精确的结果，第二层次是将正态分布推广到指数族，这种类型的也研究的比较多了，常见的情况都研究的比较清楚，罕见的情况也是大量存在的，特别是在实际应用当中，总的来说只能得到部分精确的结果，第三层次对分布没有任何限定，只要满足成为一个统计分布的条件，这种情况下就只能求助于一般的数学工具和渐进理论。

© 黄湘云

图 1: 现代统计建模的三重境界：修改自 2019 年冬季 Bradley Efron 的课程笔记（第一部分）http://statweb.stanford.edu/~ckirby/brad/STATS305B_Part-1_corrected-2.pdf

本书定位

学习本书需要读者具备基本的概率、统计知识，比如上过一学期的概率论和数理统计学，也需要读者接触过编程知识，比如至少上过一学期的 C 语言、Python 语言或 Matlab 语言。了解基本的线性代数，比如矩阵的加、减、乘、逆四则运算、线性子空间、矩阵的 LU、SVD、Eigen 等分解。

内容概要

第 一 章介绍本书的写作背景、语言环境、全书的记号约定、如何获取帮助、作者简介等信息。

第 二 章介绍 R 语言的数据结构。

第 六 章介绍数据操作，包括 Base R 、**data.table** 和 **magrittr**。

第 三 章介绍数据导入导出，**data.table** 之于 csv 文件，**openxlsx** 之于 xlsx 文件。

第 十一 章介绍数据可视化，分四个部分，基础元素、常用图形、字体和颜色设置。

第 十三 章介绍动态文档，即 R Markdown 及其生态系统。

第 十二 章介绍交互图形，以常用的 **plotly** 和 **highcharter** 为主，重点介绍 R 和 JavaScript 库的对应关系。

第 十九 章介绍交互表格，分两节介绍交互式的 **DT** 和 **reactable**，静态的 **gt** 和 **kableExtra**，掌握这几个 R 包足以应付日常工作。

第 二十 章介绍交互报表开发，符合工业标准的最佳实践。

© 黄湘云

第 H 章介绍全书的数学公式符号。

第 F 章介绍文件操作。

致谢名单

特别感谢 XX，还有很多人通过提交 PR 或 Issues 的方式参与了本书的创作过程，没有这一点一滴的持续改进，本书不会达到现在的样子。

黄湘云
于北京

授权说明

警告

本书采用 知识共享署名-非商业性使用-禁止演绎 4.0 国际许可协议 许可，请君自重，别没事儿拿去传个什么新浪爱问、百度文库以及 XX 经济论坛，项目中代码使用 MIT 协议 开源

运行信息

书籍在 R version 4.1.3 (2022-03-10) 下编译，Pandoc 版本 2.16.2，最新一次编译发生在 2022-04-13 15:25:15。

© 高等教育出版社

第一章　前言

> 荃者所以在鱼，得鱼而忘荃；蹄者所以在兔；得兔而忘蹄；言者所以在意，得意而忘言。吾安得夫忘言之人而与之言哉！
>
> — 摘自《庄子 · 杂篇 · 物》

庄子谈学习，余深以为然，故引之。

> The fish trap exists because of the fish; once you've gotten the fish, you can forget the trap. The rabbit snare exists because of the rabbit; once you've gotten the rabbit, you can forget the snare. Words exist because of meaning; once you've gotten the meaning, you can forget the words. Where can I find a man who has forgotten words so I can have a word with him ? [1]
>
> — Chuang Tzu

1.1　语言抉择

行业内可以做统计分析和建模的软件汗牛充栋，比较顶级的收费产品有 SAS 和 SPSS，在科学计算领域的 Matlab 和 Mathematica 也有相当强的统计功能，而用户基数最大的是微软 Excel，抛开微软公司的商业手段不说，Excel 的市场份额却是既成事实。Brian D. Ripley 20 多年前的一句话很有意思，放在当下也是适用的。

> Let's not kid ourselves: the most widely used piece of software for statistics is Excel.
>
> — Brian D. Ripley [Ripley, 2002]

有鉴于 Excel 在人文、社会、经济和管理等领域的影响力，熟悉 R 语言的人把它看作超级收费版的 Excel，这实在是一点也不过分。事实上，我司就是一个很好的明证，一个在线教育类的互联网公司，各大业务部门都在使用 Excel 作为主要的数据分析工具。然而，Excel 的不足也十分突出，工作过程无法保存和重复利用，Excel 也不是数据库，数据集稍大，操作起来愈发困难，对于复杂的展示，需要借助内嵌的 VBA，由于缺乏版本控制，随着时间的推移，几乎不可维护。所以，我们还是放弃 Excel 吧，Jenny Bryan 更在 2016 年国际 R 语言大会上的直截了当地喊出了这句话[2]。Nathan Stephens 对 Excel 的缺陷不足做了全面的总结[3]。

> Some people familiar with R describe it as a supercharged version of Microsoft's Excel spreadsheet software.

[1] 译文摘自 Eric D. Kolaczyk

[2] https://channel9.msdn.com/Events/useR-international-R-User-conference/useR2016/jailbreakr-Get-out-of-Excel-free

[3] https://resources.rstudio.com/wistia-rstudio-essentials-2/how-to-excel-without-using-excel

© 王沛勇

— Ashlee Vance [4]

另一方面，我们谈谈开源领域的佼佼者 — R (https://cran.r-project.org/)，Python (https://www.python.org/) 和 Octave (http://www.gnu.org/software/octave/)。Python 号称万能的胶水语言，从系统运维到深度学习都有它的广泛存在，它被各大主流 Linux 系统内置，语言风格上更接近于基数庞大的开发人员，形成了强大的生态平台。Octave 号称是可以替代 Matlab 的科学计算软件，在兼容 Matlab 的方面确实做的很不错，然而，根据 Julia 官网给出的各大编程语言的测试 https://julialang.org/benchmarks/，性能上不能相提并论。

图 1.1: R 语言扩展包生态系统

R 提供了丰富的图形接口，包括 Tcl/Tk , Gtk, Shiny 等，以及基于它们的衍生品 rattle（RGtk2）、Rcmdr（tcl/tk）、radiant（shiny）。更多底层介绍，见 John Chamber 的著作《Extending R》。

Eviews 时间序列和计量经济模型，相比于 Eviews，Stata 是综合型的统计软件，提供丰富的统计模型，SPSS 同 Stata 类似，Minitab，JASP 是开源的软件，Octave 是对标 Matlab 的工程计算软件，有丰富的优化功能，是一门编程语言兼软件，为求解统计模型的参数提供了广泛的基础能力。Tableau 提供强大的分析和打造数据产品的能力。TikZ 在绘制示意图方面有很大优势，特别是示意图里包含数学公式，这更是 LaTeX 所擅长的方面。

JASP https://jasp-stats.org 是一款免费的统计软件，源代码托管在 Github 上 https://github.com/jasp-stats/jasp-desktop，主要由阿姆斯特丹大学 E. J. Wagenmakers 教授 https://www.ejwagenmakers.com/

[4] https://www.nytimes.com/2009/01/07/technology/business-computing/07program.html

© 黄湘云

领导的团队维护开发，实现了很多贝叶斯和频率统计方法，相似的图形用户界面使得 JASP 可以作为 SPSS 的替代，目前实现的功能见 https://jasp-stats.org/current-functionality/，统计方法见博客 https://www.bayesianspectacles.org/。

国内可视化分析平台，比如 hiplot 基于 R 语言实现可视化分析，各类图形的介绍见文档，极大地降低数据分析人员探索分析的门槛，节省了时间，同时非专业内的人也可借助其完成分析探索的过程，只需明白各类图形的含义即可。美团也建设了自己的可视化分析平台帮助运营人员，详见文档

Patrick Burns 收集整理了 R 语言中奇葩的现象，写成 The R Inferno 直译过来就是《R 之炼狱》。这些奇葩的怪现象可以看做是 R 风格的一部分，对于编程人员来说就是一些建议和技巧，参考之可以避开某些坑。Paul E. Johnson 整理了一份真正的 R 语言建议，记录了他自己从 SAS 转换到 R 的过程中遇到的各种问题 http://pj.freefaculty.org/R/Rtips.html。Michail Tsagris 和 Manos Papadakis 也收集了 70 多条 R 编程的技巧和建议，力求以更加 R 范地将语言特性发挥到极致 [Tsagris and Papadakis, 2018]，Martin Mächler 提供了一份 Good Practices in R Programming。Python 社区广泛流传着 Tim Peters 的《Python 之禅》，它已经整合进每一版 Python 软件中，只需在 Python 控制台里执行 `import this` 可以获得。

1. Beautiful is better than ugly.
2. Explicit is better than implicit.
3. Simple is better than complex.
4. Complex is better than complicated.
5. Flat is better than nested.
6. Sparse is better than dense.
7. Readability counts.
8. Special cases aren't special enough to break the rules.
9. Although practicality beats purity.
10. Errors should never pass silently.
11. Unless explicitly silenced.
12. In the face of ambiguity, refuse the temptation to guess.
13. There should be one– and preferably only one –obvious way to do it.
14. Although that way may not be obvious at first unless you're Dutch.
15. Now is better than never.
16. Although never is often better than *right* now.
17. If the implementation is hard to explain, it's a bad idea.
18. If the implementation is easy to explain, it may be a good idea.
19. Namespaces are one honking great idea – let's do more of those!

— The Zen of Python

总之，编程语言到一定境界都是殊途同归的，对美的认识也是趋同的，道理更是相通的，Python 社区的 Pandas https://github.com/pandas-dev/pandas 和 Matplotlib https://github.com/matplotlib/matplotlib 也有数据框和图形语法的影子。Pandas https://github.com/pandas-dev/pandas 明确说了要提供与 data.frame 类似的数据结构和对应统计函数等，而 Matplotlib 偷了 ggplot2 绘图样式 https://matplotlib.org/3.2.2/gallery/style_sheets/ggplot.html。

1.2 数据科学

John M. Chambers 谈了数据科学的源起以及和 S、R 语言的渊源 [Chambers, 2020]。

© 黄湘云

1.3 获取帮助

R 社区提供了丰富的帮助资源，可以在 R 官网搜集的高频问题 https://cran.r-project.org/faqs.html 中查找，也可在线搜索 https://cran.r-project.org/search.html 或 https://rseek.org/ ，更多获取帮助方式见 https://www.r-project.org/help.html。爆栈网问题以标签分类，比如 r-plotly、r-markdown、data.table 和 ggplot2，还可以关注一些活跃的社区大佬，比如 谢益辉。

1.4 写作环境

图 1.2: 书籍项目架构图

本书 R Markdown 源文件托管在 Github 仓库里，本地使用 RStudio IDE 编辑，bookdown 组织各个章节的 Rmd 文件和输出格式，使用 Git 进行版本控制。每次提交修改到 Github 上都会触发 Travis 自动编译书

© 黄湘云

籍，将一系列 Rmd 文件经 knitr 调用 R 解释器执行里面的代码块，并将输出结果返回，Pandoc 将 Rmd 文件转化为 md 、html 或者 tex 文件。若想输出 pdf 文件，还需要准备 TeX 排版环境，最后使用 Netlify 托管书籍网站，和 Travis 一起实现连续部署，使得每次修改都会同步到网站。最近一次编译时间 2022 年 04 月 13 日 07 时 25 分 18 秒，本书用 R version 4.1.3 (2022-03-10) 编译，完整运行环境如下：

```
xfun::session_info(packages = c(
  "knitr", "rmarkdown", "bookdown"
), dependencies = FALSE)
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Locale:
##   LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##   LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##   LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##   LC_PAPER=en_US.UTF-8       LC_NAME=C
##   LC_ADDRESS=C               LC_TELEPHONE=C
##   LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## Package version:
##   bookdown_0.25  knitr_1.38     rmarkdown_2.13
##
## Pandoc version: 2.16.2
```

借助 **bookdown** [Xie, 2016] 可以将 Rmd 文件组织起来，**rmarkdown** [Allaire et al., 2021] 和 **knitr** [Xie, 2015] 将源文件编译成 Markdown 文件，Pandoc 将 Markdown 文件转化成 HTML 和 TeX 文件，TinyTeX [Xie, 2019] 可以将 TeX 文件进一步编译成 PDF 文档，书中大量的图形在用 **ggplot2** 包制作 [Wickham, 2016]，而统计理论相关的示意图用 Base R 创作。

提示

得益于 Github Action 提供的测试服务，Github Pages、Bookdown 和 Netlify 提供的部署服务，鉴于国内的网络环境，本书托管在三个地方，分别是 https://xiangyunhuang.github.io/masr/，https://bookdown.org/xiangyun/masr/ ，https://masr.netlify.app/。

1.5 记号约定

正文中的代码、函数、参数及参数值以等宽正体表示，如 `data(list = c('iris', 'BOD'))`，其中函数名称 `data()`，参数及参数值 `list = c('iris', 'BOD')` ，R 程序包用粗体表示，如 **graphics**。

```
ruler()
```

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

© 黄湘云

1.6 复现环境

构建容器

本书借助 Github Action 从 Dockerfile 构建容器镜像，然后将镜像文件推送到 Github Package。完成这些操作首先需要从 https://github.com/settings/tokens 新建拥有 GitHub Package [5] 读写删的权限的 TOKEN（俗称访问令牌或密钥），命名为 `GITHUB_PKG`，并将此令牌保存到本地 TOKEN.txt 文件中，以备后用。

镜像内默认暴露 8181 端口供外部连接使用，进入容器后，默认工作路径是 `/home/docker/`。创建好镜像后，要先登陆 GitHub Package 然后才有权限将镜像拉取下来

```
# 登陆 GitHub Package
cat ~/TOKEN.txt | docker login https://docker.pkg.github.com -u XiangyunHuang --password-stdin
# 拉取镜像
docker pull docker.pkg.github.com/xiangyunhuang/masr/masr-book:devel
```

读者可以先查看下容器内的信息

```
docker run --rm -u root -v "/${PWD}://home/docker/" \
  docker.pkg.github.com/xiangyunhuang/masr/masr-book:devel \
  bash -c "locale; fc-list | sort"
```

运行容器

下面从镜像创建一个叫 masr-book 的容器，并让它在后台运行，允许以真正的 root 账户权限交互式执行命令，停止容器后，自动销毁容器。此处，不再介绍 Docker 容器的使用，用容器打包本书所有软件环境仅供读者完整复现本书之用，感兴趣的读者可以去参考书籍Docker 从入门到实践。

```
docker run -itd -p 8282:8787 --rm --name=masr-book --privileged=true \
  docker.pkg.github.com/xiangyunhuang/masr/masr-book:devel /sbin/init
```

接着登陆进入 masr-book 容器，

```
docker exec -it masr-book bash
```

一番骚操作后，用户退出容器，然后停止容器。

```
docker stop masr-book
```

使用 RStudio Server

启动容器后，接着获取容器 masr-book 的 IP 地址，然后依据端口号 8282 从网页进入 RStudio Sever，比如 http://192.168.100.23:8282

```
docker inspect --format='{{.NetworkSettings.IPAddress}}' masr-book
```

1.7 如何发问

The phrase “does not work” is not very helpful, it can mean quite a few things including:

- Your computer exploded.

[5] https://docs.github.com/en/packages/using-github-packages-with-your-projects-ecosystem/configuring-docker-for-use-with-github-packages

© 统计之都

- No explosion, but smoke is pouring out the back and microsoft's "NoSmoke" utility is not compatible with your power supply.
- The computer stopped working.
- The computer sits around on the couch all day eating chips and watching talk shows.
- The computer has started picketing your house shouting catchy slogans and demanding better working conditions and an increase in memory.
- Everything went dark and you cannot check the cables on the back of the computer because the lights are off due to the power outage.
- R crashed, but the other programs are still working.
- R gave an error message and stopped processing your code after running for a while.
- R gave an error message without running any of your code (and is waiting for your next command).
- R is still running your code and the time has exceeded your patience so you think it has hung.
- R completed and returned a result, but also gave warnings.
- R completed your command, but gave an incorrect answer.
- R completed your command but the answer is different from what you expect (but is correct according to the documentation).

There are probably others. Running your code I think the answer is the last one.

— Greg Snow [6]

1.8 作者简介

热心开源事业，统计之都副主编，经常混迹于统计之都论坛、Github 和爆栈网。个人主页 https://xiangyun.rbind.io/

[6] 来自 R 社区论坛收集的智语 `fortunes::fortune(324)`。

© 黄湘云

第一部分

数据整理

© 黄湘云

介绍

数据整理

© 董潇石

第二章　数据结构

网站 https://r-coder.com/ 主要介绍 Base R，特点是全面细致，排版精美

> 可用于绘图的数据对象，向量 vector 等，只涉及基础操作和绘图，关键在入门引导式的介绍，点到即止

数据类型：字符、数值：字符数据操作：按数据类型介绍各类数据操作，重复之处如前所述，数据处理的分类：按数据类型来，一共是 table matrix data.frame 和 vector

> The trouble with nonstandard evaluation is that it doesn't follow standard evaluation rules...
>
> — Peter Dalgaard (about nonstandard evaluation in the `curve()` function) R-help (June 2011)

向量，列表，

数据框 data frame 在 R 里面可以用三种不同类型的数据对象来表达

> 从历史脉络来看，为什么会出现三种不同的东西，它们之间的区别和联系是什么，能否用一张表来描述

data.frame 设计的历史，首次包含在 R 里面是什么时候，R 是否一发布就包含了这个数据类型

The function `data.frame()` creates data frames, tightly coupled collections of variables which share many of the properties of matrices and of lists, used as the fundamental data structure by most of R's modeling software.

data.table 2006 年 4 月 15 日首次登陆 CRAN 发布 1.0 版本，差不多恰好 10 年后

tibble 在 2016 年 3 月 23 日首次登陆 CRAN 发布 1.0 版本

`data.frame()`，`tibble()` 和 `data.table()` 的区别，去看函数的帮助文档

Provides a 'tbl_df' class (the 'tibble') that provides stricter checking and better formatting than the traditional data frame.

vctrs 和 rlang 包 R 内置的 R Language Definition

2.1　类型

```
x <- "abc" # 数据对象
typeof(x) # 数据类型
```

```
## [1] "character"
```

© 草稿

```
mode(x) # 存储模式
```

```
## [1] "character"
```

```
storage.mode(x) # 存储类型
```

```
## [1] "character"
```

表 2.1: 函数 `typeof()` 返回的数据类型 [1]

符号	含义
`NULL`	空值
`symbol`	可变的名称/变量
`pairlist`	pairlist 对象 ***
`closure`	函数闭包
`environment`	环境
`promise`	实现惰性求值的对象
`language`	R 语言构造
`special`	内部函数，不计算参数
`builtin`	内部函数，计算参数
`char`	scalar 型字符对象 ***
`logical`	逻辑向量
`integer`	整数向量
`double`	实值向量
`complex`	复值向量
`character`	字符向量
`...`	可变长度的参数 ***
`any`	匹配任意类型
`expression`	表达式对象
`list`	列表
`bytecode`	位代码 ***
`externalptr`	外部指针对象
`weakref`	弱引用对象
`raw`	位向量
`S4`	S4 对象

表 2.2: R/Rcpp 提供的基本数据类型

Value	R vector	Rcpp vector	Rcpp matrix	Rcpp scalar	C++ scalar
Logical	`logical`	`LogicalVector`	`LogicalMatrix`	-	`bool`
Integer	`integer`	`IntegerVector`	`IntegerMatrix`	-	`int`
Real	`numeric`	`NumericVector`	`NumericMatrix`	-	`double`

[1]表格中带 *** 标记的类型，用户不能轻易获得

©清华大学出版社

Value	R vector	Rcpp vector	Rcpp matrix	Rcpp scalar	C++ scalar
Complex	`complex`	`ComplexVector`	`ComplexMatrix`	`Rcomplex`	`complex`
String	`character`	`CharacterVector` (`StringVector`)	`CharacterMatrix` (`StringMatrix`)	`String`	`string`
Date	`Date`	`DateVector`	-	`Date`	-
Datetime	`POSIXct`	`DatetimeVector`	-	`Datetime`	`time_t`

2.2 字符

2.3 向量

2.4 矩阵

2.5 数组

更多数组操作 rray

2.6 表达式

```
# %||% 中缀符
# x 是空值或者长度为 0 则保留 y 否则保留 x
function(x, y) if (is.null(x) || length(x) == 0) y else x
```

```
## function(x, y) if (is.null(x) || length(x) == 0) y else x
```

2.7 列表

```
x <- list(a = 1, b = 2, c = list(d = c(1, 2, 3), e = "hello"))
print(x)
```

```
## $a
## [1] 1
##
## $b
## [1] 2
##
## $c
## $c$d
```

©李漫漫

```
## [1] 1 2 3
##
## $c$e
## [1] "hello"
```

```
base::print.simple.list(x)
```

```
##      _
## a    1
## b    2
## c.d1 1
## c.d2 2
## c.d3 3
## c.e  hello
```

2.8 日期

注意观察时间转化

```
Sys.Date()
```

```
## [1] "2022-04-13"
```

```
Sys.time()
```

```
## [1] "2022-04-13 07:25:20 UTC"
```

```
c(Sys.time(), Sys.Date())
```

```
## [1] "2022-04-13 07:25:20 UTC" "2022-04-13 00:00:00 UTC"
```

```
data.table::year(Sys.Date())
```

```
## [1] 2022
```

```
data.table::year(Sys.time())
```

```
## [1] 2022
```

```
data.table::year(c(Sys.time(), Sys.Date()))
```

```
## [1] 2022 2022
```

```
x <- Sys.time()
class(x)
```

```
## [1] "POSIXct" "POSIXt"
```

```
format(x, format = "%Y-%m-%d")
```

```
## [1] "2022-04-13"
```

```
x <- c("2019-12-21", "2019/12/21")
data.table::year("2019-12-21")
```

© 王建硕


```
## [1] 2019
```

```
data.table::year("2019/12/21")
```

```
## [1] 2019
```

但是，下面这样会报错

```
data.table::year(x)
```

```
## Error in as.POSIXlt.character(x): character string is not in a standard unambiguous format
```

正确的姿势是首先将表示日期的字符串格式统一

```
gsub(pattern = "/", replacement = "-", x) %>%
  data.table::year()
```

```
## [1] 2019 2019
```

date-times 表示 POSIXct 和 POSIXlt 类型的日期对象

```
(x <- Sys.time())
```

```
## [1] "2022-04-13 07:25:20 UTC"
```

```
class(x)
```

```
## [1] "POSIXct" "POSIXt"
```

```
data.table::second(x) # 取秒
```

```
## [1] 20
```

```
format(x, format = "%S")
```

```
## [1] "20"
```

```
data.table::minute(x) # 取分
```

```
## [1] 25
```

```
format(x, format = "%M")
```

```
## [1] "25"
```

```
data.table::hour(x) # 取时
```

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

```
format(x, format = "%H")
```

```
## [1] "07"
```

```
data.table::yday(x) # 此刻在一年的第几天
```

```
## [1] 103
```

```
data.table::wday(x) # 此刻在一周的第几天，星期日是第1天，星期六是第7天
```

```
## [1] 4
```

非洲 ©

```
data.table::mday(x) # 此刻在当月第几天
```

```
## [1] 13
```

```
format(x, format = "%d")
```

```
## [1] "13"
```

```
weekdays(x)
```

```
## [1] "Wednesday"
```

```
weekdays(x, abbreviate = T)
```

```
## [1] "Wed"
```

```
data.table::week(x) # 此刻在第几周
```

```
## [1] 15
```

```
data.table::isoweek(x)
```

```
## [1] 15
```

```
data.table::month(x) # 此刻在第几月
```

```
## [1] 4
```

```
format(x, format = "%m")
```

```
## [1] "04"
```

```
months(x)
```

```
## [1] "April"
```

```
months(x, abbreviate = T)
```

```
## [1] "Apr"
```

```
data.table::quarter(x) # 此刻在第几季度
```

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

```
quarters(x)
```

```
## [1] "Q2"
```

```
data.table::year(x) # 取年
```

```
## [1] 2022
```

```
format(x, format = "%Y")
```

```
## [1] "2022"
```

© 王沛勇

提示

format() 是一个泛型函数，此刻命名空间有 91 方法。format.Date()，format.difftime()，format.POSIXct() 和 format.POSIXlt() 四个函数通过格式化不同类型的日期数据对象抽取指定部分。

```
format(difftime(Sys.time(), x, units = "secs"))
```

```
## [1] "0.09684896 secs"
```

日期转化详见 [Ripley and Hornik, 2001, Grothendieck and Petzoldt, 2004]

上个季度最后一天

```
# https://d.cosx.org/d/421162/16
as.Date(cut(as.Date(c("2020-02-01", "2020-05-02")), "quarter")) - 1
```

```
## [1] "2019-12-31" "2020-03-31"
```

本季度第一天

```
as.Date(cut(as.Date(c("2020-02-01", "2020-05-02")), "quarter"))
```

```
## [1] "2020-01-01" "2020-04-01"
```

类似地，本月第一天和上月最后一天

```
# 本月第一天
as.Date(cut(as.Date(c("2020-02-01", "2020-05-02")), "month"))
```

```
## [1] "2020-02-01" "2020-05-01"
```

```
# 上月最后一天
as.Date(cut(as.Date(c("2020-02-01", "2020-05-02")), "month")) - 1
```

```
## [1] "2020-01-31" "2020-04-30"
```

timeDate 提供了很多日期计算函数，比如季初、季末、月初、月末等

```
library(timeDate)
# 季初
as.Date(format(timeFirstDayInQuarter(charvec = c("2020-02-01", "2020-05-02")), format = "%Y-%m-%d"))
# 季末
as.Date(format(timeLastDayInQuarter(charvec = c("2020-02-01", "2020-05-02")), format = "%Y-%m-%d"))
# 月初
as.Date(format(timeFirstDayInMonth(charvec = c("2020-02-01", "2020-05-02")), format = "%Y-%m-%d"))
# 月末
as.Date(format(timeLastDayInMonth(charvec = c("2020-02-01", "2020-05-02")), format = "%Y-%m-%d"))
```

cut.Date() 是一个泛型函数，查看它的所有 S3 方法

```
methods(cut)
```

```
## [1] cut.Date        cut.default     cut.dendrogram* cut.IDate*
## [5] cut.POSIXt
## see '?methods' for accessing help and source code
```

© 基础

格式化输出日期类型数据

```
formatC(round(runif(1, 1e8, 1e9)), digits = 10, big.mark = ",")
```

```
## [1] "837,074,657"
```

```
# Sys.setlocale(locale = "C") # 如果是 Windows 系统，必须先设置，否则转化结果是 NA
as.Date(paste("1990-January", 1, sep = "-"), format = "%Y-%B-%d")
```

```
## [1] "1990-01-01"
```

获取当日零点

```
format(as.POSIXlt(Sys.Date()), "%Y-%m-%d %H:%M:%S")
```

```
## [1] "2022-04-13 00:00:00"
```

从 POSIXt 数据对象中，抽取小时和分钟部分，返回字符串

```
strftime(x = Sys.time(), format = "%H:%M")
```

```
## [1] "07:25"
```

表 2.3: 日期表格

代码	含义	代码	含义
%a	Abbreviated weekday	%A	Full weekday
%b	Abbreviated month	%B	Full month
%c	Locale-specific date and time	%d	Decimal date
%H	Decimal hours (24 hour)	%I	Decimal hours (12 hour)
%j	Decimal day of the year	%m	Decimal month
%M	Decimal minute	%p	Locale-specific AM/PM
%S	Decimal second	%U	Decimal week of the year (starting on Sunday)
%w	Decimal Weekday (0=Sunday)	%W	Decimal week of the year (starting on Monday)
%x	Locale-specific Date	%X	Locale-specific Time
%y	2-digit year	%Y	4-digit year
%z	Offset from GMT	%Z	Time zone (character)

本节介绍了 R 本身提供的基础日期操作，第三十章着重介绍一般的时间序列类型的数据对象及其操作。

Jeffrey A. Ryan 开发的 xts 和 quantmod 包，Joshua M. Ulrich 开发的 zoo 是处理时间序列数据的主要工具

Jeffrey A. Ryan 在开设了一门免费课程教大家如何在 R 语言中使用 xts 和 zoo 包操作时间序列数据 [2]

xts (eXtensible Time Series) 扩展的 zoo 对象

```
xts(x = NULL,
    order.by = index(x),
    frequency = NULL,
    unique = TRUE,
```

[2] https://www.datacamp.com/courses/manipulating-time-series-data-in-r-with-xts-zoo

© 石慧


```
    tzone = Sys.getenv("TZ"),
    ...)
```

```
library(zoo)
library(xts)
x = matrix(1:4, ncol = 2,nrow = 2)
idx <- as.Date(c("2018-01-01", "2019-12-12"))
# xts = matrix + index
xts(x, order.by = idx)
```

```
##            [,1] [,2]
## 2018-01-01    1    3
## 2019-12-12    2    4
```

Date，POSIX times，timeDate，chron 等各种各样处理日期数据的对象

2.9 空值

移除 `list()` 列表里的为 `NULL` 元素

```
rm_null <- function(l) Filter(Negate(is.null), l)
```

©黄湘云

第三章　数据搬运

导入数据与导出数据，各种数据格式，数据库

处理 Excel 2003 (XLS) 和 Excel 2007 (XLSX) 文件还可以使用 WriteXLS 包，不过它依赖于 Perl，另一个 R 包 xlsx 与之功能类似，依赖 Java 环境。Jennifer Bryan 和 Hadley Wickham 开发的 readxl 包和 Jeroen Ooms 开发的 writexl 包专门处理 xlsx 格式并且无任何系统依赖。

3.1　导入数据

Base R 针对不同的数据格式文件，提供了大量的数据导入和导出函数，不愧是专注数据分析 20 余年的优秀统计软件。除了函数 `write.ftable` 和 `read.ftable` 来自 stats 包，都来自 base 和 utils 包

```
# 当前环境的搜索路径
searchpaths()
```

```
## [1] ".GlobalEnv"
## [2] "/opt/R/4.1.3/lib/R/library/stats"
## [3] "/opt/R/4.1.3/lib/R/library/graphics"
## [4] "/opt/R/4.1.3/lib/R/library/grDevices"
## [5] "/opt/R/4.1.3/lib/R/library/utils"
## [6] "/opt/R/4.1.3/lib/R/library/datasets"
## [7] "/opt/R/4.1.3/lib/R/library/methods"
## [8] "Autoloads"
## [9] "/opt/R/4.1.3/lib/R/library/base"
```

```
# 返回匹配结果及其所在路径的编号
apropos("^(read|write)", where = TRUE, mode = "function")
```

```
##                 5                 5                 9                 5
##         "read.csv"        "read.csv2"         "read.dcf"       "read.delim"
##                 5                 5                 5                 2
##      "read.delim2"         "read.DIF"     "read.fortran"      "read.ftable"
##                 5                 5                 5                 9
##         "read.fwf"      "read.socket"       "read.table"          "readBin"
##                 9                 5                 9                 9
##         "readChar" "readCitationFile"         "readline"        "readLines"
##                 9                 9                 9                 5
##          "readRDS"     "readRenviron"            "write"        "write.csv"
```

©李渊

```
##                  5                  9                  2                  5
##       "write.csv2"        "write.dcf"     "write.ftable"     "write.socket"
##                  5                  9                  9                  9
##      "write.table"         "writeBin"        "writeChar"       "writeLines"
```

3.1.1 scan

```
scan(file = "", what = double(), nmax = -1, n = -1, sep = "",
     quote = if(identical(sep, "\n")) "" else "'\"", dec = ".",
     skip = 0, nlines = 0, na.strings = "NA",
     flush = FALSE, fill = FALSE, strip.white = FALSE,
     quiet = FALSE, blank.lines.skip = TRUE, multi.line = TRUE,
     comment.char = "", allowEscapes = FALSE,
     fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)
```

首先让我们用 cat 函数创建一个练习数据集 ex.data

```
cat("TITLE extra line", "2 3 5 7", "11 13 17")
```

```
## TITLE extra line 2 3 5 7 11 13 17
```

```
cat("TITLE extra line", "2 3 5 7", "11 13 17", file = "data/ex.data", sep = "\n")
```

以此练习数据集，介绍 scan 函数最常用的参数

```
scan("data/ex.data")
```

```
## Error in scan("data/ex.data"): scan() expected 'a real', got 'TITLE'
```

从上面的报错信息，我们发现 scan 函数只能读取同一类型的数据，如布尔型 logical，整型 integer，数值型 numeric(double)，复数型 complex，字符型 character，raw 和列表 list。所以我们设置参数 skip = 1 把第一行跳过，就成功读取了数据

```
scan("data/ex.data", skip = 1)
```

```
## [1]  2  3  5  7 11 13 17
```

如果设置参数 quiet = TRUE 就不会报告读取的数据量

```
scan("data/ex.data", skip = 1, quiet = TRUE)
```

```
## [1]  2  3  5  7 11 13 17
```

参数 nlines = 1 表示只读取一行数据

```
scan("data/ex.data", skip = 1, nlines = 1) # only 1 line after the skipped one
```

```
## [1] 2 3 5 7
```

默认参数 flush = TRUE 表示读取最后一个请求的字段后，刷新到行尾，下面对比一下读取的结果

```
scan("data/ex.data", what = list("", "", "")) # flush is F -> read "7"
```

```
## Warning in scan("data/ex.data", what = list("", "", "")): number of items read
## is not a multiple of the number of columns
```

© 统计学习

```
## [[1]]
## [1] "TITLE" "2"     "7"     "17"
##
## [[2]]
## [1] "extra" "3"     "11"    ""
##
## [[3]]
## [1] "line" "5"    "13"   ""
```

```
scan("data/ex.data", what = list("", "", ""), flush = TRUE)
```

```
## [[1]]
## [1] "TITLE" "2"     "11"
##
## [[2]]
## [1] "extra" "3"     "13"
##
## [[3]]
## [1] "line" "5"    "17"
```

临时文件 ex.data 用完了，我们调用 `unlink` 函数将其删除，以免留下垃圾文件

```
unlink("data/ex.data") # tidy up
```

3.1.2 read.table

```
read.table(file,
  header = FALSE, sep = "", quote = "\"'",
  dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
  row.names, col.names, as.is = !stringsAsFactors,
  na.strings = "NA", colClasses = NA, nrows = -1,
  skip = 0, check.names = TRUE, fill = !blank.lines.skip,
  strip.white = FALSE, blank.lines.skip = TRUE,
  comment.char = "#",
  allowEscapes = FALSE, flush = FALSE,
  stringsAsFactors = default.stringsAsFactors(),
  fileEncoding = "", encoding = "unknown", text, skipNul = FALSE
)

read.csv(file,
  header = TRUE, sep = ",", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "", ...
)

read.csv2(file,
  header = TRUE, sep = ";", quote = "\"",
```

© 黄湘云


```
  dec = ",", fill = TRUE, comment.char = "", ...
)

read.delim(file,
  header = TRUE, sep = "\t", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "", ...
)

read.delim2(file,
  header = TRUE, sep = "\t", quote = "\"",
  dec = ",", fill = TRUE, comment.char = "", ...
)
```

变量名是不允许以下划线开头的，同样在数据框里，列名也不推荐使用下划线开头。默认情况下，`read.table` 都会通过参数 `check.names` 检查列名的有效性，该参数实际调用了函数 `make.names` 去检查。如果想尽量保持数据集原来的样子可以设置参数 `check.names = FALSE, stringsAsFactors = FALSE`。默认情形下，`read.table` 还会将字符串转化为因子变量，这是 R 的历史原因，作为一门统计学家的必备语言，在统计模型中，字符常用来描述类别，而类别变量在 R 环境中常用因子类型来表示，而且大量内置的统计模型也是将它们视为因子变量，如 `lm` 、`glm` 等

```
dat1 = read.table(header = TRUE, check.names = TRUE, text = "
_a _b _c
1 2 a1
3 4 a2
")
dat1
```

```
##   X_a X_b X_c
## 1   1   2  a1
## 2   3   4  a2
```

```
dat2 = read.table(header = TRUE, check.names = FALSE, text = "
_a _b _c
1 2 a1
3 4 a2
")
dat2
```

```
##   _a _b _c
## 1  1  2 a1
## 2  3  4 a2
```

```
dat3 <- read.table(header = TRUE, check.names = FALSE,
  stringsAsFactors = FALSE, text = "
_a _b _c
1 2 a1
3 4 a2
"
```

©黄湘云

```
)
dat3
```

```
##   _a _b _c
## 1  1  2 a1
## 2  3  4 a2
```

3.1.3 readLines

```
readLines(con = stdin(), n = -1L, ok = TRUE, warn = TRUE,
          encoding = "unknown", skipNul = FALSE)
```

让我们折腾一波，读进来又写出去，只有 R 3.5.3 以上才能保持原样的正确输入输出，因为这里有一个之前版本包含的 BUG

```
writeLines(readLines(system.file("DESCRIPTION", package = "splines")), "data/DESCRIPTION")
# 比较一下
identical(
  readLines(system.file("DESCRIPTION", package = "splines")),
  readLines("data/DESCRIPTION")
)
```

```
## [1] TRUE
```

这次我们创建一个真的临时文件，因为重新启动 R 这个文件和文件夹就没有了，回收掉了

```
fil <- tempfile(fileext = ".data")
cat("TITLE extra line", "2 3 5 7", "", "11 13 17", file = fil,
    sep = "\n")
fil
```

```
## [1] "/tmp/Rtmpcbvh7t/file25a9f5247d3bf.data"
```

设置参数 `n = -1` 表示将文件 fil 的内容从头读到尾

```
readLines(fil, n = -1)
```

```
## [1] "TITLE extra line" "2 3 5 7"          ""                 "11 13 17"
```

作为拥有良好习惯的 R 用户，这种垃圾文件最好用后即焚

```
unlink(fil) # tidy up
```

再举个例子，我们创建一个新的临时文件 `fil`，文件内容只有

```
cat("123\nabc")
```

```
## 123
## abc
```

```
fil <- tempfile("test")
cat("123\nabc\n", file = fil, append = TRUE)
fil
```

© 王建平


```
## [1] "/tmp/Rtmpcbvh7t/test25a9f5bbd2a3a"
```

```
readLines(fil)
```

```
## [1] "123" "abc"
```

这次读取文件的过程给出了警告，原因是 fil 没有以空行结尾，`warn = TRUE` 表示这种情况要给出警告，如果设置参数 `warn = FALSE` 就没有警告。我们还是建议大家尽量遵循规范。

再举一个例子，从一个连接读取数据，建立连接的方式有很多，参见 `?file`，下面设置参数 `blocking`

```
con <- file(fil, "r", blocking = FALSE)
readLines(con)
```

```
## [1] "123" "abc"
```

```
cat(" def\n", file = fil, append = TRUE)
readLines(con)
```

```
## [1] " def"
```

```
# 关闭连接
close(con)
# 清理垃圾文件
unlink(fil)
```

3.1.4 readRDS

序列化数据操作，Mark Klik 开发的 fst 和 Travers Ching 开发的 qs，Hadley Wickham 开发的 feather 包实现跨语言环境快速的读写数据

表 3.1: fst 序列化数据框对象性能比较 BaseR、data.table 和 feather [1]

Method	Format	Time (ms)	Size (MB)	Speed (MB/s)	N
readRDS	bin	1577	1000	633	112
saveRDS	bin	2042	1000	489	112
fread	csv	2925	1038	410	232
fwrite	csv	2790	1038	358	241
read_feather	bin	3950	813	253	112
write_feather	bin	1820	813	549	112
read_fst	**bin**	**457**	**303**	**2184**	**282**
write_fst	**bin**	**314**	**303**	**3180**	**291**

目前比较好的是 qs 和 fst 包

[1] https://www.fstpackage.org/

©黄海广

3.2 其它数据格式

来自其它格式的数据形式，如 JSON、XML、YAML 需要转化清理成 R 中数据框的形式 data.frame

1. Data Rectangling with jq
2. Mongolite User Manual introduction to using MongoDB with the mongolite client in R

jsonlite 读取 `*.json` 格式的文件，`jsonlite::write_json` 函数将 R 对象保存为 JSON 文件，`jsonlite::fromJSON` 将 json 字符串或文件转化为 R 对象，`jsonlite::toJSON` 函数正好与之相反

```
library(jsonlite)
# 从 json 格式的文件导入
# jsonlite::read_json(path = "path/to/filename.json")
# A JSON array of primitives
json <- '["Mario", "Peach", null, "Bowser"]'

# 简化为原子向量atomic vector
fromJSON(json)
```

```
## [1] "Mario"  "Peach"  NA       "Bowser"
```

```
# 默认返回一个列表
fromJSON(json, simplifyVector = FALSE)
```

```
## [[1]]
## [1] "Mario"
##
## [[2]]
## [1] "Peach"
##
## [[3]]
## NULL
##
## [[4]]
## [1] "Bowser"
```

yaml 包读取 `*.yml` 格式文件，返回一个列表，`yaml::write_yaml` 函数将 R 对象写入 yaml 格式

```
library(yaml)
yaml::read_yaml(file = '_bookdown.yml')
```

```
## $book_filename
## [1] "masr"
##
## $delete_merged_file
## [1] TRUE
##
## $language
## $language$label
## $language$label$fig
```

©清华大学出版社

```
## [1] "图 "
##
## $language$label$tab
## [1] "表 "
##
##
## $language$ui
## $language$ui$edit
## [1] "编辑"
##
## $language$ui$chapter_name
## [1] "第 " " 章"
##
## $language$ui$appendix_name
## [1] "附录 "
##
##
##
## $new_session
## [1] TRUE
##
## $before_chapter_script
## [1] "_common.R"
##
## $rmd_files
##  [1] "index.Rmd"                      "preface.Rmd"
##  [3] "data-wrangling.Rmd"             "data-structure.Rmd"
##  [5] "data-transportation.Rmd"        "string-operations.Rmd"
##  [7] "regular-expressions.Rmd"        "data-manipulation.Rmd"
##  [9] "advanced-manipulation.Rmd"      "parallel-manipulation.Rmd"
## [11] "other-manipulation.Rmd"         "statistical-graphics.Rmd"
## [13] "graphics-foundations.Rmd"       "data-visualization.Rmd"
## [15] "interactive-web-graphics.Rmd"   "dynamic-documents.Rmd"
## [17] "document-elements.Rmd"          "portable-documents.Rmd"
## [19] "web-documents.Rmd"              "office-documents.Rmd"
## [21] "reproducible-workflows.Rmd"     "advanced-documents.Rmd"
## [23] "data-product.Rmd"               "interactive-data-tables.Rmd"
## [25] "interactive-shiny-app.Rmd"      "statistical-foundations.Rmd"
## [27] "sampling-distributions.Rmd"     "parameter-estimators.Rmd"
## [29] "hypothesis-test.Rmd"            "power-analysis.Rmd"
## [31] "experimental-design.Rmd"        "statistical-models.Rmd"
## [33] "linear-models.Rmd"              "generalized-linear-models.Rmd"
## [35] "data-modeling.Rmd"              "text-analysis.Rmd"
## [37] "survival-analysis.Rmd"          "time-series-analysis.Rmd"
## [39] "spatio-temporal-data.Rmd"       "spatial-analysis.Rmd"
```

© 统计学习基础

```
## [41] "spatial-viz.Rmd"                "case-study.Rmd"
## [43] "data-explorer.Rmd"              "machine-learning.Rmd"
## [45] "gradient-boosting-machine.Rmd"  "numerical-optimization.Rmd"
## [47] "appendix.Rmd"                   "matrix-operations.Rmd"
## [49] "symbolic-computation.Rmd"       "mixed-programming.Rmd"
## [51] "object-oriented-programming.Rmd" "file-operations.Rmd"
## [53] "other-softwares.Rmd"            "notations.Rmd"
## [55] "references.Rmd"
```

表 3.2: 导入来自其它数据分析软件产生的数据集

统计软件	R 函数	R 包
ERSI ArcGIS	`read.shapefile`	shapefiles
Matlab	`readMat`	R.matlab
minitab	`read.mtp`	foreign
SAS (permanent data)	`read.ssd`	foreign
SAS (XPORT format)	`read.xport`	foreign
SPSS	`read.spss`	foreign
Stata	`read.dta`	foreign
Systat	`read.systat`	foreign
Octave	`read.octave`	foreign

表 3.3: 导入来自其它格式的数据集

文件格式	R 函数	R 包
列联表数据	`read.ftable`	stats
二进制数据	`readBin`	base
字符串数据	`readChar`	base
剪贴板数据	`readClipboard`	utils

`read.dcf` 函数读取 Debian 控制格式文件，这种类型的文件以人眼可读的形式在存储数据，如 R 包的 DESCRIPTION 文件或者包含所有 CRAN 上 R 包描述的文件 https://cran.r-project.org/src/contrib/PACKAGES

```
x <- read.dcf(file = system.file("DESCRIPTION", package = "splines"),
              fields = c("Package", "Version", "Title"))
x
```

```
##      Package   Version Title
## [1,] "splines" "4.1.3" "Regression Spline Functions and Classes"
```

最后要提及拥有瑞士军刀之称的 rio 包，它集合了当前 R 可以读取的所有统计分析软件导出的数据。

© 黄湘云

表 3.4: 数据库接口

数据库	官网	R 接口	开发仓
MySQL	https://www.mysql.com/	RMySQL	https://github.com/r-dbi/RMySQL
SQLite	https://www.sqlite.org	RSQLite	https://github.com/r-dbi/RSQLite
PostgreSQL	https://www.postgresql.org/	RPostgres	https://github.com/r-dbi/RPostgres
MariaDB	https://mariadb.org/	RMariaDB	https://github.com/r-dbi/RMariaDB

3.3 导入大数据集

在不使用数据库的情况下，从命令行导入大数据集，如几百 M 或几个 G 的 csv 文件。利用 data.table 包的 `fread` 去读取

https://stackoverflow.com/questions/1727772/

3.4 从数据库导入

Hands-On Programming with R 数据读写章节[2] 以及 R, Databases and Docker

将大量的 txt 文本存进 MySQL 数据库中，通过操作数据库来聚合文本，极大降低内存消耗[3]，而 ODBC 与 DBI 包是其它数据库接口的基础，knitr 提供了一个支持 SQL 代码的引擎，它便是基于 DBI，因此可以在 R Markdown 文档中直接使用 SQL 代码块[4]。这里制作一个归纳表格，左边数据库右边对应其 R 接口，两边都包含链接，如表 3.4 所示

3.4.1 PostgreSQL

odbc 可以支持很多数据库，下面以连接 PostgreSQL 数据库为例介绍其过程

首先在某台机器上，拉取 PostgreSQL 的 Docker 镜像

```
docker pull postgres
```

在 Docker 上运行 PostgreSQL，主机端口号 8181 映射给数据库 PostgreSQL 的默认端口号 5432（或其它你的 DBA 分配给你的端口）

```
docker run --name psql -d -p 8181:5432 -e ROOT=TRUE \
   -e USER=xiangyun -e PASSWORD=cloud postgres
```

在主机 Ubuntu 上配置

```
sudo apt-get install unixodbc unixodbc-dev odbc-postgresql
```

端口 5432 是分配给 PostgreSQL 的默认端口，`host` 可以是云端的地址，如你的亚马逊账户下的 PostgreSQL 数据库地址 `<ec2-54-83-201-96.compute-1.amazonaws.com>`，也可以是本地局域网 IP 地址，如

[2] https://rstudio-education.github.io/hopr/dataio.html

[3] https://brucezhaor.github.io/blog/2016/08/04/batch-process-txt-to-mysql

[4] https://bookdown.org/yihui/rmarkdown/language-engines.html#sql [rstudio-spark]: https://spark.rstudio.com/ [rmarkdown-teaching-demo]: https://stackoverflow.com/questions/35459166

©黄海广

<192.168.1.200>。通过参数 dbname 连接到指定的 PostgreSQL 数据库，如 Heroku，这里作为演示就以默认的数据库 postgres 为例

查看配置系统文件路径

```
odbcinst -j
```

```
unixODBC 2.3.6
DRIVERS............: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
USER DATA SOURCES..: /root/.odbc.ini
SQLULEN Size.......: 8
SQLLEN Size........: 8
SQLSETPOSIROW Size.: 8
```

不推荐修改全局配置文件，可设置 ODBCSYSINI 环境变量指定配置文件路径，如 ODBCSYSINI=~/ODBC http://www.unixodbc.org/odbcinst.html

安装完驱动程序，/etc/odbcinst.ini 文件内容自动更新，我们可以不必修改，如果你想自定义不妨手动修改，我们查看在 R 环境中注册的数据库，可以看到 PostgreSQL 的驱动已经配置好

```
odbc::odbcListDrivers()
```

```
                 name   attribute                                    value
1     PostgreSQL ANSI Description    PostgreSQL ODBC driver (ANSI version)
2     PostgreSQL ANSI      Driver                              psqlodbca.so
3     PostgreSQL ANSI       Setup                           libodbcpsqlS.so
4     PostgreSQL ANSI       Debug                                        0
5     PostgreSQL ANSI     CommLog                                        1
6     PostgreSQL ANSI  UsageCount                                        1
7  PostgreSQL Unicode Description PostgreSQL ODBC driver (Unicode version)
8  PostgreSQL Unicode      Driver                              psqlodbcw.so
9  PostgreSQL Unicode       Setup                           libodbcpsqlS.so
10 PostgreSQL Unicode       Debug                                        0
11 PostgreSQL Unicode     CommLog                                        1
12 PostgreSQL Unicode  UsageCount                                        1
```

系统配置文件 /etc/odbcinst.ini 已经包含有 PostgreSQL 的驱动配置，无需再重复配置

```
[PostgreSQL ANSI]
Description=PostgreSQL ODBC driver (ANSI version)
Driver=psqlodbca.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1

[PostgreSQL Unicode]
Description=PostgreSQL ODBC driver (Unicode version)
```

© 黄湘云


```
Driver=psqlodbcw.so
Setup=libodbcpsqlS.so
Debug=0
CommLog=1
UsageCount=1
```

只需将如下内容存放在 ~/.odbc.ini 文件中，

```
[PostgreSQL]
Driver              = PostgreSQL Unicode
Database            = postgres
Servername          = 172.17.0.1
UserName            = postgres
Password            = default
Port                = 8080
```

最后，一行命令 DNS 配置连接 https://github.com/r-dbi/odbc 这样就实现了代码中无任何敏感信息，这里为了展示这个配置过程故而把相关信息公开。

> 注意下面的内容需要在容器中运行，Windows 环境下的配置 PostgreSQL 的驱动有点麻烦就不搞了，意义也不大，现在数据库基本都是跑在 Linux 系统上

`docker-machine.exe ip default` 可以获得本地 Docker 的 IP，比如 192.168.99.101。Travis 上 `ip addr` 可以查看 Docker 的 IP，如 172.17.0.1

```r
library(DBI)
con <- dbConnect(RPostgres::Postgres(),
  dbname = "postgres",
  host = ifelse(is_on_travis, Sys.getenv("DOCKER_HOST_IP"), "192.168.99.101"),
  port = 8080,
  user = "postgres",
  password = "default"
)
```

```r
library(DBI)
con <- dbConnect(odbc::odbc(), "PostgreSQL")
```

列出数据库中的所有表

```r
dbListTables(con)
```

第一次启动从 Docker Hub 上下载的镜像，默认的数据库是 postgres 里面没有任何表，所以将 R 环境中的 mtcars 数据集写入 postgres 数据库

将数据集 mtcars 写入 PostgreSQL 数据库中，基本操作，写入表的操作也不能缓存，即不能缓存数据库中的表 mtcars

```r
dbWriteTable(con, "mtcars", mtcars, overwrite = TRUE)
```

现在可以看到数据表 mtcars 的各个字段

```r
dbListFields(con, "mtcars")
```

最后执行一条 SQL 语句

©谢益辉

```
res <- dbSendQuery(con, "SELECT * FROM mtcars WHERE cyl = 4") # 发送 SQL 语句
dbFetch(res) # 获取查询结果
dbClearResult(res) # 清理查询通道
```

或者一条命令搞定

```
dbGetQuery(con, "SELECT * FROM mtcars WHERE cyl = 4")
```

再复杂一点的 SQL 查询操作

```
dbGetQuery(con, "SELECT cyl, AVG(mpg) AS mpg FROM mtcars GROUP BY cyl ORDER BY cyl")
aggregate(mpg ~ cyl, data = mtcars, mean)
```

得益于 knitr [Xie, 2015] 开发的钩子，这里直接写 SQL 语句块，值得注意的是 SQL 代码块不能启用缓存，数据库连接通道也不能缓存，如果数据库中还没有写入表，那么写入表的操作也不能缓存，`tab.cap = "表格标题"` 输出的内容是一个表格

```
SELECT cyl, AVG(mpg) AS mpg FROM mtcars GROUP BY cyl ORDER BY cyl
```

如果将查询结果导出到变量，在 Chunk 设置 `output.var = "agg_cyl"` 可以使用缓存，下面将 mpg 按 cyl 分组聚合的结果打印出来，`ref.label = "mtcars"` 引用上一个 SQL 代码块的内容

这种基于 odbc 的方式的好处就不需要再安装 R 包 RPostgres 和相关系统依赖，最后关闭连接通道

```
dbDisconnect(con)
```

3.4.2 MySQL

MySQL 是一个很常见，应用也很广泛的数据库，数据分析的常见环境是在一个 R Notebook 里，我们可以在正文之前先设定数据库连接信息

```
```{r setup}
library(DBI)
# 指定数据库连接信息
db <- dbConnect(RMySQL::MySQL(),
  dbname = 'dbtest',
  username = 'user_test',
  password = 'password',
  host = '10.10.101.10',
  port = 3306
)
# 创建默认连接
knitr::opts_chunk$set(connection = 'db')
# 设置字符编码，以免中文查询乱码
DBI::dbSendQuery(db, 'SET NAMES utf8')
# 设置日期变量，以运用在SQL中
idate <- '2019-05-03'
```
```

SQL 代码块中使用 R 环境中的变量，并将查询结果输出为 R 环境中的数据框

© 黄湘云


```{sql, output.var='data_output'}
SELECT * FROM user_table where date_format(created_date,'%Y-%m-%d')>=?idate
```

以上代码会将 SQL 的运行结果存在 `data_output` 这是数据库中，idate 取之前设置的日期 `2019-05-03`，`user_table` 是 MySQL 数据库中的表名，`created_date` 是创建 `user_table` 时，指定的日期名。

如果 SQL 比较长，为了代码美观，把带有变量的 SQL 保存为 `demo.sql` 脚本，只需要在 SQL 的 chunk 中直接读取 SQL 文件[5]。

```{sql, code=readLines('demo.sql'), output.var='data_output'}
```

如果我们需要每天或者按照指定的日期重复地运行这个 R Markdown 文件，可以在 YAML 部分引入参数[6]

```
---
params:
  date: "2019-05-03"  # 参数化日期
---
```

```{r setup, include=FALSE}
idate = params$date # 将参数化日期传递给 idate 变量
```

我们将这个 Rmd 文件命名为 `MyDocument.Rmd`，运行这个文件可以从 R 控制台执行或在 RStudio 点击 knit。

```
rmarkdown::render("MyDocument.Rmd", params = list(
  date = "2019-05-03"
))
```

如果在文档的 YAML 位置已经指定日期，这里可以不指定。注意在这里设置日期会覆盖 YAML 处指定的参数值，这样做的好处是可以批量化操作。

3.4.3 Spark

当数据分析报告遇上 Spark 时，就需要 SparkR、sparklyr、arrow 或 rsparking 接口了，Javier Luraschi 写了一本书 The R in Spark: Learning Apache Spark with R 详细介绍了相关扩展和应用

首先安装 sparklyr 包，RStudio 公司 Javier Lurasch 开发了 sparklyr 包，作为 Spark 与 R 语言之间的接口，安装完 sparklyr 包，还是需要 Spark 和 Hadoop 环境

```
install.packages('sparklyr')
library(sparklyr)
spark_install()
# Installing Spark 2.4.0 for Hadoop 2.7 or later.
# Downloading from:
# - 'https://archive.apache.org/dist/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz'
# Installing to:
# - '~/spark/spark-2.4.0-bin-hadoop2.7'
```

[5] https://d.cosx.org/d/419974

[6] https://bookdown.org/yihui/rmarkdown/params-knit.html

© 黄湘云

```
# trying URL 'https://archive.apache.org/dist/spark/spark-2.4.0/spark-2.4.0-bin-hadoop2.7.tgz'
# Content type 'application/x-gzip' length 227893062 bytes (217.3 MB)
# ==================================================
# downloaded 217.3 MB
#
# Installation complete.
```

既然 sparklyr 已经安装了 Spark 和 Hadoop 环境，安装 SparkR 后，只需配置好路径，就可以加载 SparkR 包

```
install.packages('SparkR')
if (nchar(Sys.getenv("SPARK_HOME")) < 1) {
  Sys.setenv(SPARK_HOME = "~/spark/spark-2.4.0-bin-hadoop2.7")
}
library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))
sparkR.session(master = "local[*]", sparkConfig = list(spark.driver.memory = "2g"))
```

rscala 架起了 R 和 Scala 两门语言之间交流的桥梁，使得彼此之间可以互相调用

> 是否存在这样的可能，Spark 提供了大量的 MLib 库的调用接口，R 的功能支持是最少的，Java/Scala 是原生的，那么要么自己开发新的功能整合到 SparkR 中，要么借助 rscala 将 scala 接口代码封装进来

在本地，Windows 主机上，可以在 `.Rprofile` 中给 Spark 添加环境变量 `SPARK_HOME` 指定其安装路径，

```
# Windows 平台默认安装路径
Sys.setenv(SPARK_HOME = "C:/Users/XiangYun/AppData/Local/spark/spark-2.4.3-bin-hadoop2.7")
library(sparklyr)
sc <- spark_connect(master = "local", version = "2.4")
```

将 R 环境中的数据集 mtcars 传递到 Spark 上

```
cars <- copy_to(sc, mtcars)
cars
```

```
# Source: spark<mtcars> [?? x 11]
    mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  21       6   160   110  3.9   2.62  16.5     0     1     4     4
2  21       6   160   110  3.9   2.88  17.0     0     1     4     4
3  22.8     4   108    93  3.85  2.32  18.6     1     1     4     1
4  21.4     6   258   110  3.08  3.22  19.4     1     0     3     1
5  18.7     8   360   175  3.15  3.44  17.0     0     0     3     2
6  18.1     6   225   105  2.76  3.46  20.2     1     0     3     1
# ... with more rows
```

监控和分析命令执行的情况，可以在浏览器中，见图 3.1

```
spark_web(sc)
```

传递 SQL 查询语句，比如数据集 mtcars 有多少行

© 张翼

图 3.1: Spark Web 接口

```
library(DBI)
dbGetQuery(sc, "SELECT count(*) FROM mtcars")
```

```
  count(1)
1       32
```

进一步地，我们可以调用 dplyr 包来写数据操作，避免写复杂逻辑的 SQL 语句，

```
# library(dplyr) # 数据操作
library(tidyverse) # 提供更多功能，包括数据可视化
count(cars)
```

再举一个稍复杂的操作，先从数据集 cars 中选择两个字段 hp 和 mpg

```
select(cars, hp, mpg) %>%
  sample_n(100) %>% # 随机选择 100 行
  collect() %>% # 执行 SQL 查询，将结果返回到本地
  ggplot(aes(hp, mpg)) + # 绘图
  geom_point()
```

数据查询和结果可视化，见图 3.2

用完要记得关闭连接

```
spark_disconnect(sc)
```

不要使用 SparkR 接口，要使用 sparklyr，后者的功能已经全面覆盖前者，生态方面更是更是已经远远超越，它有大厂 RStudio 支持，是公司支持的旗舰项目。但是 sparklyr 的版本稍微比最新的 Spark 低一两个版本，这是开发周期和出于稳定性的考虑，无伤大雅！

©黄湘云

图 3.2: 数据聚合和可视化

Spark 提供了官方 R 语言接口 SparkR。Spark JVM 和 SparkR 包版本要匹配，比如从 CRAN 上安装了最新版的 SparkR，比如版本 2.4.4 就要去 Spark 官网下载最新版的预编译文件 spark-2.4.4-bin-hadoop2.7，解压到本地磁盘，比如 `D:/spark-2.4.4-bin-hadoop2.7`

```
Sys.setenv(SPARK_HOME = "D:/spark-2.4.4-bin-hadoop2.7")
# Sys.setenv(R_HOME = "C:/Program Files/R/R-3.6.1/")
library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))
sparkR.session(master = "local[*]",
               sparkConfig = list(spark.driver.memory = "4g"),
               enableHiveSupport = TRUE)
```

从数据集 mtcars（数据类型是 R 的 data.frame）创建 Spark 的 DataFrame 类型数据

```
cars <- as.DataFrame(mtcars)
# 显示 SparkDataFrame 的前几行
head(cars)
```

```
 mpg cyl disp  hp drat    wt  qsec vs am gear carb
1 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
2 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
3 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
4 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
5 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
6 18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

打印数据集 cars 的 schema 各个字段的

```
printSchema(cars)
```

```
root
 |-- mpg: double (nullable = true)
 |-- cyl: double (nullable = true)
 |-- disp: double (nullable = true)
 |-- hp: double (nullable = true)
 |-- drat: double (nullable = true)
 |-- wt: double (nullable = true)
 |-- qsec: double (nullable = true)
 |-- vs: double (nullable = true)
 |-- am: double (nullable = true)
```

© 基础设施


```
 |-- gear: double (nullable = true)
 |-- carb: double (nullable = true)
```

从本地 JSON 文件创建 DataFrame

```
path <- file.path(Sys.getenv("SPARK_HOME"), "examples/src/main/resources/people.json")
peopleDF <- read.json(path)
printSchema(peopleDF)
```

```
root
 |-- age: long (nullable = true)
 |-- name: string (nullable = true)
```

```
peopleDF
```

```
SparkDataFrame[age:bigint, name:string]
```

```
showDF(peopleDF)
```

```
+----+-------+
| age|   name|
+----+-------+
|null|Michael|
|  30|   Andy|
|  19| Justin|
+----+-------+
```

peopleDF 转成 Hive 中的表 people

```
createOrReplaceTempView(peopleDF, "people")
```

调用 `sql` 传递 SQL 语句查询数据，启动 `sparkR.session` 时，设置 `enableHiveSupport = TRUE`，就是执行不出来，报错，不知道哪里配置存在问题

```
teenagers <- SparkR::sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
show(people)
```

```
Error in handleErrors(returnStatus, conn) :
 org.apache.spark.sql.AnalysisException: java.lang.RuntimeException: java.io.IOException: (null) entry
  at org.apache.spark.sql.hive.HiveExternalCatalog.withClient(HiveExternalCatalog.scala:106)
  at org.apache.spark.sql.hive.HiveExternalCatalog.databaseExists(HiveExternalCatalog.scala:214)
  at org.apache.spark.sql.internal.SharedState.externalCatalog$lzycompute(SharedState.scala:114)
   at org.apache.spark.sql.internal.SharedState.externalCatalog(SharedState.scala:102)
  at org.apache.spark.sql.internal.SharedState.globalTempViewManager$lzycompute(SharedState.scala:14
  at org.apache.spark.sql.internal.SharedState.globalTempViewManager(SharedState.scala:136)
  at org.apache.spark.sql.hive.HiveSessionStateBuilder$$anonfun$2.apply(HiveSessionStateBuilder.scal
  at org.apache.spark.sql.hive.HiveSessionStateBuilder$$anonfun$2.apply(HiveSessionStateBuilder.scal
   at org.apache.spark.sql.catalyst.catalog.SessionCatalog.gl
```

调用 `collect` 函数执行查询，并将结果返回到本地 `data.frame` 类型

```
teenagersLocalDF <- collect(teenagers)
```

© 清华大学出版社

查看数据集 teenagersLocalDF 的属性

```
print(teenagersLocalDF)
```

最后，关闭 SparkSession 会话

```
sparkR.session.stop()
```

3.5 批量导入数据

```
library(tidyverse)
```

```
read_list <- function(list_of_datasets, read_func) {
  read_and_assign <- function(dataset, read_func) {
    dataset_name <- as.name(dataset)
    dataset_name <- read_func(dataset)
  }

  # invisible is used to suppress the unneeded output
  output <- invisible(
    sapply(list_of_datasets,
      read_and_assign,
      read_func = read_func, simplify = FALSE, USE.NAMES = TRUE
    )
  )

  # Remove the extension at the end of the data set names
  names_of_datasets <- c(unlist(strsplit(list_of_datasets, "[.]"))[c(T, F)])
  names(output) <- names_of_datasets
  return(output)
}
```

批量导入文件扩展名为 .csv 的数据文件，即逗号分割的文件

```
data_files <- list.files(path = "path/to/csv/dir",
                         pattern = ".csv", full.names = TRUE)
print(data_files)
```

相比于 Base R 提供的 read.csv 函数，使用 readr 包的 read_csv 函数可以更快地读取 csv 格式文件，特别是在读取 GB 级数据文件时，效果特别明显。

```
list_of_data_sets <- read_list(data_files, readr::read_csv)
```

使用 tibble 包的 glimpse 函数可以十分方便地对整个数据集有一个大致的了解，展示方式和信息量相当于 str 加 head 函数

```
tibble::glimpse(list_of_data_sets)
```

©黄湘云

3.6 批量导出数据

假定我们有一个列表，其每个元素都是一个数据框，现在要把每个数据框分别存入 xlsx 表的工作薄中，以 mtcars 数据集为例，将其按分类变量 cyl 分组拆分，获得一个列表 list

```
dat <- split(mtcars, mtcars$cyl)
dat
```

```
## $`4`
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230       22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona  21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
##
## $`6`
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Valiant        18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Merc 280       19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C      17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
##
## $`8`
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
```

```
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
```

将 xlsx 表格初始化，创建空白的工作薄，openxlsx 包不依赖 Java 环境，读写效率也高

```
## 加载 openxlsx 包
library(openxlsx)
## 创建空白的工作薄
wb <- createWorkbook()
```

将列表里的每张表分别存入 xlsx 表格的每个 worksheet，worksheet 的名字就是分组变量的名字

```
Map(function(data, name){
    addWorksheet(wb, name)
    writeData(wb, name, data)

}, dat, names(dat))
```

最后保存数据到磁盘，见图 3.3

```
saveWorkbook(wb, file = "data/matcars.xlsx", overwrite = TRUE)
```

图 3.3: 批量导出数据

3.7 导出数据

3.7.1 导出运行结果

```
capture.output(..., file = NULL, append = FALSE,
               type = c("output", "message"), split = FALSE)
```

capture.output 将一段 R 代码执行结果，保存到文件，参数为表达式。capture.output 和 sink 的关系相当于 with 和 attach 的关系。

© 谢湘


```
glmout <- capture.output(summary(glm(case ~ spontaneous + induced,
  data = infert, family = binomial()
)), file = "data/capture.txt")
capture.output(1 + 1, 2 + 2)
```

```
## [1] "[1] 2" "[1] 4"
```

```
capture.output({
  1 + 1
  2 + 2
})
```

```
## [1] "[1] 4"
```

sink 函数将控制台输出结果保存到文件，只将 outer 函数运行的结果保存到 ex-sink.txt 文件，outer 函数计算的是直积，在这里相当于 seq(10) %*% t(seq(10))，而在 R 语言中，更加有效的计算方式是 tcrossprod(seq(10),seq(10))

```
sink("data/ex-sink.txt")
i <- 1:10
outer(i, i, "*")
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
##  [1,]    1    2    3    4    5    6    7    8    9    10
##  [2,]    2    4    6    8   10   12   14   16   18    20
##  [3,]    3    6    9   12   15   18   21   24   27    30
##  [4,]    4    8   12   16   20   24   28   32   36    40
##  [5,]    5   10   15   20   25   30   35   40   45    50
##  [6,]    6   12   18   24   30   36   42   48   54    60
##  [7,]    7   14   21   28   35   42   49   56   63    70
##  [8,]    8   16   24   32   40   48   56   64   72    80
##  [9,]    9   18   27   36   45   54   63   72   81    90
## [10,]   10   20   30   40   50   60   70   80   90   100
```

```
sink()
```

3.7.2 导出数据对象

```
load(file, envir = parent.frame(), verbose = FALSE)

save(..., list = character(),
     file = stop("'file' must be specified"),
     ascii = FALSE, version = NULL, envir = parent.frame(),
     compress = isTRUE(!ascii), compression_level,
     eval.promises = TRUE, precheck = TRUE)

save.image(file = ".RData", version = NULL, ascii = FALSE,
           compress = !ascii, safe = TRUE)
```

©黄湘云

load 和 save 函数加载或保存包含工作环境信息的数据对象，save.image 保存当前工作环境到磁盘，即保存工作空间中所有数据对象，数据格式为 .RData，即相当于

```
save(list = ls(all.names = TRUE), file = ".RData", envir = .GlobalEnv)
```

dump 保存数据对象 AirPassengers 到文件 AirPassengers.txt，文件内容是 R 命令，可把 AirPassengers.txt 看作代码文档执行，dput 保存数据对象内容到文件 AirPassengers.dat，文件中不包含变量名 AirPassengers。注意到 dump 输入是一个字符串，而 dput 要求输入数据对象的名称，source 函数与 dump 对应，而 dget 与 dput 对应。

```
# 加载数据
data(AirPassengers, package = "datasets")
# 将数据以R代码块的形式保存到文件
dump('AirPassengers', file = 'data/AirPassengers.txt')
# source(file = 'data/AirPassengers.txt')
```

接下来，我们读取 AirPassengers.txt 的文件内容，可见它是一段完整的 R 代码，可以直接复制到 R 的控制台中运行，并且得到一个与原始 AirPassengers 变量一样的结果

```
cat(readLines('data/AirPassengers.txt'), sep = "\n")
```

```
## AirPassengers <-
## structure(c(112, 118, 132, 129, 121, 135, 148, 148, 136, 119,
## 104, 118, 115, 126, 141, 135, 125, 149, 170, 170, 158, 133, 114,
## 140, 145, 150, 178, 163, 172, 178, 199, 199, 184, 162, 146, 166,
## 171, 180, 193, 181, 183, 218, 230, 242, 209, 191, 172, 194, 196,
## 196, 236, 235, 229, 243, 264, 272, 237, 211, 180, 201, 204, 188,
## 235, 227, 234, 264, 302, 293, 259, 229, 203, 229, 242, 233, 267,
## 269, 270, 315, 364, 347, 312, 274, 237, 278, 284, 277, 317, 313,
## 318, 374, 413, 405, 355, 306, 271, 306, 315, 301, 356, 348, 355,
## 422, 465, 467, 404, 347, 305, 336, 340, 318, 362, 348, 363, 435,
## 491, 505, 404, 359, 310, 337, 360, 342, 406, 396, 420, 472, 548,
## 559, 463, 407, 362, 405, 417, 391, 419, 461, 472, 535, 622, 606,
## 508, 461, 390, 432), .Tsp = c(1949, 1960.916666666699, 12), class = "ts")
```

dput 函数类似 dump 函数，保存数据对象到磁盘文件

```
# 将 R 对象保存/导出到磁盘
dput(AirPassengers, file = 'data/AirPassengers.dat')
AirPassengers
```

```
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
```

© 黄湘云


```
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
```

```r
# dget 作用与 dput 相反
AirPassengers2 <- dget(file = 'data/AirPassengers.dat')
AirPassengers2
```

```
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1949 112 118 132 129 121 135 148 148 136 119 104 118
1950 115 126 141 135 125 149 170 170 158 133 114 140
1951 145 150 178 163 172 178 199 199 184 162 146 166
1952 171 180 193 181 183 218 230 242 209 191 172 194
1953 196 196 236 235 229 243 264 272 237 211 180 201
1954 204 188 235 227 234 264 302 293 259 229 203 229
1955 242 233 267 269 270 315 364 347 312 274 237 278
1956 284 277 317 313 318 374 413 405 355 306 271 306
1957 315 301 356 348 355 422 465 467 404 347 305 336
1958 340 318 362 348 363 435 491 505 404 359 310 337
1959 360 342 406 396 420 472 548 559 463 407 362 405
1960 417 391 419 461 472 535 622 606 508 461 390 432
```

同样地，现在我们观察 `dput` 函数保存的文件 `AirPassengers.dat` 内容，和 `dump` 函数保存的文件 `AirPassengers.txt` 相比，就缺一个赋值变量

```r
cat(readLines('data/AirPassengers.dat'), sep = "\n")
```

```
structure(c(112, 118, 132, 129, 121, 135, 148, 148, 136, 119,
104, 118, 115, 126, 141, 135, 125, 149, 170, 170, 158, 133, 114,
140, 145, 150, 178, 163, 172, 178, 199, 199, 184, 162, 146, 166,
171, 180, 193, 181, 183, 218, 230, 242, 209, 191, 172, 194, 196,
196, 236, 235, 229, 243, 264, 272, 237, 211, 180, 201, 204, 188,
235, 227, 234, 264, 302, 293, 259, 229, 203, 229, 242, 233, 267,
269, 270, 315, 364, 347, 312, 274, 237, 278, 284, 277, 317, 313,
318, 374, 413, 405, 355, 306, 271, 306, 315, 301, 356, 348, 355,
422, 465, 467, 404, 347, 305, 336, 340, 318, 362, 348, 363, 435,
491, 505, 404, 359, 310, 337, 360, 342, 406, 396, 420, 472, 548,
559, 463, 407, 362, 405, 417, 391, 419, 461, 472, 535, 622, 606,
508, 461, 390, 432), .Tsp = c(1949, 1960.91666666667, 12), class = "ts")
```

openxlsx 可以读写 XLSX 文档

美团使用的大数据工具有很多，最常用的 Hive、Spark、Kylin、Impala、Presto 等，详见 https://tech.meituan.com/2018/08/02/mt-r-practice.html。下面主要介绍如何在 R 中连接 MySQL、Presto 和 Spark。

sparklyr.flint 支持 Spark 的时间序列库 flint，sparkxgb 为 Spark 上的 XGBoost 提供 R 接口，sparkwarc 支持加载 Web ARChive 文件到 Spark 里 sparkavro 支持从 Apache Avro (https://avro.apache.org/) 读取文件到 Spark 里，sparkbq 是一个 sparkly 扩展包，集成 Google BigQuery 服务，geospark 提供 GeoSpark

非商业 ©

库的 R 接口，并且以 sf 的数据操作方式，rsparkling H2O Sparkling Water 机器学习库的 R 接口。

Spark 性能优化，参考三篇博文

- Spark 在美团的实践
- Spark 性能优化指南——基础篇
- Spark 性能优化指南——高级篇

其他材料

- 朱俊晖收集的 Spark 资源列表 https://github.com/harryprince/awesome-sparklyr，推荐使用 sparklyr https://github.com/sparklyr/sparklyr 连接 Spark
- Spark 与 R 语言 https://docs.microsoft.com/en-us/azure/databricks/spark/latest/sparkr/
- Mastering Spark with R https://therinspark.com/

3.8 Spark 与 R 语言

3.8.1 sparklyr

警告

Spark 依赖特定版本的 Java、Hadoop，三者之间的版本应该要相融。

在 MacOS 上配置 Java 环境，注意 Spark 仅支持 Java 8 至 11，所以安装指定版本的 Java 开发环境

```
# 安装 openjdk 11
brew install openjdk@11
# 全局设置 JDK 11
sudo ln -sfn /usr/local/opt/openjdk@11/libexec/openjdk.jdk /Library/Java/JavaVirtualMachines/openjdk-11.jd
# Java 11 JDK 添加到 .zshrc
export CPPFLAGS="-I/usr/local/opt/openjdk@11/include"
export PATH="/usr/local/opt/openjdk@11/bin:$PATH"
```

配置 R 环境，让 R 能够识别 Java 环境，再安装 **rJava** 包

```
# 配置
sudo R CMD javareconf
# 系统软件依赖
brew install pcre2
# 安装 rJava
Rscript -e 'install.packages("rJava", type="source")'
```

最后安装 **sparklyr** 包，以及 Spark 环境，可以借助 `spark_install()` 安装 Spark，比如下面 Spark 3.0 连同 hadoop 2.7 一起安装。

```
install.packages('sparklyr')
sparklyr::spark_install(version = '3.0', hadoop_version = '2.7')
```

也可以先手动下载 Spark 软件环境，建议选择就近镜像站点下载，比如在北京选择清华站点 https://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-3.0.1/spark-3.0.1-bin-hadoop2.7.tgz，此

© 清华大学出版社

环境自带 R 和 Python 接口。为了供 sparklyr 调用，先设置 SPARK_HOME 环境变量指向 Spark 安装位置，再连接单机版 Spark。

```
# 排错 https://github.com/sparklyr/sparklyr/issues/2827
options(sparklyr.log.console = FALSE)
# 连接 Spark
library(sparklyr)
library(ggplot2)
sc <- spark_connect(
  master = "local",
  # config = list(sparklyr.gateway.address = "127.0.0.1"),
  spark_home = Sys.getenv("SPARK_HOME")
)
# diamonds 数据集导入 Spark
diamonds_tbl <- copy_to(sc, ggplot2::diamonds, "diamonds")
```

做数据的聚合统计，有两种方式。一种是使用用 R 包 dplyr 提供的数据操作语法，下面以按 cut 分组统计钻石的数量为例，说明 dplyr 提供的数据操作方式。

```
library(dplyr)
# 列出数据源下所有的表 tbls
src_tbls(sc)

diamonds_tbl <- diamonds_tbl %>%
  group_by(cut) %>%
  summarise(cnt = n()) %>%
  collect
```

另一种是使用结构化查询语言 SQL，这自不必说，大多数情况下，使用和一般的 SQL 没什么两样。

```
library(DBI)
diamonds_preview <- dbGetQuery(sc, "SELECT count(*) as cnt, cut FROM diamonds GROUP BY cut")
diamonds_preview
```

```
##     cnt       cut
## 1 21551     Ideal
## 2 13791   Premium
## 3  4906      Good
## 4  1610      Fair
## 5 12082 Very Good
```

©清华大学出版社

```
# SQL 中的 AVG 和 R 中的 mean 函数是类似的
diamonds_price <- dbGetQuery(sc, "SELECT AVG(p                    BY cut")
diamonds_price
```

```
##   mean_price       cut
## 1   3457.542     Ideal
## 2   4584.258   Premium
## 3   3928.864      Good
## 4   4358.758      Fair
## 5   3981.760 Very Good
```

```
library(ggplot2)
library(data.table)
diamonds <- as.data.table(diamonds)
diamonds[,.(mean_price = mean(price)), by = .(cut)]
```

```
##          cut mean_price
## 1:     Ideal   3457.542
## 2:   Premium   4584.258
## 3:      Good   3928.864
## 4: Very Good   3981.760
## 5:      Fair   4358.758
```

将结果数据用 ggplot2 呈现出来

```
ggplot(diamonds_preview, aes(cut, cnt)) +
  geom_col() +
  theme_minimal()
```

diamonds 数据集总共 53940 条数据，下面用 BUCKET 分桶抽样，将原数据随机分成 1000 个桶，取其中的一个桶，由于是随机分桶，所以每次的结果都不一样，解释详见https://spark.apache.org/docs/latest/sql-ref-syntax-qry-select-sampling.html

```
diamonds_sample <- dbGetQuery(sc, "SELECT * FROM diamonds TABLESAMPLE (BUCKET 1 OUT OF 1000) LIMIT 6")
diamonds_sample
```

```
##   carat       cut color clarity depth table price    x    y    z
## 1  0.82     Ideal     I     SI1  61.8    55  2959 6.00 6.03 3.72
## 2  0.88 Very Good     E     SI2  59.1    58  3150 6.22 6.30 3.70
```

©清华大学出版社


```
## 3  0.78     Ideal     D     VS2  61.7    57  3595 5.87 5.93 3.64
## 4  0.91 Very Good     I     VS2  61.0    57  3664 6.20 6.26 3.80
## 5  1.01      Fair     F     SI2  58.2    61  4127 6.48 6.52 3.78
## 6  1.01   Premium     H     SI1  60.3    59  4191 6.45 6.42 3.88
```

将抽样的结果用窗口函数 RANK() 排序，详见 https://spark.apache.org/docs/latest/sql-ref-syntax-qry-select-window.html

窗口函数 https://www.cnblogs.com/ZackSun/p/9713435.html

```
diamonds_rank <- dbGetQuery(sc, "
  SELECT cut, price, RANK() OVER (PARTITION BY cut ORDER BY price) AS rank
  FROM diamonds TABLESAMPLE (BUCKET 1 OUT OF 1000)
  LIMIT 6
")
diamonds_rank
```

```
##    cut price rank
## 1 Fair  2912    1
## 2 Fair  4915    2
## 3 Fair  4928    3
## 4 Fair 11156    4
## 5 Good   830    1
## 6 Good  1034    2
```

LATERAL VIEW 把一列拆成多行

https://liam.page/2020/03/09/LATERAL-VIEW-in-Hive-SQL/ https://spark.apache.org/docs/latest/sql-ref-syntax-qry-select-lateral-view.html

创建数据集

```
# 先删除存在的表 person
dbGetQuery(sc, "DROP TABLE IF EXISTS person")
# 创建表 person
dbGetQuery(sc, "CREATE TABLE IF NOT EXISTS person (id INT, name STRING, age INT, class INT, address S
# 插入数据到表 person
dbGetQuery(sc, "
INSERT INTO person VALUES
    (100, 'John', 30, 1, 'Street 1'),
    (200, 'Mary', NULL, 1, 'Street 2'),
    (300, 'Mike', 80, 3, 'Street 3'),
    (400, 'Dan', 50, 4, 'Street 4')
")
```

查看数据集

```
dbGetQuery(sc, "SELECT * FROM person")
```

```
##    id name age class  address
## 1 100 John  30     1 Street 1
```

© 草稿

```
## 2 200 Mary  NA     1 Street 2
## 3 300 Mike  80     3 Street 3
## 4 400  Dan  50     4 Street 4
```

行列转换 https://www.cnblogs.com/kimbo/p/6208973.html，LATERAL VIEW 展开

```
dbGetQuery(sc,"
SELECT * FROM person
    LATERAL VIEW EXPLODE(ARRAY(30, 60)) tabelName AS c_age
    LATERAL VIEW EXPLODE(ARRAY(40, 80)) AS d_age
LIMIT 6
")
```

```
##    id name age class  address c_age d_age
## 1 100 John  30     1 Street 1    30    40
## 2 100 John  30     1 Street 1    30    80
## 3 100 John  30     1 Street 1    60    40
## 4 100 John  30     1 Street 1    60    80
## 5 200 Mary  NA     1 Street 2    30    40
## 6 200 Mary  NA     1 Street 2    30    80
```

日期相关的函数 https://spark.apache.org/docs/latest/sql-ref-functions-builtin.html#date-and-timestamp-functions

```
# 今天
dbGetQuery(sc, "select current_date")
```

```
##   current_date()
## 1     2022-04-13
```

```
# 昨天
dbGetQuery(sc, "select date_sub(current_date, 1)")
```

```
##   date_sub(current_date(), 1)
## 1                  2022-04-12
```

```
# 本月最后一天 current_date 所属月份的最后一天
dbGetQuery(sc, "select last_day(current_date)")
```

```
##   last_day(current_date())
## 1               2022-04-30
```

```
# 星期几
dbGetQuery(sc, "select dayofweek(current_date)")
```

```
##   dayofweek(current_date())
## 1                         4
```

最后，使用完记得关闭 Spark 连接

```
spark_disconnect(sc)
```

© 黄湘云

3.8.2 SparkR

注意

考虑到和第3.8.1节的重合性，以及 sparklyr 的优势，本节代码都不会执行，仅作为补充信息予以描述。完整的介绍见 SparkR 包

```
if (nchar(Sys.getenv("SPARK_HOME")) < 1) {
  Sys.setenv(SPARK_HOME = "/opt/spark/spark-3.0.1-bin-hadoop2.7")
}
library(SparkR, lib.loc = c(file.path(Sys.getenv("SPARK_HOME"), "R", "lib")))
sparkR.session(master = "local[*]", sparkConfig = list(spark.driver.memory = "2g"))
```

警告

SparkR 要求 Java 版本满足：大于等于 8，而小于 12，本地 MacOS 安装高版本，比如 oracle-jdk 16.0.1 会报不兼容的错误。

```
Spark package found in SPARK_HOME: /opt/spark/spark-3.1.1-bin-hadoop3.2
Error in checkJavaVersion() :
  Java version, greater than or equal to 8 and less than 12, is required for this package; found version
```

`sparkConfig` 有哪些参数可以传递

Property Name	Property group	`spark-submit` equivalent
`spark.master`	Application Properties	`--master`
`spark.kerberos.keytab`	Application Properties	`--keytab`
`spark.kerberos.principal`	Application Properties	`--principal`
`spark.driver.memory`	Application Properties	`--driver-memory`
`spark.driver.extraClassPath`	Runtime Environment	`--driver-class-path`
`spark.driver.extraJavaOptions`	Runtime Environment	`--driver-java-options`
`spark.driver.extraLibraryPath`	Runtime Environment	`--driver-library-path`

将 data.frame 转化为 SparkDataFrame

```
faithful_sdf <- as.DataFrame(faithful)
```

SparkDataFrame

```
head(faithful_sdf)
```

查看结构

```
str(faithful_sdf)
```

3.9 数据库与 R 语言

Presto 的 R 接口 https://github.com/prestodb/RPresto 和文档 https://prestodb.io/docs/current/index.html，Presto 数据库

© 黄湘云

```
install.packages('RPresto')
```

MySQL 为例介绍 odbc 的连接和使用，详见 从 R 连接 MySQL

```
-- !preview conn=DBI::dbConnect(odbc::odbc(),  driver = "MariaDB", database = "demo",
--                              uid = "root", pwd = "cloud", host = "localhost", port = 3306)

SELECT * FROM mtcars
LIMIT 10
```

我的系统已经安装了多款数据库的 ODBC 驱动

```
dnf install -y unixODBC unixODBC-devel mariadb mariadb-server mariadb-devel mariadb-connector-odbc
```

```
odbc::odbcListDrivers()
```

```
# Driver from the mariadb-connector-odbc package
# Setup from the unixODBC package
[MariaDB]
Description     = ODBC for MariaDB
Driver          = /usr/lib/libmaodbc.so
Driver64        = /usr/lib64/libmaodbc.so
FileUsage       = 1
```

3.10 批量读取 csv 文件

iris 数据转化为 data.table 类型，按照 Species 分组拆成单独的 csv 文件，各个文件的文件名用鸢尾花的类别名表示

```
# 批量分组导出
library(data.table)
as.data.table(iris)[, fwrite(.SD, paste0("data/user_", unique(Species), ".csv")), by = Species, .SDcols =
```

读取文件夹 data/ 所有 csv 数据文件

```
library(data.table)
merged_df <- do.call('rbind', lapply(list.files(pattern = "*.csv", path = "data/"), fread ) )
# 或者
merged_df <- rbindlist(lapply(list.files(pattern = "*.csv", path = "data/"), fread ))
```

```
xdf$date <- as.Date(xdf$date)
xdf$ts <- as.POSIXct(as.numeric(xdf$ts), origin = "1978-01-01")
split(xdf[order(xdf$ts), ], interaction(xdf$study, xdf$port)) %>%
  lapply(function(.x) {
    .x[nrow(.x), ]
  }) %>%
  unname() %>%
  Filter(function(.x) {
    nrow(.x) > 0
```

©清华大学出版社

```
  }, .) %>%
  do.call(rbind.data.frame, .)

library(dplyr)
xdf %>%
  mutate(
    date = as.Date(date),
    ts = anytime::anytime(as.numeric(ts))
  ) %>%
  arrange(ts) %>%
  group_by(study, port) %>%
  slice(n()) %>%
  ungroup()
```

```
library(tibble)
library(magrittr)

mtcars <- tibble(mtcars)

mtcars %>%
  print(n = 16, width = 69)
```

```
## # A tibble: 32 x 11
##      mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  21       6  160    110  3.9   2.62  16.5     0     1     4     4
##  2  21       6  160    110  3.9   2.88  17.0     0     1     4     4
##  3  22.8     4  108     93  3.85  2.32  18.6     1     1     4     1
##  4  21.4     6  258    110  3.08  3.22  19.4     1     0     3     1
##  5  18.7     8  360    175  3.15  3.44  17.0     0     0     3     2
##  6  18.1     6  225    105  2.76  3.46  20.2     1     0     3     1
##  7  14.3     8  360    245  3.21  3.57  15.8     0     0     3     4
##  8  24.4     4  147.    62  3.69  3.19  20       1     0     4     2
##  9  22.8     4  141.    95  3.92  3.15  22.9     1     0     4     2
## 10  19.2     6  168.   123  3.92  3.44  18.3     1     0     4     4
## 11  17.8     6  168.   123  3.92  3.44  18.9     1     0     4     4
## 12  16.4     8  276.   180  3.07  4.07  17.4     0     0     3     3
## 13  17.3     8  276.   180  3.07  3.73  17.6     0     0     3     3
## 14  15.2     8  276.   180  3.07  3.78  18       0     0     3     3
## 15  10.4     8  472    205  2.93  5.25  18.0     0     0     3     4
## 16  10.4     8  460    215  3     5.42  17.8     0     0     3     4
## # ... with 16 more rows
```

```
mtcars %>%
  print(., n = nrow(.)/4)
```

```
## # A tibble: 32 x 11
```

©黄湘云

```
##     mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  21       6  160    110  3.9   2.62  16.5     0     1     4     4
## 2  21       6  160    110  3.9   2.88  17.0     0     1     4     4
## 3  22.8     4  108     93  3.85  2.32  18.6     1     1     4     1
## 4  21.4     6  258    110  3.08  3.22  19.4     1     0     3     1
## 5  18.7     8  360    175  3.15  3.44  17.0     0     0     3     2
## 6  18.1     6  225    105  2.76  3.46  20.2     1     0     3     1
## 7  14.3     8  360    245  3.21  3.57  15.8     0     0     3     4
## 8  24.4     4  147.    62  3.69  3.19  20       1     0     4     2
## # ... with 24 more rows
```

3.11 批量导出 xlsx 文件

将 R 环境中的数据集导出为 xlsx 表格

```
## 加载 openxlsx 包
library(openxlsx)
## 创建空白的工作薄，标题为鸢尾花数据集
wb <- createWorkbook(title = "鸢尾花数据集")
## 添加 sheet 页
addWorksheet(wb, sheetName = "iris")
# 将数据写进 sheet 页
writeData(wb, sheet = "iris", x = iris, colNames = TRUE)
# 导出数据到本地
saveWorkbook(wb, file = "iris.xlsx", overwrite = TRUE)
```

```
library(openxlsx)
xlsxFile <- system.file("extdata", "readTest.xlsx", package = "openxlsx")
# 导入
dat = read.xlsx(xlsxFile = xlsxFile)
# 导出
write.xlsx(dat, xlsxfile)
```

3.12 运行环境

```
xfun::session_info()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Locale:
##   LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
```

© 黄湘云


```
##   LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##   LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##   LC_PAPER=en_US.UTF-8       LC_NAME=C
##   LC_ADDRESS=C               LC_TELEPHONE=C
##   LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## Package version:
##   askpass_1.1        assertthat_0.2.1   base64enc_0.1-3    blob_1.2.2
##   bookdown_0.25      bslib_0.3.1        cli_3.2.0          codetools_0.2.18
##   colorspace_2.0-3   compiler_4.1.3     config_0.3.1       cpp11_0.4.2
##   crayon_1.5.1       curl_4.3.2         data.table_1.14.2  DBI_1.1.2
##   dbplyr_2.1.1       digest_0.6.29      dplyr_1.0.8        ellipsis_0.3.2
##   evaluate_0.15      fansi_1.0.3        farver_2.1.0       fastmap_1.1.0
##   forge_0.2.0        fs_1.5.2           generics_0.1.2     ggplot2_3.3.5
##   globals_0.14.0     glue_1.6.2         graphics_4.1.3     grDevices_4.1.3
##   grid_4.1.3         gtable_0.3.0       highr_0.9          htmltools_0.5.2
##   htmlwidgets_1.5.4  httr_1.4.2         isoband_0.2.5      jquerylib_0.1.4
##   jsonlite_1.8.0     knitr_1.38         labeling_0.4.2     lattice_0.20.45
##   lifecycle_1.0.1    magrittr_2.0.3     MASS_7.3.56        Matrix_1.4.1
##   methods_4.1.3      mgcv_1.8.40        mime_0.12          munsell_0.5.0
##   nlme_3.1.157       openssl_2.0.0      parallel_4.1.3     pillar_1.7.0
##   pkgconfig_2.0.3    png_0.1-7          purrr_0.3.4        r2d3_0.2.6
##   R6_2.5.1           rappdirs_0.3.3     RColorBrewer_1.1.2 rlang_1.0.2
##   rmarkdown_2.13     rprojroot_2.0.2    rstudioapi_0.13    sass_0.4.1
##   scales_1.1.1       sparklyr_1.7.5     splines_4.1.3      stats_4.1.3
##   stringi_1.7.6      stringr_1.4.0      sys_3.4            sysfonts_0.8.8
##   tibble_3.1.6       tidyr_1.2.0        tidyselect_1.1.2   tinytex_0.38
##   tools_4.1.3        utf8_1.2.2         utils_4.1.3        uuid_1.0.4
##   vctrs_0.4.0        viridisLite_0.4.0  withr_2.5.0        xfun_0.30
##   xml2_1.3.3         yaml_2.3.5
```

© 基础

第四章　字符串操作

Handling Strings with R 和 R for Data Science 提供字符串入门介绍，Sara Stoudt 整理了 stringr 包与 Base R 正则表达式函数的对应表 https://stringr.tidyverse.org/articles/from-base.html

stringr 基于 stringi 包字符串处理包，re2r 包基于 Google 开发的 C++ 库 re2，Google 编程之夏项目提供了一份 正则表达式性能综述，stringdist Approximate String Matching and String Distance Functions 近似字符串匹配和字符串距离计算函数 [van der Loo, 2014]

- janitor
- Manipulating strings with the stringr package
- filesstrings 基于 stringr 操作字符串
- strex 一些没有包含在 stringr 或者 stringi 中的字符串操作函数
- tidytext Text mining using dplyr, ggplot2, and other tidy tools

stringdist stringfish stringb stringi stringr

字符和字符串类型的数据值得单独拿出来讲，不仅因为内容多，而且比较难，应用范围最广，特别是面对文本类型的数据时，几乎是避不开的！R 的前身是 S，S 的前身是一些 Fortran 和 C 子程序，最早在贝尔实验室是用于文本分析领域，因此在 R 基础包中提供了丰富的字符串处理函数，你可以在 R 控制台中执行如下一行命令查看

```
help.search(keyword = "character", package = "base")
```

本章主要介绍 R 内置的字符串操作函数

4.1　字符数统计

`nchar` 函数统计字符串向量中每个元素的字符个数，注意与函数 `length` 的差别，它统计向量中元素的个数，即向量的长度。

```
nchar(c("Hello", "world", "!"))
```

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

```
R.version.string
```

```
## [1] "R version 4.1.3 (2022-03-10)"
```

```
nchar(R.version.string)
```

```
## [1] 28
```

© 黄湘云


```
deparse(base::mean)
```

```
## [1] "function (x, ...) "  "UseMethod(\"mean\")"
```

```
nchar(deparse(base::mean))
```

```
## [1] 18 17
```

一些特殊的情况

```
nchar("")
```

```
## [1] 0
```

```
nchar(NULL)
```

```
## integer(0)
```

```
nchar(0)
```

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

```
pi
```

```
## [1] 3.141593
```

```
nchar(pi)
```

```
## [1] 16
```

```
exp(1)
```

```
## [1] 2.718282
```

```
nchar(exp(1))
```

```
## [1] 16
```

```
nchar(NA)
```

```
## [1] NA
```

4.2 字符串翻译

tolower 将字符串或字符串向量中含有的大写字母全都转化为小写，toupper 函数正好与之相反.

```
tolower(c("HELLO", "Hello, R", "hello"))
```

```
## [1] "hello"    "hello, r" "hello"
```

```
toupper(c("HELLO", "Hello, R", "hello"))
```

```
## [1] "HELLO"    "HELLO, R" "HELLO"
```

©黄湘云

4.3 字符串连接

paste 函数设置参数 sep 作为连接符，设置参数 collapse 可以将字符串拼接后连成一个字符串

```
paste("A", "B", sep = "")
```

```
## [1] "AB"
```

```
paste(c("A", "B", "C"), 1:3, sep = "-")
```

```
## [1] "A-1" "B-2" "C-3"
```

```
paste(c("A", "B", "C"), 1:3, sep = "-", collapse = ";")
```

```
## [1] "A-1;B-2;C-3"
```

paste0 相当于 sep 设为空，没有连接符

```
paste0("A", "B")
```

```
## [1] "AB"
```

```
paste0(c("A", "B", "C"), 1:3)
```

```
## [1] "A1" "B2" "C3"
```

```
paste0(c("A", "B", "C"), 1:3, collapse = ";")
```

```
## [1] "A1;B2;C3"
```

4.4 字符串拆分

```
strsplit(x, split, fixed = FALSE, perl = FALSE, useBytes = FALSE)
```

strsplit 函数用于字符串拆分，参数 x 是被拆分的字符串向量，其每个元素都会被拆分，而参数 split 表示拆分的位置，可以用正则表达式来描述位置，拆分的结果是一个列表。

参数 fixed 默认设置 fixed = FALSE 表示正则表达式匹配，而 fixed = TRUE 表示正则表达式的精确匹配或者按文本字符的字面意思匹配，即按普通文本匹配。我们知道按普通文本匹配速度快。

当启用 perl = TRUE 时，由 PCRE_use_JIT 控制细节。perl 参数的设置与 Perl 软件版本有关，如果正则表达式很长，除了正确设置正则表达式，使用 perl = TRUE 可以提高运算速度

参数 useBytes 设置是否按照逐个字节地进行匹配，默认设置为 FALSE，即按照字符而不是字节进行匹配

```
x <- c(as = "asfef", qu = "qwerty", "yuiop[", "b", "stuff.blah.yech")
# 按字母 e 拆分字符串向量 x
strsplit(x, "e")
```

```
## $as
## [1] "asf" "f"
##
## $qu
## [1] "qw"  "rty"
```

© 黄湘云


```
##
## [[3]]
## [1] "yuiop["
##
## [[4]]
## [1] "b"
##
## [[5]]
## [1] "stuff.blah.y" "ch"
```

参数 split 支持通过正则表达式的方式指明拆分位置

```r
# 默认将点号 . 看作一个正则表达式，它是一个元字符，匹配任意字符
strsplit("a.b.c", ".")
```

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

```r
# 这才是按点号拆分
strsplit("a.b.c", ".", fixed = TRUE)
```

```
## [[1]]
## [1] "a" "b" "c"
```

```r
# 或者
strsplit("a.b.c", "[.]")
```

```
## [[1]]
## [1] "a" "b" "c"
```

```r
# 或者转义点号，去掉元字符的特殊意义
strsplit("a.b.c", "\\.")
```

```
## [[1]]
## [1] "a" "b" "c"
```

这里介绍一个将字符串逆序的函数 `str_rev`

```r
str_rev <- function(x)
        sapply(lapply(strsplit(x, NULL), rev), paste, collapse = "")
str_rev(c("abc", "Statistics"))
```

```
## [1] "cba"        "scitsitatS"
```

为了加深理解，再举几个例子

```r
# 最后一个空字符没有产生
strsplit(paste(c("", "a", ""), collapse="#"), split="#")
```

```
## [[1]]
## [1] ""  "a"
```

© 非洲出版

```r
# 空字符只有有定义的时候才会产生
strsplit("", " ")
```

```
## [[1]]
## character(0)
```

```r
strsplit(" ", " ")
```

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

4.5 字符串匹配

agrep 和 agrepl 函数做近似（模糊）匹配 (Approximate Matching or Fuzzy Matching)，对于匹配，考虑到参数 pattern 在参数 x 中匹配时，允许参数值 x 存在最小可能的插入、删除和替换，这种修改叫做 Levenshtein 编辑距离，max.distance 控制其细节

```r
agrep(pattern, x, max.distance = 0.1, costs = NULL,
      ignore.case = FALSE, value = FALSE, fixed = TRUE,
      useBytes = FALSE)

agrepl(pattern, x, max.distance = 0.1, costs = NULL,
       ignore.case = FALSE, fixed = TRUE, useBytes = FALSE)
```

agrep 函数返回 pattern 在 x 中匹配到的一个位置向量，agrepl 返回一个逻辑向量，这一点类似 grep 和 grepl 这对函数，下面举例子说明

```r
agrep("lasy", "1 lazy 2")
```

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

```r
# sub = 0 表示匹配时不考虑替换
agrep("lasy", c(" 1 lazy 2", "1 lasy 2"), max = list(sub = 0))
```

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

```r
# 默认设置下，匹配时区分大小写
agrep("laysy", c("1 lazy", "1", "1 LAZY"), max = 2)
```

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

```r
# 返回匹配到值，而不是位置下标，类似 grep(..., value = TRUE) 的返回值
agrep("laysy", c("1 lazy", "1", "1 LAZY"), max = 2, value = TRUE)
```

```
## [1] "1 lazy"
```

```r
# 不区分大小写
agrep("laysy", c("1 lazy", "1", "1 LAZY"), max = 2, ignore.case = TRUE)
```

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

©黄湘云


```
startsWith(x, prefix)
  endsWith(x, suffix)
```

startsWith 和 endsWith 函数用来匹配字符串的前缀和后缀，返回值是一个逻辑向量，参数 prefix 和 suffix 不要包含特殊的正则表达式字符，如点号.，举例子

```
# 字符串向量
search()
```

```
## [1] ".GlobalEnv"        "package:stats"     "package:graphics"
## [4] "package:grDevices" "package:utils"     "package:datasets"
## [7] "package:methods"   "Autoloads"         "package:base"
```

```
# 匹配以 package: 开头的字符串
startsWith(search(), "package:")
```

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

```
# 或者
grepl("^package:", search())
```

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

当前目录下，列出扩展名为 .Rmd 的文件

```
# list.files(path = ".", pattern = "\\.Rmd$")
# 而不是 endsWith(list.files(), "\\.Rmd")
endsWith(list.files(), ".Rmd")
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
##  [13]  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [25] FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
##  [37] FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
##  [61]  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE
##  [73] FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE
##  [85]  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
##  [97]  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [109]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE
## [121]  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
## [133]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE
## [145] FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [157]  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE
## [169] FALSE FALSE  TRUE  TRUE
```

```
# 或者
grepl("\\.Rmd$", list.files())
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE
##  [13]  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [25] FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
```

© 黄湘云

```
##  [37] FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE  TRUE
##  [61]  TRUE FALSE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE FALSE  TRUE
##  [73] FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE
##  [85]  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE
##  [97]  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [109]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE
## [121]  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE
## [133]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE
## [145] FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [157]  TRUE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE
## [169] FALSE FALSE  TRUE  TRUE
```

部分匹配 (Partial String Matching)

```
match(x, table, nomatch = NA_integer_, incomparables = NULL)
x %in% table
charmatch(x, table, nomatch = NA_integer_)
pmatch(x, table, nomatch = NA_integer_, duplicates.ok = FALSE)
```

这几个 `match` 函数的返回值都是一个向量，每个元素是参数 x 在参数 table 中第一次匹配到的位置，`charmatch` 与 `pmatch(x, table, nomatch = NA_integer_, duplicates.ok = TRUE)` 类似，所以 `pmatch` 在默认 `duplicates.ok = FALSE` 的情况下，若 x 在第二个参数 table 中有多次匹配就会返回 NA，因此，实际上 `pmatch` 只允许在第二个参数中匹配一次

```
match("xx", c("abc", "xx", "xxx", "xx"))
```

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

```
1:10 %in% c(1,3,5,9)
```

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

```
# charmatch 就比较奇怪，规则太多
charmatch("", "")                              # returns 1
```

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

```
# 多个精确匹配到，或者多个部分匹配到，则返回 0
charmatch("m",   c("mean", "median", "mode", "quantile")) # returns 0
```

```
## [1] 0
```

```
# med 只在table参数值的第二个位置部分匹配到，所以返回2
charmatch("med", c("mean", "median", "mode", "quantile")) # returns 2
```

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

```
charmatch("xx", "xx")
```

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

© 王建硕


```
charmatch("xx", "xxa")
```

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

```
charmatch("xx", "axx")
```

```
## [1] NA
```

```
# 注意比较与 charmatch 的不同
pmatch("", "")                          # returns NA
```

```
## [1] NA
```

```
pmatch("m",   c("mean", "median", "mode")) # returns NA
```

```
## [1] NA
```

```
pmatch("med", c("mean", "median", "mode")) # returns 2
```

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

4.6 字符串查询

```
grep(pattern, x,
  ignore.case = FALSE, perl = FALSE, value = FALSE,
  fixed = FALSE, useBytes = FALSE, invert = FALSE
)
grepl(pattern, x,
  ignore.case = FALSE, perl = FALSE,
  fixed = FALSE, useBytes = FALSE
)
```

grep 和 grepl 是一对字符串查询函数，查看字符串向量 x 中是否包含正则表达式 pattern 描述的内容

- ignore.case: TRUE 表示忽略大小写，FALSE 表示匹配的时候区分大小写
- fixed = TRUE 表示启用 literal regular expression 字面正则表达式，默认情况下 fixed = FALSE
- grep 函数返回匹配到的字符串向量 x 的元素的下标，如果 value=TRUE 则返回下标对应的值
- grepl 函数返回一个逻辑向量，检查字符串向量 x 中的每个元素是否匹配到，匹配到返回 TRUE，没有匹配到返回 FALSE

```
# 返回下标位置
grep("[a-z]", letters)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26
```

```
# 返回查询到的值
grep("[a-z]", letters, value = TRUE)
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

©黄湘云

继续举例子

```
grep(x = c("apple", "banana"), pattern = "a")
```

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

```
grep(x = c("apple", "banana"), pattern = "b")
```

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

```
grep(x = c("apple", "banana"), pattern = "a", value = TRUE)
```

```
## [1] "apple"  "banana"
```

```
grep(x = c("apple", "banana"), pattern = "b", value = TRUE)
```

```
## [1] "banana"
```

关于 grepl 函数的使用例子

```
grepl(x = c("apple", "banana"), pattern = "a")
```

```
## [1] TRUE TRUE
```

```
grepl(x = c("apple", "banana"), pattern = "b")
```

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

R 语言是用字符串来表示正则表达式的，但是正则表达式不是字符串，字符串的构造类似算术表达式

在 R 里面分别表示 a\\b 和 a\b

```
writeLines(c("a\\\\b", "a\\b"))
```

```
## a\\b
## a\b
```

下面在 R 里面分别匹配字符串 a\\b 和 a\b 中的 \\ 和 \

```
# 匹配字符串中的一个反斜杠
grep(x = c("a\\\\b", "a\\b"), pattern = "\\", value = TRUE, fixed = TRUE)
```

```
## [1] "a\\\\b" "a\\b"
```

```
grep(x = c("a\\\\b", "a\\b"), pattern = "\\\\", value = TRUE, fixed = FALSE)
```

```
## [1] "a\\\\b" "a\\b"
```

```
# 匹配字符串中的两个反斜杠
grep(x = c("a\\\\b", "a\\b"), pattern = "\\\\", value = TRUE, fixed = TRUE)
```

```
## [1] "a\\\\b"
```

```
grep(x = c("a\\\\b", "a\\b"), pattern = "\\\\\\\\", value = TRUE, fixed = FALSE)
```

```
## [1] "a\\\\b"
```

```
# 匹配字符串中的两个反斜杠 \\
grepl(x = "a\\\\b", pattern = "\\\\\\\\", fixed = FALSE)
```

© 黄湘云


```
## [1] TRUE
```

```
grepl(x = "a\\\\b", pattern = "\\\\\\\\", fixed = TRUE)
```

```
## [1] FALSE
```

```
grepl(x = "a\\\\b", pattern = "\\\\", fixed = TRUE)
```

```
## [1] TRUE
```

```
regexpr(pattern, text,
  ignore.case = FALSE, perl = FALSE,
  fixed = FALSE, useBytes = FALSE
)
gregexpr(pattern, text,
  ignore.case = FALSE, perl = FALSE,
  fixed = FALSE, useBytes = FALSE
)
regexec(pattern, text,
  ignore.case = FALSE, perl = FALSE,
  fixed = FALSE, useBytes = FALSE
)
```

当启用 `perl=TRUE` 时，函数 `regexpr` 和 `gregexpr` 支持 Python 环境下的命名捕获 (named captures)，但是不支持长向量的输入。如果一个分组被命名了，如 `(?<first>[A-Z][a-z]+)` 那么匹配到的位置按命名返回。函数 `sub` 不支持命名反向引用 (Named backreferences)

函数 `regmatches` 用来提取函数 `regexpr`, `gregexpr` 和 `regexec` 匹配到的子字符串

`useBytes = FALSE` 匹配位置和长度默认是按照字符级别来的，如果 `useBytes = TRUE` 则是按照逐个字节的匹配结果

如果使用到了 **命名捕获**则会返回更多的属性 “capture.start”，“capture.length” 和 “capture.names”，分别表示捕获的起始位置、捕获的长度和捕获的命名。

- `regexpr` 函数返回一个整型向量，第一次匹配的初始位置，-1 表示没有匹配到，返回的属性 `match.length` 表示匹配的字符数量，是一个整型向量，向量长度是匹配的文本的长度，-1 表示没有匹配到

```
text <- c("Hellow, Adam!", "Hi, Adam!", "How are you, Adam.")
regexpr("Adam", text)
```

```
## [1]  9  5 14
## attr(,"match.length")
## [1] 4 4 4
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

```
txt <- c(
  "The", "licenses", "for", "most", "software", "are",
```

©黄湘云

```
  "designed", "to", "take", "away", "your", "freedom",
  "to", "share", "and", "change", "it.",
  "", "By", "contrast,", "the", "GNU", "General", "Public", "License",
  "is", "intended", "to", "guarantee", "your", "freedom", "to",
  "share", "and", "change", "free", "software", "--",
  "to", "make", "sure", "the", "software", "is",
  "free", "for", "all", "its", "users"
)
# gregexpr("en", txt)
regexpr("en", txt)
```

```
##  [1] -1  4 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  2 -1  4
## [26] -1  4 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
## attr(,"match.length")
##  [1] -1  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1  2 -1  2
## [26] -1  2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

- `gregexpr` 函数返回一个列表，返回列表的长度与字符串向量的长度一样，列表中每个元素的形式与 `regexpr` 的返回值一样, except that the starting positions of every (disjoint) match are given.

```
gregexpr("Adam", text)
```

```
## [[1]]
## [1] 9
## attr(,"match.length")
## [1] 4
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
##
## [[2]]
## [1] 5
## attr(,"match.length")
## [1] 4
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
##
## [[3]]
## [1] 14
## attr(,"match.length")
```

©黄湘云


```
## [1] 4
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

- regexec 函数返回一个列表，类似函数 gregexpr 的返回结果，长度与字符串向量的长度一样，如果没有匹配到就返回 -1，匹配到了就返回一个匹配的初值位置的整型序列，所有子字符串与括号分组的正则表达式的子表达式对应，属性 “match.length” 是一个表示匹配的长度的向量，如果是 -1 表示没有匹配到。位置、长度和属性的解释与 regexpr 一致

```
regexec("Adam", text)
```

```
## [[1]]
## [1] 9
## attr(,"match.length")
## [1] 4
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
##
## [[2]]
## [1] 5
## attr(,"match.length")
## [1] 4
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
##
## [[3]]
## [1] 14
## attr(,"match.length")
## [1] 4
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

由于资源限制（特别是 PCRE）导致的匹配失败，会视为没有匹配，通常伴随一个警告

下面这个将链接分解的例子由 Luke Tierney 提供[1]

```
x <- "http://stat.umn.edu:80/xyz"
m <- regexec("^(([^:]+)://)?([^:/]+)(:([0-9]+))?(/.*)", x)
m
```

[1] https://homepage.divms.uiowa.edu/~luke/R/regexp.html

© 基础

```
## [[1]]
## [1]  1  1  1  8 20 21 23
## attr(,"match.length")
## [1] 26  7  4 12  3  2  4
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

这里 x 是一个字符串，所以函数 `regexec` 返回的列表长度为 1，正则表达式 `^(([^:]+)://)?([^:/]+)(:([0-9]+))?(/.*)` 括号分组匹配到了 7 次，第一次匹配整个字符串，所以起始位置是 1，而匹配长度是 26，即整个字符串的长度，读者可以调用函数 `nchar(x)` 算一下，如果你愿意手动数一下也可以哈！余下不一一介绍，可以根据返回结果和图4.1一起看，最后还可以调用 `regmatches` 函数抽取匹配到的结果

```
regmatches(x, m)
```

```
## [[1]]
## [1] "http://stat.umn.edu:80/xyz" "http://"
## [3] "http"                       "stat.umn.edu"
## [5] ":80"                        "80"
## [7] "/xyz"
```

我们可以在 https://regex101.com/ 上测试表达式，如图4.1所示，表达式 `^(([^:]+)://)?([^:/]+)(:([0-9]+))?(/.*)` 包含 7 个组，每个组的匹配结果见图的右下角，这样我们不难理解，函数 `regmatches` 返回的第列表中，第 3 个位置是传输协议 protocol `http`，第 4 个位置是主机 host `stat.umn.edu`，第 6 个位置是端口 port `80`，第 7 个位置是路径 path `/xyz`，所以函数 `regmatches` 的作用就是根据函数 `regexec` 匹配的结果抽取子字符串。

图 4.1: 正则表达式匹配结果

进一步，我们可以用 `regmatches` 函数抽取 URL 的部分内容，如前面提到的传输协议，主机等

©黄湘云


```
URL_parts <- function(x) {
  m <- regexec("^(([^:]+)://)?([^:/]+)(:([0-9]+))?(/.*)", x)
  parts <- do.call(
    rbind,
    lapply(regmatches(x, m), `[`, c(3L, 4L, 6L, 7L))
    # 3,4,6,7是索引位置
  )
  colnames(parts) <- c("protocol", "host", "port", "path")
  parts
}
URL_parts(x)
```

```
##      protocol host           port path
## [1,] "http"   "stat.umn.edu" "80" "/xyz"
```

目前还没有 gregexec 函数，但是可以模拟一个，首先用 gregexpr 函数返回匹配的位置，regmatches 抽取相应的值，然后用 regexec 作用到每一个提取的值，做再一次匹配和值的抽取，实现了全部的匹配。另一个例子

```
## There is no gregexec() yet, but one can emulate it by running
## regexec() on the regmatches obtained via gregexpr().  E.g.:
pattern <- "([[:alpha:]]+)([[:digit:]]+)"
s <- "Test: A1 BC23 DEF456"
gregexpr(pattern, s)
```

```
## [[1]]
## [1]  7 10 15
## attr(,"match.length")
## [1] 2 4 6
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
```

```
regmatches(s, gregexpr(pattern, s))
```

```
## [[1]]
## [1] "A1"     "BC23"   "DEF456"
```

```
lapply(
  regmatches(s, gregexpr(pattern, s)),
  function(e) regmatches(e, regexec(pattern, e))
)
```

```
## [[1]]
## [[1]][[1]]
## [1] "A1" "A"  "1"
##
## [[1]][[2]]
```

© 黄湘云

```
## [1] "BC23" "BC"   "23"
##
## [[1]][[3]]
## [1] "DEF456" "DEF"    "456"
```

4.7 字符串替换

chartr 支持正则表达式的替换，chartr 是对应字符的替换操作

```
x <- "MiXeD cAsE 123"
# 将字符 iXs 替换为 why
chartr("iXs", "why", x)
```

```
## [1] "MwheD cAyE 123"
```

```
# 将字符串 a-cX 中的字符挨个对应地替换为 D-Fw
chartr("a-cX", "D-Fw", x)
```

```
## [1] "MiweD FAsE 123"
```

两个 *sub 函数的区别：sub 替换第一次匹配到的结果，gsub 替换所有匹配的结果

```
sub(" .*", "", extSoftVersion()["PCRE"])
```

```
##    PCRE
## "10.39"
```

参数 replacement 的值是正则表达式，其包含反向引用的用法，\\1 即引用表达式 ([ab])

```
gsub(pattern =  "([ab])", replacement = "\\1_\\1_", x = "abc and ABC")
```

```
## [1] "a_a_b_b_c a_a_nd ABC"
```

4.8 字符串提取

```
substr(x, start, stop)
substring(text, first, last = 1000000L)
```

substr 和 substring 函数通过位置进行字符串的拆分和提取，它们本身不使用正则表达式，结合其他正则表达式函数 regexpr, gregexpr 和 regexec，可以很方便地从大量文本中提取所需的信息。作用类似之前提到的 regmatches 函数

参数设置基本相同

- x/text 是要拆分的字符串向量
- start/first 截取的起始位置向量
- stop/last 截取的终止位置向量

返回值有差别

- substr 返回的字串个数等于第一个参数 x 的长度

© 基础

- substring 返回字串个数等于三个参数中最长向量长度，短向量循环使用。

```
x <- "123456789"
substr(x, c(2, 4), c(4, 5, 8))
```

```
## [1] "234"
```

```
substring(x, c(2, 4), c(4, 5, 8))
```

```
## [1] "234"     "45"      "2345678"
```

```
substr("abcdef", 2, 4)
```

```
## [1] "bcd"
```

```
substring("abcdef", 1:6, 1:6)
```

```
## [1] "a" "b" "c" "d" "e" "f"
```

因为 x 的向量长度为 1，所以 substr 获得的结果只有 1 个字串，即第 2 和第 3 个参数向量只用了第一个组合：起始位置 2，终止位置 4。而 substring 的语句三个参数中最长的向量为 c(4,5,8)，执行时按短向量循环使用的规则第一个参数事实上就是 c(x,x,x)，第二个参数就成了 c(2,4,2)，最终截取的字串起始位置组合为：2-4, 4-5 和 2-8。

```
x <- c("123456789", "abcdefghijklmnopq")
substr(x, c(2, 4), c(4, 5, 8))
```

```
## [1] "234" "de"
```

```
substring(x, c(2, 4), c(4, 5, 8))
```

```
## [1] "234"     "de"      "2345678"
```

更加高级的字符串抽取

```
# 从字符串中抽取固定模式的文本，替代 stringr::str_extract
# 只抽取一个匹配的
extract_str <- function(text, pattern) regmatches(text, regexpr(pattern, text))
# 符合模式的全部抽取
gextract_str <- function(text, pattern) regmatches(text, gregexpr(pattern, text))
```

举例子，抽取连续的数字

```
# 两个例子
extract_str(text = "abd123da345das", pattern = "(\\d+){3}")
```

```
## [1] "123"
```

```
gextract_str(text = "abd123da345das", pattern = "(\\d+){3}")
```

```
## [[1]]
## [1] "123" "345"
```

例子来自于 https://recology.info/2018/10/limiting-dependencies/

©黄湘云

4.9 命名捕捉

函数 regexpr(..., perl = TRUE) 和 gregexpr(..., perl = TRUE) 支持命名捕捉

```
## named capture
notables <- c("  Ben Franklin and Jefferson Davis",
              "\tMillard Fillmore")
# name groups 'first' and 'last'
name.rex <- "(?<first>[[:upper:]][[:lower:]]+) (?<last>[[:upper:]][[:lower:]]+)"

(parsed <- regexpr(name.rex, notables, perl = TRUE))
```

```
## [1] 3 2
## attr(,"match.length")
## [1] 12 16
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
## attr(,"capture.start")
##      first last
## [1,]     3    7
## [2,]     2   10
## attr(,"capture.length")
##      first last
## [1,]     3    8
## [2,]     7    8
## attr(,"capture.names")
## [1] "first" "last"
```

```
attr(parsed, 'capture.names')
```

```
## [1] "first" "last"
```

```
regmatches(notables, parsed)
```

```
## [1] "Ben Franklin"     "Millard Fillmore"
```

希望返回一个 data.frame，列名是指定的 named group 名字

```
# 有多个结果
(idx <- gregexpr(name.rex, notables, perl = TRUE))
```

```
## [[1]]
## [1]  3 20
## attr(,"match.length")
## [1] 12 15
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
```

©黄海洋


```
## [1] TRUE
## attr(,"capture.start")
##      first last
## [1,]     3    7
## [2,]    20   30
## attr(,"capture.length")
##      first last
## [1,]     3    8
## [2,]     9    5
## attr(,"capture.names")
## [1] "first" "last"
##
## [[2]]
## [1] 2
## attr(,"match.length")
## [1] 16
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
## attr(,"capture.start")
##      first last
## [1,]     2   10
## attr(,"capture.length")
##      first last
## [1,]     7    8
## attr(,"capture.names")
## [1] "first" "last"
```

```
regmatches(notables, idx)
```

```
## [[1]]
## [1] "Ben Franklin"    "Jefferson Davis"
##
## [[2]]
## [1] "Millard Fillmore"
```

```
attr(idx[[1]], 'capture.names')
```

```
## [1] "first" "last"
```

```
library(magrittr)
data.frame(notable = notables) %>%
tidyr::extract(
    notable, c("first", "last"), name.rex,
    remove = FALSE
  )
```

©黄海广

```
##                                notable   first     last
## 1   Ben Franklin and Jefferson Davis     Ben Franklin
## 2                  \tMillard Fillmore Millard Fillmore
```

4.10 精确匹配

fixed = TRUE

4.11 模糊匹配

近似字符串匹配 (Approximate String Matching) 也叫模糊匹配 (Fuzzy Matching)

agrep() agrepl() aregexec() adist()

```r
agrep(pattern = "lasy", x = "1 lazy 2")
```

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

```r
agrep("lasy", c(" 1 lazy 2", "1 lasy 2"), max = list(sub = 0))
```

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

```r
agrep("laysy", c("1 lazy", "1", "1 LAZY"), max = 2)
```

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

```r
agrep("laysy", c("1 lazy", "1", "1 LAZY"), max = 2, value = TRUE)
```

```
## [1] "1 lazy"
```

```r
agrep("laysy", c("1 lazy", "1", "1 LAZY"), max = 2, ignore.case = TRUE)
```

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

```r
agrepl(pattern = "lasy", x = "1 lazy 2")
```

```
## [1] TRUE
```

```r
## Cf. the examples for agrep.
x <- c("1 lazy", "1", "1 LAZY")

aregexec("laysy", x, max.distance = 2)
```

```
## [[1]]
## [1] 3
## attr(,"match.length")
## [1] 4
##
## [[2]]
## [1] -1
## attr(,"match.length")
```

©清华大学出版社


```
## [1] -1
##
## [[3]]
## [1] -1
## attr(,"match.length")
## [1] -1
```

```
aregexec("(lay)(sy)", x, max.distance = 2)
```

```
## [[1]]
## [1] 3 3 5
## attr(,"match.length")
## [1] 4 2 2
##
## [[2]]
## [1] -1
## attr(,"match.length")
## [1] -1
##
## [[3]]
## [1] -1
## attr(,"match.length")
## [1] -1
```

```
aregexec("(lay)(sy)", x, max.distance = 2, ignore.case = TRUE)
```

```
## [[1]]
## [1] 3 3 6
## attr(,"match.length")
## [1] 4 3 1
##
## [[2]]
## [1] -1
## attr(,"match.length")
## [1] -1
##
## [[3]]
## [1] 3 3 6
## attr(,"match.length")
## [1] 4 3 1
```

```
m <- aregexec("(lay)(sy)", x, max.distance = 2)
regmatches(x, m)
```

```
## [[1]]
## [1] "lazy" "la"   "zy"
##
## [[2]]
```

```
## character(0)
##
## [[3]]
## character(0)
```

```
## Cf. https://en.wikipedia.org/wiki/Levenshtein_distance
adist("kitten", "sitting")
```

```
##      [,1]
## [1,]    3
```

```
## To see the transformation counts for the Levenshtein distance:
drop(attr(adist("kitten", "sitting", counts = TRUE), "counts"))
```

```
## ins del sub
##   1   0   2
```

```
## To see the transformation sequences:
attr(adist(c("kitten", "sitting"), counts = TRUE), "trafos")
```

```
##      [,1]      [,2]
## [1,] "MMMMMM"  "SMMMSMI"
## [2,] "SMMMSMD" "MMMMMMM"
```

```
## Cf. the examples for agrep:
adist("lasy", "1 lazy 2")
```

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

```
## For a "partial approximate match" (as used for agrep):
adist("lasy", "1 lazy 2", partial = TRUE)
```

```
##      [,1]
## [1,]    1
```

案例

```
help.search()
```

4.12 高级的替换

相比于 sprintf() 格式化输出字符串的方式替换，它的优势在于提示性，或者说代码的可读性

```
glue_data <- function(param, text) {
  idx <- gregexpr('\\{[^}]*\\}', text)[[1L]]
  keys <- substring(text, idx, idx + attr(idx, 'match.length') - 1L)
  for (key in keys) {
    text <- gsub(key, param[[gsub('[{}]', '', key)]], text, fixed = TRUE)
  }
  text
```

©黄天


```
}
cat(glue_data(
  param = list(table = 'flights', origin = 'JFK'),
  text = "
  select count(*) as n
  from {table}
  where origin = '{origin}'
  "
))
```

```
##
##   select count(*) as n
##   from flights
##   where origin = 'JFK'
##
```

4.13 高级的提取

从 text 中抽取给定模式 pattern 的字符串

```
str_extract <- function(text, pattern, ...) regmatches(text, regexpr(pattern, text, ...))
```

举个栗子，比如提取数字

```
shopping_list <- c("apples x4", "bag of flour", "bag of sugar", "milk x2")
stringr::str_extract(shopping_list, "\\d")
```

```
## [1] "4" NA  NA  "2"
```

```
# 注意二者的差别
str_extract(shopping_list, "\\d")
```

```
## [1] "4" "2"
```

提取所有符合匹配模式的字符串

```
str_extract_all <- function(text, pattern, ...) regmatches(text, gregexpr(pattern, text, ...))
```

举个栗子，提取其中的英文字母

```
str_extract_all(shopping_list, "[a-z]+")
```

```
## [[1]]
## [1] "apples" "x"
##
## [[2]]
## [1] "bag"   "of"    "flour"
##
## [[3]]
## [1] "bag"   "of"    "sugar"
```

©清华大学出版社

```
##
## [[4]]
## [1] "milk" "x"
```

```
stringr::str_extract_all(shopping_list, "[a-z]+")
```

```
## [[1]]
## [1] "apples" "x"
##
## [[2]]
## [1] "bag"   "of"    "flour"
##
## [[3]]
## [1] "bag"   "of"    "sugar"
##
## [[4]]
## [1] "milk" "x"
```

4.14 其它操作

4.14.1 strwrap

```
strwrap(x, width = 0.9 * getOption("width"), indent = 0,
        exdent = 0, prefix = "", simplify = TRUE, initial = prefix)
```

该函数把一个字符串当成一个段落的文字（不管字符串中是否有换行符），按照段落的格式（缩进和长度）和断字方式进行分行，每一行是结果中的一个字符串。

```
# 读取一段文本
x <- paste(readLines(file.path(R.home("doc"), "THANKS")), collapse = "\n")
## 将文本拆分为段落，且移除前三段
x <- unlist(strsplit(x, "\n[ \t\n]*\n"))[-(1:3)]
# 每一段换两行
x <- paste(x, collapse = "\n\n")
# 每一行的宽度设定为60个字符
writeLines(strwrap(x, width = 60))
```

```
## J. D. Beasley, David J. Best, Richard Brent, Kevin Buhr,
## Michael A. Covington, Bill Cleveland, Robert Cleveland,, G.
## W. Cran, C. G. Ding, Ulrich Drepper, Paul Eggert, J. O.
## Evans, David M. Gay, H. Frick, G. W. Hill, Richard H.
## Jones, Eric Grosse, Shelby Haberman, Bruno Haible, John
## Hartigan, Andrew Harvey, Trevor Hastie, Min Long Lam,
## George Marsaglia, K. J. Martin, Gordon Matzigkeit, C. R.
## Mckenzie, Jean McRae, Cyrus Mehta, Fionn Murtagh, John C.
## Nash, Finbarr O'Sullivan, R. E. Odeh, William Patefield,
```

©清华大学出版社

```
## Nitin Patel, Alan Richardson, D. E. Roberts, Patrick
## Royston, Russell Lenth, Ming-Jen Shyu, Richard C.
## Singleton, S. G. Springer, Supoj Sutanthavibul, Irma
## Terpenning, G. E. Thomas, Rob Tibshirani, Wai Wan Tsang,
## Berwin Turlach, Gary V. Vaughan, Michael Wichura, Jingbo
## Wang, M. A. Wong, and the Free Software Foundation (for
## autoconf code and utilities). See also files under
## src/extras.
##
## Many more, too numerous to mention here, have contributed
## by sending bug reports and suggesting various improvements.
##
## Simon Davies whilst at the University of Auckland wrote the
## original version of glm().
##
## Julian Harris and Wing Kwong (Tiki) Wan whilst at the
## University of Auckland assisted Ross Ihaka with the
## original Macintosh port.
##
## R was inspired by the S environment which has been
## principally developed by John Chambers, with substantial
## input from Douglas Bates, Rick Becker, Bill Cleveland,
## Trevor Hastie, Daryl Pregibon and Allan Wilks.
##
## A special debt is owed to John Chambers who has graciously
## contributed advice and encouragement in the early days of R
## and later became a member of the core team.
##
## Stefano Iacus (a former member of R Core) and Simon Urbanek
## developed the macOS port, including the R.app GUI,
## toolchains and packaging.
##
## The Windows port was developed by Guido Masarotto (for a
## while a member of R Core) and Brian Ripley, then Duncan
## Murdoch (a former member of R Core) and currently by Jeroen
## Ooms (base) and Uwe Ligges (packages).
##
## Tomas Kalibera's work has been sponsored by Jan Vitek and
## funded by his European Research Council grant "Evolving
## Language Ecosystems (ELE)".
##
## Computing support (including hardware, hosting and
## infrastructure) has been provided/funded by the R
## Foundation, employers of R-Core members (notably WU Wien,
## ETH Zurich, U Oxford and U Iowa) and by Northeastern
```

©清华大学出版社

```
## University and the University of Kent.
##
## Distributions of R contain the recommended packages, whose
## authors/contributors are listed in their DESCRIPTION files.
```

```r
# 每一段的段首缩进5个字符
writeLines(strwrap(x, width = 60, indent = 5))
```

```
##      J. D. Beasley, David J. Best, Richard Brent, Kevin
## Buhr, Michael A. Covington, Bill Cleveland, Robert
## Cleveland,, G. W. Cran, C. G. Ding, Ulrich Drepper, Paul
## Eggert, J. O. Evans, David M. Gay, H. Frick, G. W. Hill,
## Richard H. Jones, Eric Grosse, Shelby Haberman, Bruno
## Haible, John Hartigan, Andrew Harvey, Trevor Hastie, Min
## Long Lam, George Marsaglia, K. J. Martin, Gordon
## Matzigkeit, C. R. Mckenzie, Jean McRae, Cyrus Mehta, Fionn
## Murtagh, John C. Nash, Finbarr O'Sullivan, R. E. Odeh,
## William Patefield, Nitin Patel, Alan Richardson, D. E.
## Roberts, Patrick Royston, Russell Lenth, Ming-Jen Shyu,
## Richard C. Singleton, S. G. Springer, Supoj Sutanthavibul,
## Irma Terpenning, G. E. Thomas, Rob Tibshirani, Wai Wan
## Tsang, Berwin Turlach, Gary V. Vaughan, Michael Wichura,
## Jingbo Wang, M. A. Wong, and the Free Software Foundation
## (for autoconf code and utilities). See also files under
## src/extras.
##
##      Many more, too numerous to mention here, have
## contributed by sending bug reports and suggesting various
## improvements.
##
##      Simon Davies whilst at the University of Auckland
## wrote the original version of glm().
##
##      Julian Harris and Wing Kwong (Tiki) Wan whilst at the
## University of Auckland assisted Ross Ihaka with the
## original Macintosh port.
##
##      R was inspired by the S environment which has been
## principally developed by John Chambers, with substantial
## input from Douglas Bates, Rick Becker, Bill Cleveland,
## Trevor Hastie, Daryl Pregibon and Allan Wilks.
##
##      A special debt is owed to John Chambers who has
## graciously contributed advice and encouragement in the
## early days of R and later became a member of the core team.
##
```

©黄湘云

```
##      Stefano Iacus (a former member of R Core) and Simon
## Urbanek developed the macOS port, including the R.app GUI,
## toolchains and packaging.
##
##      The Windows port was developed by Guido Masarotto (for
## a while a member of R Core) and Brian Ripley, then Duncan
## Murdoch (a former member of R Core) and currently by Jeroen
## Ooms (base) and Uwe Ligges (packages).
##
##      Tomas Kalibera's work has been sponsored by Jan Vitek
## and funded by his European Research Council grant "Evolving
## Language Ecosystems (ELE)".
##
##      Computing support (including hardware, hosting and
## infrastructure) has been provided/funded by the R
## Foundation, employers of R-Core members (notably WU Wien,
## ETH Zurich, U Oxford and U Iowa) and by Northeastern
## University and the University of Kent.
##
##      Distributions of R contain the recommended packages,
## whose authors/contributors are listed in their DESCRIPTION
## files.
```

```r
# 除了段首，每一段的余下诸行都缩进5个字符
writeLines(strwrap(x, width = 60, exdent = 5))
```

```
## J. D. Beasley, David J. Best, Richard Brent, Kevin Buhr,
##      Michael A. Covington, Bill Cleveland, Robert
##      Cleveland,, G. W. Cran, C. G. Ding, Ulrich Drepper,
##      Paul Eggert, J. O. Evans, David M. Gay, H. Frick, G.
##      W. Hill, Richard H. Jones, Eric Grosse, Shelby
##      Haberman, Bruno Haible, John Hartigan, Andrew Harvey,
##      Trevor Hastie, Min Long Lam, George Marsaglia, K. J.
##      Martin, Gordon Matzigkeit, C. R. Mckenzie, Jean McRae,
##      Cyrus Mehta, Fionn Murtagh, John C. Nash, Finbarr
##      O'Sullivan, R. E. Odeh, William Patefield, Nitin
##      Patel, Alan Richardson, D. E. Roberts, Patrick
##      Royston, Russell Lenth, Ming-Jen Shyu, Richard C.
##      Singleton, S. G. Springer, Supoj Sutanthavibul, Irma
##      Terpenning, G. E. Thomas, Rob Tibshirani, Wai Wan
##      Tsang, Berwin Turlach, Gary V. Vaughan, Michael
##      Wichura, Jingbo Wang, M. A. Wong, and the Free
##      Software Foundation (for autoconf code and utilities).
##      See also files under src/extras.
##
## Many more, too numerous to mention here, have contributed
```

```
##      by sending bug reports and suggesting various
##      improvements.
##
## Simon Davies whilst at the University of Auckland wrote the
##      original version of glm().
##
## Julian Harris and Wing Kwong (Tiki) Wan whilst at the
##      University of Auckland assisted Ross Ihaka with the
##      original Macintosh port.
##
## R was inspired by the S environment which has been
##      principally developed by John Chambers, with
##      substantial input from Douglas Bates, Rick Becker,
##      Bill Cleveland, Trevor Hastie, Daryl Pregibon and
##      Allan Wilks.
##
## A special debt is owed to John Chambers who has graciously
##      contributed advice and encouragement in the early days
##      of R and later became a member of the core team.
##
## Stefano Iacus (a former member of R Core) and Simon Urbanek
##      developed the macOS port, including the R.app GUI,
##      toolchains and packaging.
##
## The Windows port was developed by Guido Masarotto (for a
##      while a member of R Core) and Brian Ripley, then
##      Duncan Murdoch (a former member of R Core) and
##      currently by Jeroen Ooms (base) and Uwe Ligges
##      (packages).
##
## Tomas Kalibera's work has been sponsored by Jan Vitek and
##      funded by his European Research Council grant
##      "Evolving Language Ecosystems (ELE)".
##
## Computing support (including hardware, hosting and
##      infrastructure) has been provided/funded by the R
##      Foundation, employers of R-Core members (notably WU
##      Wien, ETH Zurich, U Oxford and U Iowa) and by
##      Northeastern University and the University of Kent.
##
## Distributions of R contain the recommended packages, whose
##      authors/contributors are listed in their DESCRIPTION
##      files.
```

©清华大学出版社

```r
# 在输出的每一行前面添加前缀
writeLines(strwrap(x, prefix = "THANKS> "))
```

```
## THANKS> J. D. Beasley, David J. Best, Richard Brent, Kevin Buhr,
## THANKS> Michael A. Covington, Bill Cleveland, Robert Cleveland,, G. W.
## THANKS> Cran, C. G. Ding, Ulrich Drepper, Paul Eggert, J. O. Evans,
## THANKS> David M. Gay, H. Frick, G. W. Hill, Richard H. Jones, Eric
## THANKS> Grosse, Shelby Haberman, Bruno Haible, John Hartigan, Andrew
## THANKS> Harvey, Trevor Hastie, Min Long Lam, George Marsaglia, K. J.
## THANKS> Martin, Gordon Matzigkeit, C. R. Mckenzie, Jean McRae, Cyrus
## THANKS> Mehta, Fionn Murtagh, John C. Nash, Finbarr O'Sullivan, R. E.
## THANKS> Odeh, William Patefield, Nitin Patel, Alan Richardson, D. E.
## THANKS> Roberts, Patrick Royston, Russell Lenth, Ming-Jen Shyu, Richard
## THANKS> C. Singleton, S. G. Springer, Supoj Sutanthavibul, Irma
## THANKS> Terpenning, G. E. Thomas, Rob Tibshirani, Wai Wan Tsang, Berwin
## THANKS> Turlach, Gary V. Vaughan, Michael Wichura, Jingbo Wang, M. A.
## THANKS> Wong, and the Free Software Foundation (for autoconf code and
## THANKS> utilities). See also files under src/extras.
## THANKS>
## THANKS> Many more, too numerous to mention here, have contributed by
## THANKS> sending bug reports and suggesting various improvements.
## THANKS>
## THANKS> Simon Davies whilst at the University of Auckland wrote the
## THANKS> original version of glm().
## THANKS>
## THANKS> Julian Harris and Wing Kwong (Tiki) Wan whilst at the
## THANKS> University of Auckland assisted Ross Ihaka with the original
## THANKS> Macintosh port.
## THANKS>
## THANKS> R was inspired by the S environment which has been principally
## THANKS> developed by John Chambers, with substantial input from Douglas
## THANKS> Bates, Rick Becker, Bill Cleveland, Trevor Hastie, Daryl
## THANKS> Pregibon and Allan Wilks.
## THANKS>
## THANKS> A special debt is owed to John Chambers who has graciously
## THANKS> contributed advice and encouragement in the early days of R and
## THANKS> later became a member of the core team.
## THANKS>
## THANKS> Stefano Iacus (a former member of R Core) and Simon Urbanek
## THANKS> developed the macOS port, including the R.app GUI, toolchains
## THANKS> and packaging.
## THANKS>
## THANKS> The Windows port was developed by Guido Masarotto (for a while
## THANKS> a member of R Core) and Brian Ripley, then Duncan Murdoch (a
## THANKS> former member of R Core) and currently by Jeroen Ooms (base)
```

© 基础

```
## THANKS> and Uwe Ligges (packages).
## THANKS>
## THANKS> Tomas Kalibera's work has been sponsored by Jan Vitek and
## THANKS> funded by his European Research Council grant "Evolving
## THANKS> Language Ecosystems (ELE)".
## THANKS>
## THANKS> Computing support (including hardware, hosting and
## THANKS> infrastructure) has been provided/funded by the R Foundation,
## THANKS> employers of R-Core members (notably WU Wien, ETH Zurich, U
## THANKS> Oxford and U Iowa) and by Northeastern University and the
## THANKS> University of Kent.
## THANKS>
## THANKS> Distributions of R contain the recommended packages, whose
## THANKS> authors/contributors are listed in their DESCRIPTION files.
```

再举一个烧脑的例子

```
x <- paste(sapply(
  sample(10, 100, replace = TRUE), # 从1-10个数字中有放回的随机抽取100个数
  function(x) substring("aaaaaaaaaa", 1, x)
), collapse = " ")
sapply(
  10:40,
  function(m)
    c(target = m, actual = max(nchar(strwrap(x, m))))
)
```

```
##        [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## target   10   11   12   13   14   15   16   17   18    19    20    21    22
## actual   10   10   11   12   13   14   15   16   17    18    19    20    21
##        [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## target    23    24    25    26    27    28    29    30    31    32    33    34
## actual    22    23    24    25    26    27    28    29    30    31    32    33
##        [,26] [,27] [,28] [,29] [,30] [,31]
## target    35    36    37    38    39    40
## actual    34    35    36    37    38    39
```

4.14.2 strtrim

```
strtrim(x, width)
```

strtrim 函数将字符串 x 修剪到特定的显示宽度，返回的字符串向量的长度等于字符串向量 x 的长度，如果 width 的参数值（它是一个整型向量）的长度小于 x 的，就循环补齐。

```
strtrim(c("abcdef", "abcdef", "abcdef"), c(1, 5, 10))
```

```
## [1] "a"      "abcde"  "abcdef"
```

© 黄湘云

4.14.3 strrep

```
strrep(x, times)
```

以给定的次数重复字符串向量中每个元素的个数，并连接字符串的各个副本

```
strrep("ABC", 2)
```

```
## [1] "ABCABC"
```

```
strrep(c("A", "B", "C"), 1 : 3)
```

```
## [1] "A"   "BB"  "CCC"
```

```
# 创建一个字符串向量，指定每个元素中空格的数量
strrep(" ", 1 : 5)
```

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

4.14.4 trimws

```
trimws(x, which = c("both", "left", "right"), whitespace = "[ \t\r\n]")
```

trimws 函数用于移除字符串中的空格，这种空格可以来自制表符、回车符和换行符，位置可以位于字符串的开头或者结尾，which 参数指定空格的大致位置。举例如下

```
x <- "  Some text. "
x
```

```
## [1] "  Some text. "
```

```
trimws(x)
```

```
## [1] "Some text."
```

```
trimws(x, "l")
```

```
## [1] "Some text. "
```

```
trimws(x, "r")
```

```
## [1] "  Some text."
```

```
shopping_list <- c("apples x4", "bag of flour", "bag of sugar", "milk x2")

stringr::str_replace(string = shopping_list, pattern = "\\d", replacement = "aa")
```

```
## [1] "apples xaa"   "bag of flour" "bag of sugar" "milk xaa"
```

```
# https://github.com/hadley/stringb/issues/5
# x is vector
str_replace <- function(x, pattern, fun, ...) {
  loc <- gregexpr(pattern, text = x, perl = TRUE)
  matches <- regmatches(x, loc)
```

© 黄湘云


```
  out <- lapply(matches, fun, ...)

  regmatches(x, loc) <- out
  x
}


loc <- gregexpr(pattern = "\\d", text = shopping_list, perl = TRUE)

matches = regmatches(x = shopping_list, loc)

matches

out <- lapply(matches, transform, "aa")

regmatches(x = shopping_list, loc) <- out


shopping_list


str_replace(shopping_list, pattern = "\\\\d", replace = "aa")
```

4.14.5 tolower

tolower 和 toupper 是一对，将大写转小写，小写转大写

```
simpleCap <- function(x) {
  x <- tolower(x)
  s <- strsplit(x, " ")[[1]]
  paste(toupper(substring(s, 1, 1)), substring(s, 2),
    sep = "", collapse = " "
  )
}
# 参考文献条目里需要将每个英文单词的首字母大写
simpleCap(x = "THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS")
```

```
## [1] "The Use Of Multiple Measurements In Taxonomic Problems"
```

4.15 字符串加密

字符串编码加密，**openssl** 包提供了 sha1 函数 [2]

[2] 参考刘思㊣的两篇博文：利用 R 函数生成差异化密码 和 在 R 中各种码的转换

©黄湘云


```
library(openssl)
encode_mobile <- function(phone_number) paste("*", paste(toupper(sha1(sha1(charToRaw(paste(phone_numb
# 随意模拟两个手机号
mobile_vec <- c("18601013453", "13811674545")
sapply(mobile_vec, encode_mobile)
```

```
##                                 18601013453
## "*B1D46D1D62C7280137F0E14249EE500865247B7B"
##                                 13811674545
## "*0554DA6E403491F58F1567DF2EDEB19186B77173"
```

4.16 处理性能

当你对一个很长的字符串进行大量的正则表达式匹配的时候，你需要考虑性能问题了，这时候该考虑启用合适的选项，一般来讲，PCRE 比默认的正则表达式引擎快，`fixed=TRUE` 可以继续加快匹配速度，特别是当每个模式只匹配少量次数时。

连接字符串，`paste/c/bfile/bracket` 函数性能比较 https://wch.github.io/string_builder/index.html

R 内置的默认正则表达式匹配方式是基于 PCRE 的匹配，`options` 控制 PCRE 默认的三个选项 `PCRE_limit_recursion=NA` 、`PCRE_study=10` 和 `PCRE_use_JIT=TRUE`，当前系统环境下 PCRE 的支持情况

```
pcre_config()
```

```
##              UTF-8 Unicode properties                 JIT               stack
##                               TRUE                TRUE                TRUE               FALSE
```

查看 R 环境的 PCRE 配置

```
sapply(c("PCRE_limit_recursion", "PCRE_study", "PCRE_use_JIT"), getOption)
```

```
## PCRE_limit_recursion           PCRE_study         PCRE_use_JIT
##                   NA                FALSE                 TRUE
```

4.17 网络爬虫

用 R 语言写爬虫 curl、httr、xml2、XML 和 rvest 解析网页[3]

```
# 查看 libcurl 库的版本
libcurlVersion()
```

```
## [1] "7.68.0"
## attr(,"ssl_version")
## [1] "OpenSSL/1.1.1f"
## attr(,"libssh_version")
## [1] "libssh/0.9.3/openssl/zlib"
```

[3]Jeroen Ooms 已经确认 RCurl 早已经不再维护，取代它的是 curl/httr，不要使用不再维护的 R 包 https://frie.codes/curl-vs-rcurl/

谢益辉 ©

```
## attr(,"protocols")
##  [1] "dict"   "file"   "ftp"    "ftps"   "gopher" "http"   "https"  "imap"
##  [9] "imaps"  "ldap"   "ldaps"  "pop3"   "pop3s"  "rtmp"   "rtsp"   "scp"
## [17] "sftp"   "smb"    "smbs"   "smtp"   "smtps"  "telnet" "tftp"
```

于主编利用 tidyRSS 包抓取解析博客站点的订阅信息，并将此设置为定时任务，创建自动更新内容的博客聚合网站 Daily R

抓取地震台信息

一个爬网页的练习：看看 R 邮件列表中最热门的讨论是什么

4.18 文本挖掘

How did Axios rectangle Trump's PDF schedule? A try with R 使用 pdftools 和 magick 处理表格，这两个 R 包分别依赖 Poppler C++ 和 ImageMagick++，在 Ubuntu 上安装 pdftools 和 magick 包

```
sudo apt-get install libpoppler-cpp-dev libmagick++-dev
```

```
install.packages(c("pdftools", "magick"))
```

除了 pdftools 包外，PDF 文档中表格抽取工具还有 tabulizer。扫描版 PDF 文档需要 OCR 识别技术支持的 tesseract 包

4.19 运行环境

```
xfun::session_info()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Locale:
##   LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##   LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##   LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##   LC_PAPER=en_US.UTF-8       LC_NAME=C
##   LC_ADDRESS=C               LC_TELEPHONE=C
##   LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## Package version:
##   askpass_1.1      assertthat_0.2.1 base64enc_0.1.3  bookdown_0.25
##   bslib_0.3.1      cli_3.2.0        compiler_4.1.3   cpp11_0.4.2
##   crayon_1.5.1     curl_4.3.2       DBI_1.1.2        digest_0.6.29
##   dplyr_1.0.8      ellipsis_0.3.2   evaluate_0.15    fansi_1.0.3
##   fastmap_1.1.0    fs_1.5.2         generics_0.1.2   glue_1.6.2
```

© 黄湘云


```
##   graphics_4.1.3   grDevices_4.1.3  highr_0.9        htmltools_0.5.2
##   jquerylib_0.1.4  jsonlite_1.8.0   knitr_1.38       lifecycle_1.0.1
##   magrittr_2.0.3   methods_4.1.3    openssl_2.0.0    pillar_1.7.0
##   pkgconfig_2.0.3  purrr_0.3.4      R6_2.5.1         rappdirs_0.3.3
##   rlang_1.0.2      rmarkdown_2.13   sass_0.4.1       stats_4.1.3
##   stringi_1.7.6    stringr_1.4.0    sys_3.4          sysfonts_0.8.8
##   tibble_3.1.6     tidyr_1.2.0      tidyselect_1.1.2 tinytex_0.38
##   tools_4.1.3      utf8_1.2.2       utils_4.1.3      vctrs_0.4.0
##   xfun_0.30        yaml_2.3.5
```

© 王沛勇

第五章　正则表达式

> Douglas Bates: If you really want to be cautious you could use an octal representation like `sep="\\007"` to get a character that is very unlikely to occur in a factor level.
>
> Ed L. Cashin: I definitely want to be cautious. Instead of the bell character I think I'll use the field separator character, `"\\034"`, just because this is the first time I've been able to use it for it's intended purpose! ;)
>
> Douglas Bates: Yes, but with `"\\034"` you don't get to make obscure James Bond references :-)
>
> — Douglas Bates and Ed L. Cashin R-help (April 2004)

维基百科关于 正则表达式的描述，学习正则表达式

```
# 毒鸡汤用来做文本分析
# https://github.com/egotong/nows/blob/master/soul.sql
```

R 内置的三种匹配模式

1. `fixed = TRUE`: 字面意思匹配 exact matching.
2. `perl = TRUE`: 使用 Perl 正则表达式.
3. `fixed = FALSE, perl = FALSE`: 使用 POSIX 1003.2 extended 正则表达式 (默认设置).

不要拘泥于一种解决方案，比如清理数据中正则表达式有 Base R 提供的一套，stringr 又一套，提高效率的工具 RStudio 插件 regexplain 和辅助创建正则表达式 RVerbalExpressions 包。

有几个名词需要单独拎出来解释的

- literal character strings 字面字符串
- metacharacters 元字符
- extended regular expressions 在下文中约定翻译为默认正则表达式
- character class 字符集 `[abc]`
- Perl-like regular expressions Perl 风格的正则表达式

以下所述，都不考虑函数中参数 `perl=TRUE` 的情况，R 语言中提供了扩展的（默认的）和 Perl 风格的两套正则表达式。作为入门，我们这里只关注前者，启用 Perl 正则表达式只需在函数如 `grep` 中将选项 `perl = TRUE` 即可，并将后者统一命名为 Perl 正则表达式[1]。

正则表达式 (**regular expression**，简称 regexp)，函数 `regexpr` 和 `gregexpr` 的名称就好理解了，在控制台输入 `?regex` 查看 R 支持的正则表达式，这个文档看上百八十回也不过分。R 内支持正则表达式的函数有 `grep`、`grepl`、`sub`、`gsub`、`regexpr`、`gregexpr` 、`regexec` 和 `strsplit`。函数 `apropos`，`browseEnv`，

[1]推荐的学习正则表达式的路径可以见统计之都论坛 https://d.cosx.org/d/420410

© 王建湘

help.search，list.files 和 ls 是通过函数 grep 来使用正则表达式的，它们全都使用 extended regular expressions

```
grep(pattern, x, ignore.case = FALSE, perl = FALSE, value = FALSE,
     fixed = FALSE, useBytes = FALSE, invert = FALSE)
```

匹配模式 pattern 的内容可以用函数 cat 打印出来，注意反斜杠进入 R 字符串中时，需要用两个，反斜杠 \ 本身是转义符，否则会报错。

```
cat("\\") # \ 反斜杠是转义字符
```

```
## \
```

```
cat("\\.")
```

```
## \.
```

```
cat("\\\n") # 注意 \n 表示换行
```

```
## \
```

5.1 字符常量

单引号 ' 双引号 " 和反引号 ‘ 三种类型的引用 (quotes) 是 R 语法的一部分[2]，此外反斜杠 \ 用来转义下面的字符

表 5.1: 字符常量表

字符常量	含义
\n	换行 newline
\r	回车 carriage return
\t	制表符 tab
\b	退格 backspace
\a	警报（铃）alert (bell)
\f	换页 form feed
\v	垂直制表符 vertical tab
\\	反斜杠 backslash \
\'	单引号 ASCII apostrophe '
\"	双引号 ASCII quotation mark "
\`	反引号或沉音符 ASCII grave accent (backtick) ‘
\nnn	八进制 character with given octal code (1, 2 or 3 digits)
\xnn	十六进制 character with given hex code (1 or 2 hex digits)
\unnnn	Unicode character with given code (1–4 hex digits)
\Unnnnnnnn	Unicode character with given code (1–8 hex digits)

[2] https://stat.ethz.ch/R-manual/R-devel/library/base/html/Quotes.html

© 黄海广

5.2 软件环境

R 内置的正则表达式实现是基于 PCRE ICU TRE iconv 等第三方库，搞清楚自己使用的版本信息是重要的，一些字符集的解释与区域环境有关，如 [:alnum:] 和 [:alpha:] 等，所以获取当前的区域设置也很重要

```
# find a suitable coding for the current locale
localeToCharset(locale = Sys.getlocale("LC_CTYPE"))
```

```
## [1] "UTF-8"     "ISO8859-1"
```

```
# 软件版本信息
extSoftVersion()
```

```
##                                             zlib
##                                          "1.2.11"
##                                            bzlib
##                              "1.0.8, 13-Jul-2019"
##                                               xz
##                                           "5.2.4"
##                                             PCRE
##                                "10.39 2021-10-29"
##                                              ICU
##                                           "66.1"
##                                              TRE
##                          "TRE 0.8.0 R_fixes (BSD)"
##                                            iconv
##                                      "glibc 2.31"
##                                         readline
##                                            "8.0"
##                                             BLAS
## "/usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0"
```

```
# 区域及其编码信息
l10n_info()
```

```
## $MBCS
## [1] TRUE
##
## $`UTF-8`
## [1] TRUE
##
## $`Latin-1`
## [1] FALSE
##
## $codeset
## [1] "UTF-8"
```

©黄湘云


```r
# 表示数字、货币的细节
Sys.localeconv()
```

```
##     decimal_point     thousands_sep          grouping   int_curr_symbol
##               "."                ""                ""            "USD "
##   currency_symbol mon_decimal_point mon_thousands_sep      mon_grouping
##               "$"               "."               ","        "\003\003"
##     positive_sign     negative_sign   int_frac_digits       frac_digits
##                ""               "-"               "2"               "2"
##     p_cs_precedes    p_sep_by_space     n_cs_precedes    n_sep_by_space
##               "1"               "0"               "1"               "0"
##       p_sign_posn       n_sign_posn
##               "1"               "1"
```

```r
# PCRE 启用的配置选项
pcre_config()
```

```
##              UTF-8 Unicode properties                JIT              stack
##               TRUE               TRUE               TRUE              FALSE
```

```r
# 比较全的字符信息
stringi::stri_info()
```

```
## $Unicode.version
## [1] "13.0"
##
## $ICU.version
## [1] "66.1"
##
## $Locale
## $Locale$Language
## [1] "en"
##
## $Locale$Country
## [1] "US"
##
## $Locale$Variant
## [1] ""
##
## $Locale$Name
## [1] "en_US"
##
##
## $Charset.internal
## [1] "UTF-8"  "UTF-16"
##
## $Charset.native
```

```
## $Charset.native$Name.friendly
## [1] "UTF-8"
##
## $Charset.native$Name.ICU
## [1] "UTF-8"
##
## $Charset.native$Name.UTR22
## [1] NA
##
## $Charset.native$Name.IBM
## [1] "ibm-1208"
##
## $Charset.native$Name.WINDOWS
## [1] "windows-65001"
##
## $Charset.native$Name.JAVA
## [1] "UTF-8"
##
## $Charset.native$Name.IANA
## [1] "UTF-8"
##
## $Charset.native$Name.MIME
## [1] "UTF-8"
##
## $Charset.native$ASCII.subset
## [1] TRUE
##
## $Charset.native$Unicode.1to1
## [1] NA
##
## $Charset.native$CharSize.8bit
## [1] FALSE
##
## $Charset.native$CharSize.min
## [1] 1
##
## $Charset.native$CharSize.max
## [1] 3
##
##
## $ICU.system
## [1] TRUE
##
## $ICU.UTF8
## [1] TRUE
```

© 基湘石

需要临时改变区域环境设置，配合特殊的画图和文本输出要求。

```
# 获取当前默认的区域设置
Sys.getlocale()
foo <- Sys.getlocale()
# 恢复默认的区域设置
Sys.setlocale("LC_ALL", locale = foo)
```

5.3 基本概念

正则表达式的构造方式类似算术表达式，通过各种操作组合子（更小的）表达式，整个表达式匹配一个或多个字符[3]。大多数字符，包括所有的字母和数字，是匹配自身的正则表达式。元字符 `. \ | ( ) [ { ^ $ * + ?` 需要转义才能表达其自身的含义，转义的方式是在元字符前面添加反斜杠，如要表达点号 `.` 需要使用 `\.`。要注意，它们是否有特殊意义取决于所在的内容。

一个字符集 (character class) 是用一对中括号 `[]` 括起来的字符列表，用来匹配列表中的任意单个字符，除非列表中的第一个字符是 `^`，它用来匹配不在这个列表中的字符。`[0123456789]` 用来匹配任意单个数字，`[^abc]` 用来匹配除字符 `a,b,c` 以外的任意字符。字符范围 (character ranges) 可以通过第一个和最后一个字符指定，中间用连字符 (hyphen) 连接，由于这种解释依赖于区域和具体实现，所以指定字符范围的使用方式最好避免。唯一可移植（便携，通用）的方式是作为字符集，在列表中列出所有的 ASCII 字母，`[ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz]`.

预定义的一些字符类，它们的解释依赖于当前的语言区域，下面是 POSIX locale 环境下的解释

- `[:alnum:]` 表示 `[:alpha:]` 和 `[:digit:]`，含义是 `[0-9A-Za-z]`，但是前者与区域和字符集无关，后者依赖于当前的区域设置和字符编码。要注意在这些字符集名 class names 中，中括号 `[]` 是符号名的一部分，是必须要包含的。在字符集中，大多数元字符失去它们特殊的意义。
- `[:alpha:]` 表示 `[:lower:]` 和 `[:upper:]`
- `[:blank:]` 表示空格 space 制表符 tab
- `[:cntrl:]` 表示控制符，在 ASCII 字符集里里，这些字符有八进制代码，从 000 到 037，和 177(DEL)。
- `[:digit:]` 表示数字 0,1,2,3,4,5,6,7,8,9
- `[:graph:]` 表示 `[:alnum:]` 和 `[:punct:]`.
- `[:lower:]` 表示当前区域下的小写字母
- `[:print:]` 表示可打印的字符 `[:alnum:]`, `[:punct:]` 和空格.
- `[:punct:]` 表示标点字符

  `! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` { | } ~``
- `[:space:]` 表示空格字符：水平制表符 tab，换行符 newline，垂直制表符 vertical tab，换页符 form feed，回车符 carriage return，空格符 space
- `[:xdigit:]` 表示 16 进制数字 0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f.

[3] `useBytes = TRUE` 表示把字符看作字节。字符、字节和比特的关系是，一个字节 byte 八个比特 bit，一个英文字符 character 用一个字节表示，而一个中、日、韩文字符需要两个字节表示

© 黄湘云

要包含字面的] 就把它放在列表的开头，类似地，要包含字面 ^，除了开头可以放在任意位置。要包含字面 - 把它放在开头或者结尾。只有 ^ - \ ] 在字符集内是有特殊的含义

点号 . 匹配任意单个字符，\w 匹配一个词 word 字符 (是 [[:alnum:]_] 的同义词，一个扩展)，而 \W 是 \w 取反，意味着 [^[:alnum:]_]。\d, \s, \D 和 \S 表示数字和空格类和它们的取反

脱字符 caret ^ 和美元符号 $ 是元字符，分别匹配一行的开头和结尾。符号 \< 和 \> 分别匹配一个词的开头和结尾的空字符串。\b 匹配词边缘的空字符串，\B 匹配不在词边缘的空字符串。词 word 的解释依赖于区域和实现。

5.4 字符串匹配

默认的匹配方式是贪婪的，会使用尽可能多的匹配次数，这个可以变为最小的匹配次数，通过在其之后添加 ?，一个正则表达式可能跟着重复量词，下面的限定符都是限定在它前面的正则表达式

表 5.2: 贪婪匹配限定符

符号	描述
?	匹配至多 1 次
*	匹配 0 次或多次
+	匹配至少 1 次
{n}	匹配 n 次
{n,}	匹配至少 n 次
{n,m}	匹配至少 n 次，至多 m 次

5.5 级联表达式

> Regular expressions may be concatenated; the resulting regular expression matches any string formed by concatenating the substrings that match the concatenated subexpressions.

正则表达式可以是级联 concatenation 的，是不是在讲一个正则表达式里面嵌套一个正则表达式?

两个正则表达式可以通过中缀符号 | 联合，用两个子表达式的任意一个去匹配字符串，例如 abba | cde 要么匹配字符串 abba 要么匹配字符串 cde，要注意在字符集内，即 abba|cde，二选一的匹配不凑效，因为中缀符 | 有它的字面意思。

重复匹配 Repetition 的优先级高于级联，级联高于 | 。整个子表达式可以括号括起来覆盖这些优先级规则。

5.6 反向引用

反向引用 \N 这里 N 可取 1,2,...,9 匹配被之前第 N 个括起来的子表达式匹配的子字符串，例子见 COS 论坛 https://d.cosx.org/d/420570/5

© 黄湘云

5.7 命名捕捉

模式 (?:...) 包住的字符就是括号分组，但是不做反向查找。模式 (?<=...) 和 (?<!...) 都是反向查找，它们不允许跟限制符，在 ... 也不允许出现 \C。表 5.3 展示四个反向引用

表 5.3: 环顾四周查找

符号	描述
?=	正向肯定查找
?!	正向否定查找
?<=	反向肯定查找
?<!	反向否定查找

函数 regexpr 和 gregexpr 支持命名捕捉 (named capture). 如果一个组被命名了，如 (?<first>[A-Z][a-z]+) 那么，匹配的位置是按名字返回。

下面举个例子说明，从字符串向量 notables 中获得了三组匹配 name.rex 是一段正则表达式，描述的模式是人名

```
## named capture
notables <- c("  Ben Franklin and Jefferson Davis",
              "\tMillard Fillmore")
# name groups 'first' and 'last'
name.rex <- "(?<first>[[:upper:]][[:lower:]]+) (?<last>[[:upper:]][[:lower:]]+)"
parsed <- regexpr(name.rex, notables, perl = TRUE)
parsed
```

```
## [1] 3 2
## attr(,"match.length")
## [1] 12 16
## attr(,"index.type")
## [1] "chars"
## attr(,"useBytes")
## [1] TRUE
## attr(,"capture.start")
##      first last
## [1,]     3    7
## [2,]     2   10
## attr(,"capture.length")
##      first last
## [1,]     3    8
## [2,]     7    8
## attr(,"capture.names")
## [1] "first" "last"
```

notables 是一个长度为 2 的字符串向量，所以获得两组匹配，捕捉到匹配开始的位置 capture.start 和匹配的长度 capture.length 都是两组，按列来看，字符 B 出现在字符串 Ben Franklin and Jefferson

© 博硕

Davis 的第三个位置，匹配的长度 Ben 是三个字符，长度是 3，如图 5.1 所示，需要注意的是一定要设置 perl = TRUE 才能使用命名捕捉功能，函数 sub 不支持命名反向引用 Named backreferences

图 5.1: 命名捕捉

Atomic grouping 原子分组, possessive qualifiers 占有限定 and conditional 条件 and recursive 递归等模式超出介绍的范围，不在此处详述，感兴趣的读者可参考，此外，插播一条漫画 5.2

图 5.2: 正则表达式漫画

正则表达式的直观解释 https://github.com/gadenbuie/regexplain

5.8 表达式注释

The sequence (?# marks the start of a comment which continues up to the next closing parenthesis. Nested parentheses are not permitted. The characters that make up a comment play no part at all in the pattern matching.

If the extended option is set, an unescaped # character outside a character class introduces a comment that continues up to the next newline character in the pattern.

批量转换驼峰式命名

© 谢益辉


```
old_name <- list.files(".", pattern = "^[A-Z].*.Rmd$")
new_name <- gsub("rmd", "Rmd", tolower(old_name))
file.rename(from = old_name, to = new_name)
```

```
html_lines <- readLines("https://movie.douban.com/top250")
doc <- paste0(html_lines, collapse = "")

title_lines <- grep('class="title"', html_lines, value = T)
titles <- gsub(".*>(.*?)<.*", "\\1", title_lines, perl = T)

gsub(".*>(.*?)<.*", "\\1", '<span class="title">肖生克的救赎</span>', perl = T)
```

解析术之 XPath

```
library(xml2)
dom = read_html(doc)
title_nodes = xml_find_all(dom, './/span[@class="title"]')
xml_text(title_nodes)
```

解析术之 CSS Selector

```
library(rvest)
read_html(doc) %>%
html_nodes('.title') %>% # class="title"的标签
html_text()
```

© 黄湘云

第六章　数据操作

data.table 诞生于 2006 年 4 月 15 日（以在 CRAN 上发布的第一个版本时间为准），是基于 `data.frame` 的扩展和 Base R 的数据操作连贯一些，dplyr 诞生于 2014 年 1 月 29 日，号称数据操作的语法，其实二者套路一致，都是借用 SQL 语言的设计，实现方式不同罢了，前者主要依靠 C 语言完成底层数据操作，总代码量 1.29M，C 占 65.6%，后者主要依靠 C++ 语言完成底层数据操作，总代码量 1.2M，C++ 占 34.4%，上层的高级操作接口都是 R 语言。像这样的大神在写代码，码力应该差不多，编程语言会对数据操作的性能有比较大的影响，我想这也是为什么在很多场合下 data.table 霸榜！

关于 data.table 和 dplyr 的对比，参看爆栈网的帖子 https://stackoverflow.com/questions/21435339

> **提示**
>
> 学习 data.table 包最快的方式就是在 R 控制台运行 `example(data.table)` 并研究其输出。

data.table 大大加强了 Base R 提供的数据操作，poorman 提供最常用的数据操作，但是不依赖 dplyr，fst，arrow 和 feather 提供更加高效的数据读写性能。

collapse 提供一系列高级和快速的数据操作，支持 Base R、dplyr、tibble、data.table、plm 和 sf 数据框结构类型。关键的特点有：1. 高级的统计编程，提供一系列统计函数支持在向量、矩阵和数据框上做分组和带权计算。fastverse 提供丰富的数据操作和统计计算功能，意图打造一个 tidyverse 替代品。

更多参考材料见A data.table and dplyr tour，Big Data in Economics: Data cleaning and wrangling 和 DataCamp's data.table cheatsheet，关于采用 Base R 还是 tidyverse 做数据操作的 讨论，数据操作的动画展示参考 https://github.com/gadenbuie/tidyexplain。

什么是 Base R? Base R 指的是 R 语言/软件的核心组件，由 R Core Team 维护

```
Pkgs <- sapply(list.files(R.home("library")), function(x)
  packageDescription(pkg = x, fields = "Priority"))
names(Pkgs[Pkgs == "base" & !is.na(Pkgs)])
```

```
##  [1] "base"      "compiler"  "datasets"  "graphics"  "grDevices" "grid"
##  [7] "methods"   "parallel"  "splines"   "stats"     "stats4"    "tcltk"
## [13] "tools"     "utils"
```

```
names(Pkgs[Pkgs == "recommended" & !is.na(Pkgs)])
```

```
##  [1] "boot"       "class"      "cluster"    "codetools"  "foreign"
##  [6] "KernSmooth" "lattice"    "MASS"       "Matrix"     "mgcv"
## [11] "nlme"       "nnet"       "rpart"      "spatial"    "survival"
```

数据变形，分组统计聚合等，用以作为模型的输入，绘图的对象，操作的数据对象是数据框 (data.frame) 类型的，而且如果没有特别说明，文中出现的数据集都是 Base R 内置的，第三方 R 包或者来源于网上的

©黄湘云

图 6.1: Tidyverse 和 Base R 的关系

数据集都会加以说明。

```
# 给定一个/些 R 包名，返回该 R 包存放的位置
sapply(.libPaths(), function(pkg_path) {
  c("survival", "ggplot2") %in% .packages(T, lib.loc = pkg_path)
})
```

```
##      /home/runner/work/_temp/Library /opt/R/4.1.3/lib/R/library
## [1,]                            TRUE                       TRUE
## [2,]                            TRUE                      FALSE
```

6.1 查看数据

查看属性

```
str(iris)
```

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

查看部分数据集

```
head(iris, 5)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
```

内部资料 ©

```
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
```

```
tail(iris, 5)
```

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

查看文件前（后）5 行

```
head -n 5 test.csv
tail -n 5 test.csv
```

对象的类型，存储方式

```
class(iris)
```

```
## [1] "data.frame"
```

```
mode(iris)
```

```
## [1] "list"
```

```
typeof(iris)
```

```
## [1] "list"
```

查看对象在 R 环境中所占空间的大小

```
object.size(iris)
```

```
## 7256 bytes
```

```
object.size(letters)
```

```
## 1712 bytes
```

```
object.size(ls)
```

```
## 89880 bytes
```

```
format(object.size(library), units = "auto")
```

```
## [1] "1.8 Mb"
```

©黄湘云

6.2 提取子集

```
subset(x, subset, select, drop = FALSE, ...)
```

参数 subset 代表行操作，select 代表列操作，函数 subset 从数据框中提取部分数据

```
subset(iris, subset = Species == "virginica" & Sepal.Length > 7.5)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 106          7.6         3.0          6.6         2.1 virginica
## 118          7.7         3.8          6.7         2.2 virginica
## 119          7.7         2.6          6.9         2.3 virginica
## 123          7.7         2.8          6.7         2.0 virginica
## 132          7.9         3.8          6.4         2.0 virginica
## 136          7.7         3.0          6.1         2.3 virginica
```

```
# summary(iris$Sepal.Length)  mean(iris$Sepal.Length)
# 且的逻辑
# subset(iris, Species == "virginica" & Sepal.Length > 5.8)
subset(iris, Species == "virginica" &
  Sepal.Length == median(Sepal.Length))
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 102          5.8         2.7          5.1         1.9 virginica
## 115          5.8         2.8          5.1         2.4 virginica
## 143          5.8         2.7          5.1         1.9 virginica
```

```
# 在行的子集范围内
subset(iris, Species %in% c("virginica", "versicolor") &
  Sepal.Length == median(Sepal.Length))
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 68           5.8         2.7          4.1         1.0 versicolor
## 83           5.8         2.7          3.9         1.2 versicolor
## 93           5.8         2.6          4.0         1.2 versicolor
## 102          5.8         2.7          5.1         1.9  virginica
## 115          5.8         2.8          5.1         2.4  virginica
## 143          5.8         2.7          5.1         1.9  virginica
```

```
# 在列的子集内 先选中列
subset(iris, Sepal.Length == median(Sepal.Length),
  select = c("Sepal.Length", "Species")
)
```

```
##     Sepal.Length    Species
## 15           5.8     setosa
## 68           5.8 versicolor
## 83           5.8 versicolor
## 93           5.8 versicolor
```

内部资料 ©

```
## 102          5.8  virginica
## 115          5.8  virginica
## 143          5.8  virginica
```

高级操作：加入正则表达式筛选

```r
## sometimes requiring a logical 'subset' argument is a nuisance
nm <- rownames(state.x77)
start_with_M <- nm %in% grep("^M", nm, value = TRUE)
subset(state.x77, start_with_M, Illiteracy:Murder)
```

```
##               Illiteracy Life Exp Murder
## Maine                0.7    70.39    2.7
## Maryland             0.9    70.22    8.5
## Massachusetts        1.1    71.83    3.3
## Michigan             0.9    70.63   11.1
## Minnesota            0.6    72.96    2.3
## Mississippi          2.4    68.09   12.5
## Missouri             0.8    70.69    9.3
## Montana              0.6    70.56    5.0
```

```r
# 简化
subset(state.x77, subset = grepl("^M", rownames(state.x77)), select = Illiteracy:Murder)
```

```
##               Illiteracy Life Exp Murder
## Maine                0.7    70.39    2.7
## Maryland             0.9    70.22    8.5
## Massachusetts        1.1    71.83    3.3
## Michigan             0.9    70.63   11.1
## Minnesota            0.6    72.96    2.3
## Mississippi          2.4    68.09   12.5
## Missouri             0.8    70.69    9.3
## Montana              0.6    70.56    5.0
```

```r
# 继续简化
subset(state.x77, grepl("^M", rownames(state.x77)), Illiteracy:Murder)
```

```
##               Illiteracy Life Exp Murder
## Maine                0.7    70.39    2.7
## Maryland             0.9    70.22    8.5
## Massachusetts        1.1    71.83    3.3
## Michigan             0.9    70.63   11.1
## Minnesota            0.6    72.96    2.3
## Mississippi          2.4    68.09   12.5
## Missouri             0.8    70.69    9.3
## Montana              0.6    70.56    5.0
```

© 高等教育出版社

注意

警告：这是一个为了交互使用打造的便捷函数。对于编程，最好使用标准的子集函数，如 `[`，特别地，参数 `subset` 的非标准计算 (non-standard evaluation)[a] 可能带来意想不到的后果。

[a] Thomas Lumley (2003) Standard nonstandard evaluation rules. https://developer.r-project.org/nonstandard-eval.pdf

使用索引 `[`

```
iris[iris$Species == "virginica" & iris$Sepal.Length == 5.8, ]
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 102          5.8         2.7          5.1         1.9 virginica
## 115          5.8         2.8          5.1         2.4 virginica
## 143          5.8         2.7          5.1         1.9 virginica
```

```
iris[iris$Species == "virginica" &
  iris$Sepal.Length == median(iris$Sepal.Length), ]
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 102          5.8         2.7          5.1         1.9 virginica
## 115          5.8         2.8          5.1         2.4 virginica
## 143          5.8         2.7          5.1         1.9 virginica
```

```
iris[
  iris$Species == "virginica" &
    iris$Sepal.Length == median(iris$Sepal.Length),
  c("Sepal.Length", "Species")
]
```

```
##     Sepal.Length   Species
## 102          5.8 virginica
## 115          5.8 virginica
## 143          5.8 virginica
```

```
iris[iris$Species == "setosa" & iris$Sepal.Length > 5.5, grepl("Sepal", colnames(iris))]
```

```
##    Sepal.Length Sepal.Width
## 15          5.8         4.0
## 16          5.7         4.4
## 19          5.7         3.8
```

```
subset(iris,
  subset = Species == "setosa" & Sepal.Length > 5.5,
  select = grepl("Sepal", colnames(iris))
)
```

```
##    Sepal.Length Sepal.Width
## 15          5.8         4.0
## 16          5.7         4.4
## 19          5.7         3.8
```

选择操作是针对数据框的列（变量/特征/字段）

内部使用 ©

```
library(data.table)
mtcars$cars <- rownames(mtcars)
mtcars_df <- as.data.table(mtcars)
```

```
mtcars_df[, .(mpg, disp)] |> head()
```

```
##     mpg disp
## 1: 21.0  160
## 2: 21.0  160
## 3: 22.8  108
## 4: 21.4  258
## 5: 18.7  360
## 6: 18.1  225
```

dplyr 版

```
mtcars |>
  dplyr::select(mpg, disp) |>
  head()
```

```
##                    mpg disp
## Mazda RX4         21.0  160
## Mazda RX4 Wag     21.0  160
## Datsun 710        22.8  108
## Hornet 4 Drive    21.4  258
## Hornet Sportabout 18.7  360
## Valiant           18.1  225
```

6.3 数据重塑

重复测量数据的变形 Reshape Grouped Data，将宽格式 wide 的数据框变长格式 long 的，反之也行。reshape 还支持正则表达式

```
str(Indometh)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':  66 obs. of  3 variables:
##  $ Subject: Ord.factor w/ 6 levels "1"<"4"<"2"<"5"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ time   : num  0.25 0.5 0.75 1 1.25 2 3 4 5 6 ...
##  $ conc   : num  1.5 0.94 0.78 0.48 0.37 0.19 0.12 0.11 0.08 0.07 ...
##  - attr(*, "formula")=Class 'formula'  language conc ~ time | Subject
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Time since drug administration"
##   ..$ y: chr "Indomethacin concentration"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(hr)"
##   ..$ y: chr "(mcg/ml)"
```

©李东风


```
summary(Indometh)
```

```
##  Subject      time            conc
##  1:11    Min.   :0.250   Min.   :0.0500
##  4:11    1st Qu.:0.750   1st Qu.:0.1100
##  2:11    Median :2.000   Median :0.3400
##  5:11    Mean   :2.886   Mean   :0.5918
##  6:11    3rd Qu.:5.000   3rd Qu.:0.8325
##  3:11    Max.   :8.000   Max.   :2.7200
```

```
# 长的变宽
wide <- reshape(Indometh,
  v.names = "conc", idvar = "Subject",
  timevar = "time", direction = "wide"
)
wide[, 1:6]
```

```
##    Subject conc.0.25 conc.0.5 conc.0.75 conc.1 conc.1.25
## 1        1      1.50     0.94      0.78   0.48      0.37
## 12       2      2.03     1.63      0.71   0.70      0.64
## 23       3      2.72     1.49      1.16   0.80      0.80
## 34       4      1.85     1.39      1.02   0.89      0.59
## 45       5      2.05     1.04      0.81   0.39      0.30
....
```

```
# 宽的变长
reshape(wide, direction = "long")
```

```
##        Subject time conc
## 1.0.25       1 0.25 1.50
## 2.0.25       2 0.25 2.03
## 3.0.25       3 0.25 2.72
## 4.0.25       4 0.25 1.85
## 5.0.25       5 0.25 2.05
....
```

宽的格式变成长的格式 https://stackoverflow.com/questions/2185252 或者长的格式变成宽的格式 https://stackoverflow.com/questions/5890584/

```
set.seed(45)
dat <- data.frame(
    name = rep(c("Orange", "Apple"), each=4),
    numbers = rep(1:4, 2),
    value = rnorm(8))
dat
```

```
##     name numbers      value
## 1 Orange       1  0.3407997
## 2 Orange       2 -0.7033403
```

© 基础统计

```
## 3 Orange       3 -0.3795377
## 4 Orange       4 -0.7460474
## 5  Apple       1 -0.8981073
## 6  Apple       2 -0.3347941
## 7  Apple       3 -0.5013782
## 8  Apple       4 -0.1745357
```

```
reshape(dat, idvar = "name", timevar = "numbers", direction = "wide")
```

```
##     name    value.1    value.2    value.3    value.4
## 1 Orange  0.3407997 -0.7033403 -0.3795377 -0.7460474
## 5  Apple -0.8981073 -0.3347941 -0.5013782 -0.1745357
```

```
## times need not be numeric
df <- data.frame(id = rep(1:4, rep(2,4)),
                 visit = I(rep(c("Before","After"), 4)),
                 x = rnorm(4), y = runif(4))
df
```

```
##   id  visit          x          y
## 1  1 Before  1.8090374 0.89106978
## 2  1  After -0.2301050 0.06920426
## 3  2 Before -1.1304182 0.94623103
## 4  2  After  0.2159889 0.74850150
## 5  3 Before  1.8090374 0.89106978
## 6  3  After -0.2301050 0.06920426
## 7  4 Before -1.1304182 0.94623103
## 8  4  After  0.2159889 0.74850150
```

```
reshape(df, timevar = "visit", idvar = "id", direction = "wide")
```

```
##   id  x.Before  y.Before    x.After    y.After
## 1  1  1.809037 0.8910698 -0.2301050 0.06920426
## 3  2 -1.130418 0.9462310  0.2159889 0.74850150
## 5  3  1.809037 0.8910698 -0.2301050 0.06920426
## 7  4 -1.130418 0.9462310  0.2159889 0.74850150
```

```
## warns that y is really varying
reshape(df, timevar = "visit", idvar = "id", direction = "wide", v.names = "x")
```

```
## Warning in reshapeWide(data, idvar = idvar, timevar = timevar, varying =
## varying, : some constant variables (y) are really varying
```

```
##   id         y  x.Before    x.After
## 1  1 0.8910698  1.809037 -0.2301050
## 3  2 0.9462310 -1.130418  0.2159889
## 5  3 0.8910698  1.809037 -0.2301050
## 7  4 0.9462310 -1.130418  0.2159889
```

更加复杂的例子，gambia 数据集，重塑的效果是使得个体水平的长格式变为村庄水平的宽格式

© 基础统计


```
# data(gambia, package = "geoR")
# 在线下载数据集
gambia <- read.table(
  file =
    paste("http://www.leg.ufpr.br/lib/exe/fetch.php",
      "pessoais:paulojus:mbgbook:datasets:gambia.txt",
      sep = "/"
    ), header = TRUE
)
head(gambia)
# Building a "village-level" data frame
ind <- paste("x", gambia[, 1], "y", gambia[, 2], sep = "")
village <- gambia[!duplicated(ind), c(1:2, 7:8)]
village$prev <- as.vector(tapply(gambia$pos, ind, mean))
head(village)
```

6.4 数据转换

transform 对数据框中的某些列做计算，取对数，将计算的结果单存一列加到数据框中

```
transform(iris[1:6, ], scale.sl = (max(Sepal.Length) - Sepal.Length) / (max(Sepal.Length) - min(Sepal
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species scale.sl
## 1          5.1         3.5          1.4         0.2  setosa    0.375
## 2          4.9         3.0          1.4         0.2  setosa    0.625
## 3          4.7         3.2          1.3         0.2  setosa    0.875
## 4          4.6         3.1          1.5         0.2  setosa    1.000
## 5          5.0         3.6          1.4         0.2  setosa    0.500
## 6          5.4         3.9          1.7         0.4  setosa    0.000
```

验证一下 `scale.sl` 变量的第一个值

```
(max(iris$Sepal.Length) - 5.1) / (max(iris$Sepal.Length) - min(iris$Sepal.Length))
```

```
## [1] 0.7777778
```

注意

Warning: This is a convenience function intended for use interactively. For programming it is better to use the standard subsetting arithmetic functions, and in particular the non-standard evaluation of argument `transform` can have unanticipated consequences.

6.5 按列排序

在数据框内，根据 (order) 某一列或几列对行进行排序 (sort)，根据鸢尾花 (iris) 的类别 (Species) 对萼片 (sepal) 的长度进行排序，其余的列随之变化

© 基础

```
# 先对花瓣的宽度排序，再对花瓣的长度排序
head(iris[order(iris$Species, iris$Petal.Width, iris$Petal.Length), ])
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 14          4.3         3.0          1.1         0.1  setosa
## 13          4.8         3.0          1.4         0.1  setosa
## 38          4.9         3.6          1.4         0.1  setosa
## 10          4.9         3.1          1.5         0.1  setosa
## 33          5.2         4.1          1.5         0.1  setosa
## 23          4.6         3.6          1.0         0.2  setosa
```

sort/ordered 排序，默认是升序

```
dd <- data.frame(
  b = factor(c("Hi", "Med", "Hi", "Low"),
    levels = c("Low", "Med", "Hi"), ordered = TRUE
  ),
  x = c("A", "D", "A", "C"), y = c(8, 3, 9, 9),
  z = c(1, 1, 1, 2)
)
str(dd)
```

```
## 'data.frame':    4 obs. of  4 variables:
##  $ b: Ord.factor w/ 3 levels "Low"<"Med"<"Hi": 3 2 3 1
##  $ x: chr  "A" "D" "A" "C"
##  $ y: num  8 3 9 9
##  $ z: num  1 1 1 2
```

```
dd[order(-dd[,4], dd[,1]), ]
```

```
##     b x y z
## 4 Low C 9 2
## 2 Med D 3 1
## 1  Hi A 8 1
## 3  Hi A 9 1
```

根据变量 z

```
dd[order(dd$z, dd$b), ]
```

```
##     b x y z
## 2 Med D 3 1
## 1  Hi A 8 1
## 3  Hi A 9 1
## 4 Low C 9 2
```

©黄湘云

6.6 数据拆分

数据拆分通常是按找某一个分类变量分组，分完组就是计算，计算完就把结果按照原来的分组方式合并

```
## Notice that assignment form is not used since a variable is being added
g <- airquality$Month
l <- split(airquality, g) # 分组
l <- lapply(l, transform, Oz.Z = scale(Ozone)) # 计算：按月对 Ozone 标准化
aq2 <- unsplit(l, g) # 合并
head(aq2)
```

```
##   Ozone Solar.R Wind Temp Month Day       Oz.Z
## 1    41     190  7.4   67     5   1  0.7822293
## 2    36     118  8.0   72     5   2  0.5572518
## 3    12     149 12.6   74     5   3 -0.5226399
## 4    18     313 11.5   62     5   4 -0.2526670
## 5    NA      NA 14.3   56     5   5         NA
## 6    28      NA 14.9   66     5   6  0.1972879
```

tapply 自带分组的功能，按月份 Month 对 Ozone 中心标准化，其返回一个列表

```
with(airquality, tapply(Ozone, Month, scale))
```

```
## $`5`
##              [,1]
##  [1,]  0.78222929
##  [2,]  0.55725184
##  [3,] -0.52263993
##  [4,] -0.25266698
##  [5,]          NA
##  [6,]  0.19728792
##  [7,] -0.02768953
##  [8,] -0.20767149
....
```

上面的过程等价于

```
do.call("rbind", lapply(split(airquality, airquality$Month), transform, Oz.Z = scale(Ozone)))
```

```
##       Ozone Solar.R Wind Temp Month Day         Oz.Z
## 5.1      41     190  7.4   67     5   1  0.782229293
## 5.2      36     118  8.0   72     5   2  0.557251841
## 5.3      12     149 12.6   74     5   3 -0.522639926
## 5.4      18     313 11.5   62     5   4 -0.252666984
## 5.5      NA      NA 14.3   56     5   5           NA
## 5.6      28      NA 14.9   66     5   6  0.197287919
## 5.7      23     299  8.6   65     5   7 -0.027689532
## 5.8      19      99 13.8   59     5   8 -0.207671494
## 5.9       8      19 20.1   61     5   9 -0.702621887
```

© 统计学习与R语言

....

由于上面对 Ozone 正态标准化，所以标准化后的 `Oz.z` 再按月分组计算方差自然每个月都是 1，而均值都是 0。

```
with(aq2, tapply(Oz.Z, Month, sd, na.rm = TRUE))
```

```
## 5 6 7 8 9
## 1 1 1 1 1
```

```
with(aq2, tapply(Oz.Z, Month, mean, na.rm = TRUE))
```

```
##             5             6             7             8             9
## -4.240273e-17  1.052760e-16  5.841432e-17  5.898060e-17  2.571709e-17
```

> 循着这个思路，我们可以用 tapply 实现分组计算，上面函数 `sd` 和 `mean` 完全可以用自定义的更加复杂的函数替代

`cut` 函数可以将连续型变量划分为分类变量

```
set.seed(2019)
Z <- stats::rnorm(10)
cut(Z, breaks = -6:6)
```

```
##  [1] (0,1]   (-1,0]  (-2,-1] (0,1]   (-2,-1] (0,1]   (-1,0]  (0,1]   (-2,-1]
## [10] (-1,0]
## 12 Levels: (-6,-5] (-5,-4] (-4,-3] (-3,-2] (-2,-1] (-1,0] (0,1] (1,2] ... (5,6]
```

```
# labels = FALSE 返回每个数所落的区间位置
cut(Z, breaks = -6:6, labels = FALSE)
```

```
##  [1] 7 6 5 7 5 7 6 7 5 6
```

我们还可以指定参数 `dig.lab` 设置分组的精度，`ordered` 将分组变量看作是有序的，`breaks` 传递单个数时，表示分组数，而不是断点

```
cut(Z, breaks = 3, dig.lab = 4, ordered = TRUE)
```

```
##  [1] (0.06396,0.9186]  (-0.7881,0.06396] (-1.643,-0.7881]  (0.06396,0.9186]
##  [5] (-1.643,-0.7881]  (0.06396,0.9186]  (-0.7881,0.06396] (0.06396,0.9186]
##  [9] (-1.643,-0.7881]  (-0.7881,0.06396]
## Levels: (-1.643,-0.7881] < (-0.7881,0.06396] < (0.06396,0.9186]
```

此时，统计每组的频数，如图 6.2

```
# 条形图
plot(cut(Z, breaks = -6:6))
# 直方图
hist(Z, breaks = -6:6)
```

在指定分组数的情况下，我们还想获取分组的断点

```
labs <- levels(cut(Z, 3))
labs
```

```
## [1] "(-1.64,-0.788]" "(-0.788,0.064]" "(0.064,0.919]"
```

© 黄湘云

(a) 条形图 (b) 直方图

图 6.2: 连续型变量分组统计

用正则表达式抽取断点

```
cbind(
  lower = as.numeric(sub("\\((.+),.*", "\\1", labs)),
  upper = as.numeric(sub("[^,]*,([^]]*)\\]", "\\1", labs))
)
```

```
##       lower  upper
## [1,] -1.640 -0.788
## [2,] -0.788  0.064
## [3,]  0.064  0.919
```

更多相关函数可以参考 `findInterval` 和 `embed`

tabulate 和 table 有所不同，它表示排列，由 0 和 1 组成的一个长度为 5 数组，其中 1 有 3 个，则排列组合为

```
combn(5, 3, tabulate, nbins = 5)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    1    1    1    0    0    0     0
## [2,]    1    1    1    0    0    0    1    1    1     0
## [3,]    1    0    0    1    1    0    1    1    0     1
## [4,]    0    1    0    1    0    1    1    0    1     1
## [5,]    0    0    1    0    1    1    0    1    1     1
```

6.7 数据合并

merge 合并两个数据框

```
authors <- data.frame(
  ## I(*) : use character columns of names to get sensible sort order
  surname = I(c("Tukey", "Venables", "Tierney", "Ripley", "McNeil")),
  nationality = c("US", "Australia", "US", "UK", "Australia"),
```

©梁海波

```
  deceased = c("yes", rep("no", 4))
)
authorN <- within(authors, {
  name <- surname
  rm(surname)
})
books <- data.frame(
  name = I(c(
    "Tukey", "Venables", "Tierney",
    "Ripley", "Ripley", "McNeil", "R Core"
  )),
  title = c(
    "Exploratory Data Analysis",
    "Modern Applied Statistics ...",
    "LISP-STAT",
    "Spatial Statistics", "Stochastic Simulation",
    "Interactive Data Analysis",
    "An Introduction to R"
  ),
  other.author = c(
    NA, "Ripley", NA, NA, NA, NA,
    "Venables & Smith"
  )
)

authors
```

```
##    surname nationality deceased
## 1    Tukey          US      yes
## 2 Venables   Australia       no
## 3  Tierney          US       no
## 4   Ripley          UK       no
## 5   McNeil   Australia       no
```

```
authorN
```

```
##   nationality deceased     name
## 1          US      yes    Tukey
## 2   Australia       no Venables
## 3          US       no  Tierney
## 4          UK       no   Ripley
## 5   Australia       no   McNeil
```

```
books
```

```
##       name                         title     other.author
## 1    Tukey     Exploratory Data Analysis             <NA>
## 2 Venables Modern Applied Statistics ...           Ripley
```

```
## 3  Tierney                    LISP-STAT            <NA>
## 4   Ripley            Spatial Statistics            <NA>
## 5   Ripley         Stochastic Simulation            <NA>
## 6   McNeil     Interactive Data Analysis            <NA>
## 7   R Core          An Introduction to R Venables & Smith
```

默认找到同名的列，然后是同名的行合并，多余的没有匹配到的就丢掉

```
merge(authorN, books)
```

```
##       name nationality deceased                        title other.author
## 1   McNeil   Australia       no     Interactive Data Analysis         <NA>
## 2   Ripley          UK       no            Spatial Statistics         <NA>
## 3   Ripley          UK       no         Stochastic Simulation         <NA>
## 4  Tierney          US       no                     LISP-STAT         <NA>
## 5    Tukey          US      yes     Exploratory Data Analysis         <NA>
## 6 Venables   Australia       no Modern Applied Statistics ...       Ripley
```

还可以指定合并的列，先按照 surname 合并，留下 surname

```
merge(authors, books, by.x = "surname", by.y = "name")
```

```
##    surname nationality deceased                        title other.author
## 1   McNeil   Australia       no     Interactive Data Analysis         <NA>
## 2   Ripley          UK       no            Spatial Statistics         <NA>
## 3   Ripley          UK       no         Stochastic Simulation         <NA>
## 4  Tierney          US       no                     LISP-STAT         <NA>
## 5    Tukey          US      yes     Exploratory Data Analysis         <NA>
## 6 Venables   Australia       no Modern Applied Statistics ...       Ripley
```

留下的是 name

```
merge(books, authors, by.x = "name", by.y = "surname")
```

```
##       name                        title other.author nationality deceased
## 1   McNeil     Interactive Data Analysis         <NA>   Australia       no
## 2   Ripley            Spatial Statistics         <NA>          UK       no
## 3   Ripley         Stochastic Simulation         <NA>          UK       no
## 4  Tierney                     LISP-STAT         <NA>          US       no
## 5    Tukey     Exploratory Data Analysis         <NA>          US      yes
## 6 Venables Modern Applied Statistics ...       Ripley   Australia       no
```

为了比较清楚地观察几种合并的区别，这里提供对应的动画展示 https://github.com/gadenbuie/tidyexplain

(inner, outer, left, right, cross) join 共 5 种合并方式详情请看 https://stackoverflow.com/questions/1299871

cbind 和 rbind 分别是按列和行合并数据框

© 基础统计

6.8 数据去重

单个数值型向量去重，此时和 unique 函数作用一样

```
(x <- c(9:20, 1:5, 3:7, 0:8))
```

```
##  [1]  9 10 11 12 13 14 15 16 17 18 19 20  1  2  3  4  5  3  4  5  6  7  0  1  2
## [26]  3  4  5  6  7  8
```

```
## extract unique elements
x[!duplicated(x)]
```

```
##  [1]  9 10 11 12 13 14 15 16 17 18 19 20  1  2  3  4  5  6  7  0  8
```

```
unique(x)
```

```
##  [1]  9 10 11 12 13 14 15 16 17 18 19 20  1  2  3  4  5  6  7  0  8
```

数据框类型数据中，去除重复的行，这个重复可以是多个变量对应的向量

```
set.seed(2019)
df <- data.frame(
  x = sample(0:1, 10, replace = T),
  y = sample(0:1, 10, replace = T),
  z = 1:10
)
df
```

```
##    x y  z
## 1  0 0  1
## 2  0 1  2
## 3  1 0  3
## 4  0 0  4
## 5  0 1  5
## 6  0 1  6
## 7  1 0  7
## 8  0 1  8
## 9  0 0  9
## 10 1 0 10
```

```
df[!duplicated(df[, c("x", "y")]), ]
```

```
##   x y z
## 1 0 0 1
## 2 0 1 2
## 3 1 0 3
```

© 王惠勤

提示

去掉字段 cyl 和 gear 有重复的记录，data.table 方式

```
mtcars_df[!duplicated(mtcars_df, by = c("cyl", "gear"))][,.(mpg, cyl, gear)]
```

```
##     mpg cyl gear
## 1: 21.0   6    4
## 2: 22.8   4    4
## 3: 21.4   6    3
## 4: 18.7   8    3
## 5: 21.5   4    3
## 6: 26.0   4    5
## 7: 15.8   8    5
## 8: 19.7   6    5
```

dplyr 方式

```
mtcars |>
  dplyr::distinct(cyl, gear, .keep_all = TRUE) |>
  dplyr::select(mpg, cyl, gear)
```

```
##                    mpg cyl gear
## Mazda RX4         21.0   6    4
## Datsun 710        22.8   4    4
## Hornet 4 Drive    21.4   6    3
## Hornet Sportabout 18.7   8    3
## Toyota Corona     21.5   4    3
## Porsche 914-2     26.0   4    5
## Ford Pantera L    15.8   8    5
## Ferrari Dino      19.7   6    5
```

dplyr 的去重操作不需要拷贝一个新的数据对象 mtcars_df，并且可以以管道的方式将后续的选择操作连接起来，代码更加具有可读性。

```
mtcars_df[!duplicated(mtcars_df[, c("cyl", "gear")]), c("mpg","cyl","gear")]
```

```
##     mpg cyl gear
## 1: 21.0   6    4
## 2: 22.8   4    4
## 3: 21.4   6    3
## 4: 18.7   8    3
## 5: 21.5   4    3
## 6: 26.0   4    5
## 7: 15.8   8    5
## 8: 19.7   6    5
```

Base R 和 data.table 提供的 `duplicated()` 函数和 [函数一起实现去重的操作，选择操作放在 [实现，[其实是一个函数

```
x <- 2:4
x[1]
```

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

```
`[`(x, 1)
```

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

©黄天元

6.9 数据缺失

缺失数据操作

```
data("airquality")
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

对缺失值的处理默认是 na.action = na.omit

```
# Ozone 最高的那天
aggregate(data = airquality, Ozone ~ Month, max)
```

```
##   Month Ozone
## 1     5   115
## 2     6    71
## 3     7   135
## 4     8   168
## 5     9    96
```

```
# 每月 Ozone, Solar.R, Wind, Temp 平均值
aggregate(data = airquality, Ozone ~ Month, mean)
```

```
##   Month    Ozone
## 1     5 23.61538
## 2     6 29.44444
## 3     7 59.11538
## 4     8 59.96154
## 5     9 31.44828
```

缺失值处理

```
library(DataExplorer)
plot_missing(airquality)
```

查看包含缺失的记录，不完整的记录

```
airquality[!complete.cases(airquality), ]
```

```
##     Ozone Solar.R Wind Temp Month Day
## 5      NA      NA 14.3   56     5   5
## 6      28      NA 14.9   66     5   6
## 10     NA     194  8.6   69     5  10
## 11      7      NA  6.9   74     5  11
```

© 黄湘云


```
## 25     NA      66 16.6   57     5  25
## 26     NA     266 14.9   58     5  26
## 27     NA      NA  8.0   57     5  27
## 32     NA     286  8.6   78     6   1
## 33     NA     287  9.7   74     6   2
## 34     NA     242 16.1   67     6   3
## 35     NA     186  9.2   84     6   4
## 36     NA     220  8.6   85     6   5
## 37     NA     264 14.3   79     6   6
## 39     NA     273  6.9   87     6   8
## 42     NA     259 10.9   93     6  11
## 43     NA     250  9.2   92     6  12
## 45     NA     332 13.8   80     6  14
## 46     NA     322 11.5   79     6  15
## 52     NA     150  6.3   77     6  21
## 53     NA      59  1.7   76     6  22
## 54     NA      91  4.6   76     6  23
## 55     NA     250  6.3   76     6  24
## 56     NA     135  8.0   75     6  25
## 57     NA     127  8.0   78     6  26
## 58     NA      47 10.3   73     6  27
## 59     NA      98 11.5   80     6  28
## 60     NA      31 14.9   77     6  29
## 61     NA     138  8.0   83     6  30
## 65     NA     101 10.9   84     7   4
## 72     NA     139  8.6   82     7  11
## 75     NA     291 14.9   91     7  14
## 83     NA     258  9.7   81     7  22
## 84     NA     295 11.5   82     7  23
## 96     78      NA  6.9   86     8   4
## 97     35      NA  7.4   85     8   5
## 98     66      NA  4.6   87     8   6
## 102    NA     222  8.6   92     8  10
## 103    NA     137 11.5   86     8  11
## 107    NA      64 11.5   79     8  15
## 115    NA     255 12.6   75     8  23
## 119    NA     153  5.7   88     8  27
## 150    NA     145 13.2   77     9  27
```

Ozone 和 Solar.R 同时包含缺失值的行

```
airquality[is.na(airquality$Ozone) & is.na(airquality$Solar.R), ]
```

```
##    Ozone Solar.R Wind Temp Month Day
## 5     NA      NA 14.3   56     5   5
## 27    NA      NA  8.0   57     5  27
```

© 非洲

6.10 数据聚合

分组求和 https://stackoverflow.com/questions/1660124

主要是分组统计

```
apropos("apply")
```

```
##  [1] "apply"      "dendrapply" "eapply"     "frollapply" "kernapply"
##  [6] "lapply"     "mapply"     "rapply"     "sapply"     "tapply"
## [11] "vapply"
```

```
# 分组求和 colSums colMeans max
unique(iris$Species)
```

```
## [1] setosa     versicolor virginica
## Levels: setosa versicolor virginica
```

```
# 分类求和
# colSums(iris[iris$Species == "setosa", -5])
# colSums(iris[iris$Species == "virginica", -5])
colSums(iris[iris$Species == "versicolor", -5])
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        296.8        138.5        213.0         66.3
```

```
# apply(iris[iris$Species == "setosa", -5], 2, sum)
# apply(iris[iris$Species == "setosa", -5], 2, mean)
# apply(iris[iris$Species == "setosa", -5], 2, min)
# apply(iris[iris$Species == "setosa", -5], 2, max)
apply(iris[iris$Species == "setosa", -5], 2, quantile)
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## 0%            4.3       2.300        1.000         0.1
## 25%           4.8       3.200        1.400         0.2
## 50%           5.0       3.400        1.500         0.2
## 75%           5.2       3.675        1.575         0.3
## 100%          5.8       4.400        1.900         0.6
```

aggregate: Compute Summary Statistics of Data Subsets

```
# 按分类变量 Species 分组求和
# aggregate(subset(iris, select = -Species), by = list(iris[, "Species"]), FUN = sum)
aggregate(iris[, -5], list(iris[, 5]), sum)
```

```
##      Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa        250.3       171.4         73.1        12.3
## 2 versicolor        296.8       138.5        213.0        66.3
## 3  virginica        329.4       148.7        277.6       101.3
```

```
# 先确定位置，假设有很多分类变量
ind <- which("Species" == colnames(iris))
```

© 王建国


```
# 分组统计
aggregate(iris[, -ind], list(iris[, ind]), sum)
```

```
##      Group.1 Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa        250.3       171.4         73.1        12.3
## 2 versicolor        296.8       138.5        213.0        66.3
## 3  virginica        329.4       148.7        277.6       101.3
```

按照 Species 划分的类别，分组计算，使用公式表示形式，右边一定是分类变量，否则会报错误或者警告，输出奇怪的结果，请读者尝试运行 `aggregate(Species ~ Sepal.Length, data = iris, mean)`。公式法表示分组计算，~ 左手边可以做加 + 减 - 乘 * 除 / 取余 %% 等数学运算。下面以数据集 iris 为例，只对 Sepal.Length 按 Species 分组计算

```
aggregate(Sepal.Length ~ Species, data = iris, mean)
```

```
##      Species Sepal.Length
## 1     setosa        5.006
## 2 versicolor        5.936
## 3  virginica        6.588
```

与上述分组统计结果一样的命令，在大数据集上，与 aggregate 相比，tapply 要快很多，by 是 tapply 的包裹，处理速度差不多。读者可以构造伪随机数据集验证。

```
# tapply(iris$Sepal.Length, list(iris$Species), mean)
with(iris, tapply(Sepal.Length, Species, mean))
```

```
##     setosa versicolor  virginica
##      5.006      5.936      6.588
```

```
by(iris$Sepal.Length, iris$Species, mean)
```

```
## iris$Species: setosa
## [1] 5.006
## ------------------------------------------------------------
## iris$Species: versicolor
## [1] 5.936
## ------------------------------------------------------------
## iris$Species: virginica
## [1] 6.588
```

对所有变量按 Species 分组计算

```
aggregate(. ~ Species, data = iris, mean)
```

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa        5.006       3.428        1.462       0.246
## 2 versicolor        5.936       2.770        4.260       1.326
## 3  virginica        6.588       2.974        5.552       2.026
```

对变量 Sepal.Length 和 Sepal.Width 求和后，按 Species 分组计算

©黄海广 讲义

```
aggregate(Sepal.Length + Sepal.Width ~ Species, data = iris, mean)
```

```
##      Species Sepal.Length + Sepal.Width
## 1     setosa                      8.434
## 2 versicolor                      8.706
## 3  virginica                      9.562
```

对多个分类变量做分组计算，在数据集 ChickWeight 中 Chick 和 Diet 都是数字编码的分类变量，其中 Chick 是有序的因子变量，Diet 是无序的因子变量，而 Time 是数值型的变量，表示小鸡出生的天数。

```
# 查看数据
str(ChickWeight)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':  578 obs. of  4 variables:
##  $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
##  $ Time  : num  0 2 4 6 8 10 12 14 16 18 ...
##  $ Chick : Ord.factor w/ 50 levels "18"<"16"<"15"<..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ Diet  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "formula")=Class 'formula'  language weight ~ Time | Chick
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "outer")=Class 'formula'  language ~Diet
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Time"
##   ..$ y: chr "Body weight"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(days)"
##   ..$ y: chr "(gm)"
```

查看数据集 ChickWeight 的前几行

```
head(ChickWeight)
```

```
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
....
```

```
str(ChickWeight)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':  578 obs. of  4 variables:
##  $ weight: num  42 51 59 64 76 93 106 125 149 171 ...
##  $ Time  : num  0 2 4 6 8 10 12 14 16 18 ...
##  $ Chick : Ord.factor w/ 50 levels "18"<"16"<"15"<..: 15 15 15 15 15 15 15 15 15 15 ...
##  $ Diet  : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "formula")=Class 'formula'  language weight ~ Time | Chick
```

©黄湘云

```
....
```

对于数据集 ChickWeight 中的有序变量 Chick，aggregate 会按照既定顺序返回分组计算的结果

```
aggregate(weight ~ Chick, data = ChickWeight, mean)
```

```
##    Chick    weight
## 1     18  37.00000
## 2     16  49.71429
## 3     15  60.12500
## 4     13  67.83333
## 5      9  81.16667
....
```

```
aggregate(weight ~ Diet, data = ChickWeight, mean)
```

```
##   Diet   weight
## 1    1 102.6455
## 2    2 122.6167
## 3    3 142.9500
## 4    4 135.2627
```

分类变量没有用数字编码，以 CO2 数据集为例，该数据集描述草植对二氧化碳的吸收情况，Plant 是具有 12 个水平的有序的因子变量，Type 表示植物的源头分别是魁北克 (Quebec) 和密西西比 (Mississippi)，Treatment 表示冷却 (chilled) 和不冷却 (nonchilled) 两种处理方式，conc 表示周围环境中二氧化碳的浓度，uptake 表示植物吸收二氧化碳的速率。

```
# 查看数据集
head(CO2)
```

```
##   Plant   Type  Treatment conc uptake
## 1   Qn1 Quebec nonchilled   95   16.0
## 2   Qn1 Quebec nonchilled  175   30.4
## 3   Qn1 Quebec nonchilled  250   34.8
## 4   Qn1 Quebec nonchilled  350   37.2
## 5   Qn1 Quebec nonchilled  500   35.3
## 6   Qn1 Quebec nonchilled  675   39.2
```

```
str(CO2)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':  84 obs. of  5 variables:
##  $ Plant    : Ord.factor w/ 12 levels "Qn1"<"Qn2"<"Qn3"<..: 1 1 1 1 1 1 1 2 2 2 ...
##  $ Type     : Factor w/ 2 levels "Quebec","Mississippi": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Treatment: Factor w/ 2 levels "nonchilled","chilled": 1 1 1 1 1 1 1 1 1 1 ...
##  $ conc     : num  95 175 250 350 500 675 1000 95 175 250 ...
##  $ uptake   : num  16 30.4 34.8 37.2 35.3 39.2 39.7 13.6 27.3 37.1 ...
##  - attr(*, "formula")=Class 'formula'  language uptake ~ conc | Plant
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "outer")=Class 'formula'  language ~Treatment * Type
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
```

© 基础统计

```
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Ambient carbon dioxide concentration"
##   ..$ y: chr "CO2 uptake rate"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(uL/L)"
##   ..$ y: chr "(umol/m^2 s)"
```

对单个变量分组统计

```
aggregate(uptake ~ Plant, data = CO2, mean)
```

```
##    Plant   uptake
## 1    Qn1 33.22857
## 2    Qn2 35.15714
## 3    Qn3 37.61429
## 4    Qc1 29.97143
## 5    Qc3 32.58571
## 6    Qc2 32.70000
## 7    Mn3 24.11429
## 8    Mn2 27.34286
## 9    Mn1 26.40000
## 10   Mc2 12.14286
## 11   Mc3 17.30000
## 12   Mc1 18.00000
```

```
aggregate(uptake ~ Type, data = CO2, mean)
```

```
##          Type   uptake
## 1      Quebec 33.54286
## 2 Mississippi 20.88333
```

```
aggregate(uptake ~ Treatment, data = CO2, mean)
```

```
##    Treatment   uptake
## 1 nonchilled 30.64286
## 2    chilled 23.78333
```

对多个变量分组统计，查看二氧化碳吸收速率 uptake 随类型 Type 和处理方式 Treatment

```
aggregate(uptake ~ Type + Treatment, data = CO2, mean)
```

```
##          Type  Treatment   uptake
## 1      Quebec nonchilled 35.33333
## 2 Mississippi nonchilled 25.95238
## 3      Quebec    chilled 31.75238
## 4 Mississippi    chilled 15.81429
```

```
tapply(CO2$uptake, list(CO2$Type, CO2$Treatment), mean)
```

```
##             nonchilled  chilled
## Quebec        35.33333 31.75238
## Mississippi   25.95238 15.81429
```

© 黄湘云


```
by(CO2$uptake, list(CO2$Type, CO2$Treatment), mean)
```

```
## : Quebec
## : nonchilled
## [1] 35.33333
## ------------------------------------------------------------
## : Mississippi
## : nonchilled
## [1] 25.95238
## ------------------------------------------------------------
## : Quebec
## : chilled
## [1] 31.75238
## ------------------------------------------------------------
## : Mississippi
## : chilled
## [1] 15.81429
```

在这个例子中 tapply 和 by 的输出结果的表示形式不一样，aggregate 返回一个 data.frame 数据框，tapply 返回一个表格 table，by 返回特殊的数据类型 by。

Function `by` is an object-oriented wrapper for `tapply` applied to data frames.

```
# 分组求和
# by(iris[, 1], INDICES = list(iris$Species), FUN = sum)
# by(iris[, 2], INDICES = list(iris$Species), FUN = sum)
by(iris[, 3], INDICES = list(iris$Species), FUN = sum)
```

```
## : setosa
## [1] 73.1
## ------------------------------------------------------------
## : versicolor
## [1] 213
## ------------------------------------------------------------
## : virginica
## [1] 277.6
```

```
by(iris[1:3], INDICES = list(iris$Species), FUN = sum)
```

```
## : setosa
## [1] 494.8
## ------------------------------------------------------------
## : versicolor
## [1] 648.3
## ------------------------------------------------------------
## : virginica
## [1] 755.7
```

©清华大学出版社

```r
by(iris[1:3], INDICES = list(iris$Species), FUN = summary)
```

```
## : setosa
##   Sepal.Length    Sepal.Width     Petal.Length
##  Min.   :4.300   Min.   :2.300   Min.   :1.000
##  1st Qu.:4.800   1st Qu.:3.200   1st Qu.:1.400
##  Median :5.000   Median :3.400   Median :1.500
##  Mean   :5.006   Mean   :3.428   Mean   :1.462
##  3rd Qu.:5.200   3rd Qu.:3.675   3rd Qu.:1.575
##  Max.   :5.800   Max.   :4.400   Max.   :1.900
## ------------------------------------------------------------
## : versicolor
##   Sepal.Length    Sepal.Width     Petal.Length
##  Min.   :4.900   Min.   :2.000   Min.   :3.00
##  1st Qu.:5.600   1st Qu.:2.525   1st Qu.:4.00
##  Median :5.900   Median :2.800   Median :4.35
##  Mean   :5.936   Mean   :2.770   Mean   :4.26
##  3rd Qu.:6.300   3rd Qu.:3.000   3rd Qu.:4.60
##  Max.   :7.000   Max.   :3.400   Max.   :5.10
## ------------------------------------------------------------
## : virginica
##   Sepal.Length    Sepal.Width     Petal.Length
##  Min.   :4.900   Min.   :2.200   Min.   :4.500
##  1st Qu.:6.225   1st Qu.:2.800   1st Qu.:5.100
##  Median :6.500   Median :3.000   Median :5.550
##  Mean   :6.588   Mean   :2.974   Mean   :5.552
##  3rd Qu.:6.900   3rd Qu.:3.175   3rd Qu.:5.875
##  Max.   :7.900   Max.   :3.800   Max.   :6.900
```

```r
by(iris, INDICES = list(iris$Species), FUN = summary)
```

```
## : setosa
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.300   Min.   :1.000   Min.   :0.100
##  1st Qu.:4.800   1st Qu.:3.200   1st Qu.:1.400   1st Qu.:0.200
##  Median :5.000   Median :3.400   Median :1.500   Median :0.200
##  Mean   :5.006   Mean   :3.428   Mean   :1.462   Mean   :0.246
##  3rd Qu.:5.200   3rd Qu.:3.675   3rd Qu.:1.575   3rd Qu.:0.300
##  Max.   :5.800   Max.   :4.400   Max.   :1.900   Max.   :0.600
##        Species
##  setosa    :50
##  versicolor: 0
##  virginica : 0
##
##
##
```

©黄海广


```
## ------------------------------------------------------------
## : versicolor
##   Sepal.Length    Sepal.Width     Petal.Length   Petal.Width          Species
##  Min.   :4.900   Min.   :2.000   Min.   :3.00   Min.   :1.000   setosa    : 0
##  1st Qu.:5.600   1st Qu.:2.525   1st Qu.:4.00   1st Qu.:1.200   versicolor:50
##  Median :5.900   Median :2.800   Median :4.35   Median :1.300   virginica : 0
##  Mean   :5.936   Mean   :2.770   Mean   :4.26   Mean   :1.326
##  3rd Qu.:6.300   3rd Qu.:3.000   3rd Qu.:4.60   3rd Qu.:1.500
##  Max.   :7.000   Max.   :3.400   Max.   :5.10   Max.   :1.800
## ------------------------------------------------------------
## : virginica
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.900   Min.   :2.200   Min.   :4.500   Min.   :1.400
##  1st Qu.:6.225   1st Qu.:2.800   1st Qu.:5.100   1st Qu.:1.800
##  Median :6.500   Median :3.000   Median :5.550   Median :2.000
##  Mean   :6.588   Mean   :2.974   Mean   :5.552   Mean   :2.026
##  3rd Qu.:6.900   3rd Qu.:3.175   3rd Qu.:5.875   3rd Qu.:2.300
##  Max.   :7.900   Max.   :3.800   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    : 0
##  versicolor: 0
##  virginica :50
##
##
##
```

Group Averages Over Level Combinations of Factors 分组平均

```
str(warpbreaks)
```

```
## 'data.frame':    54 obs. of  3 variables:
##  $ breaks : num  26 30 54 25 70 52 51 26 67 18 ...
##  $ wool   : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
##  $ tension: Factor w/ 3 levels "L","M","H": 1 1 1 1 1 1 1 1 1 2 ...
```

```
head(warpbreaks)
```

```
##   breaks wool tension
## 1     26    A       L
## 2     30    A       L
## 3     54    A       L
## 4     25    A       L
## 5     70    A       L
## 6     52    A       L
```

```
ave(warpbreaks$breaks, warpbreaks$wool)
```

```
##  [1] 31.03704 31.03704 31.03704 31.03704 31.03704 31.03704 31.03704 31.03704
##  [9] 31.03704 31.03704 31.03704 31.03704 31.03704 31.03704 31.03704 31.03704
```

© 黄湘云

```
## [17] 31.03704 31.03704 31.03704 31.03704 31.03704 31.03704 31.03704 31.03704
## [25] 31.03704 31.03704 31.03704 25.25926 25.25926 25.25926 25.25926 25.25926
## [33] 25.25926 25.25926 25.25926 25.25926 25.25926 25.25926 25.25926 25.25926
## [41] 25.25926 25.25926 25.25926 25.25926 25.25926 25.25926 25.25926 25.25926
## [49] 25.25926 25.25926 25.25926 25.25926 25.25926 25.25926
```

```
with(warpbreaks, ave(breaks, tension, FUN = function(x) mean(x, trim = 0.1)))
```

```
##  [1] 35.6875 35.6875 35.6875 35.6875 35.6875 35.6875 35.6875 35.6875 35.6875
## [10] 26.3125 26.3125 26.3125 26.3125 26.3125 26.3125 26.3125 26.3125 26.3125
## [19] 21.0625 21.0625 21.0625 21.0625 21.0625 21.0625 21.0625 21.0625 21.0625
## [28] 35.6875 35.6875 35.6875 35.6875 35.6875 35.6875 35.6875 35.6875 35.6875
## [37] 26.3125 26.3125 26.3125 26.3125 26.3125 26.3125 26.3125 26.3125 26.3125
## [46] 21.0625 21.0625 21.0625 21.0625 21.0625 21.0625 21.0625 21.0625 21.0625
```

```
# 分组求和
with(warpbreaks, ave(breaks, tension, FUN = function(x) sum(x)))
```

```
##  [1] 655 655 655 655 655 655 655 655 655 475 475 475 475 475 475 475 475 475 390
## [20] 390 390 390 390 390 390 390 390 655 655 655 655 655 655 655 655 655 475 475
## [39] 475 475 475 475 475 475 475 390 390 390 390 390 390 390 390 390
```

```
# 分组求和
with(iris, ave(Sepal.Length, Species, FUN = function(x) sum(x)))
```

```
##   [1] 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3
##  [13] 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3
##  [25] 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3
##  [37] 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3 250.3
##  [49] 250.3 250.3 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8
##  [61] 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8
##  [73] 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8
##  [85] 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8 296.8
##  [97] 296.8 296.8 296.8 296.8 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4
## [109] 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4
## [121] 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4
## [133] 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4 329.4
## [145] 329.4 329.4 329.4 329.4 329.4 329.4
```

6.11 表格统计

介绍操作表格的 table, addmargins, prop.table, xtabs, margin.table, ftabe 等函数

table 多个分类变量分组计数统计

- 介绍 warpbreaks 和 airquality 纽约空气质量监测数据集二维的数据框
- UCBAdmissions 1973 年加州大学伯克利分校的院系录取数据集 3 维的列联表
- Titanic 4 维的列联表数据泰坦尼克号幸存者数据集

© 王兆军


```
with(warpbreaks, table(wool, tension))
```

```
##     tension
## wool L M H
##    A 9 9 9
##    B 9 9 9
```

以 iris 数据集为例，table 的第一个参数是自己制造的第二个分类变量，原始分类变量是 Species

```
with(iris, table(Sepal.check = Sepal.Length > 7, Species))
```

```
##            Species
## Sepal.check setosa versicolor virginica
##       FALSE     50         50        38
##       TRUE       0          0        12
```

```
with(iris, table(Sepal.check = Sepal.Length > mean(Sepal.Length), Species))
```

```
##            Species
## Sepal.check setosa versicolor virginica
##       FALSE     50         24         6
##       TRUE       0         26        44
```

以 airquality 数据集为例，看看月份中臭氧含量比较高的几天

```
aiq.tab <- with(airquality, table(Oz.high = Ozone > 80, Month))
aiq.tab
```

```
##        Month
## Oz.high  5  6  7  8  9
##   FALSE 25  9 20 19 27
##   TRUE   1  0  6  7  2
```

对表格按行和列求和，即求表格的边际，查看总体情况

```
addmargins(aiq.tab, 1:2)
```

```
##        Month
## Oz.high   5   6   7   8   9 Sum
##   FALSE  25   9  20  19  27 100
##   TRUE    1   0   6   7   2  16
##   Sum    26   9  26  26  29 116
```

臭氧含量超 80 的天数在每个月的占比，addmargins 的第二个参数 1 表示对列求和

```
aiq.prop <- prop.table(aiq.tab, 2)
aiq.prop
```

```
##        Month
## Oz.high          5          6          7          8          9
##   FALSE 0.96153846 1.00000000 0.76923077 0.73076923 0.93103448
##   TRUE  0.03846154 0.00000000 0.23076923 0.26923077 0.06896552
```

©黄湘云

```
aiq.marprop <- addmargins(aiq.prop, 1)
aiq.marprop
```

```
##        Month
## Oz.high          5          6          7          8          9
##   FALSE 0.96153846 1.00000000 0.76923077 0.73076923 0.93103448
##   TRUE  0.03846154 0.00000000 0.23076923 0.26923077 0.06896552
##   Sum   1.00000000 1.00000000 1.00000000 1.00000000 1.00000000
```

转换成百分比，将小数四舍五入转化为百分数，保留两位小数点

```
round(100 * aiq.marprop, 2)
```

```
##        Month
## Oz.high      5      6      7      8      9
##   FALSE  96.15 100.00  76.92  73.08  93.10
##   TRUE    3.85   0.00  23.08  26.92   6.90
##   Sum   100.00 100.00 100.00 100.00 100.00
```

```
pairs(airquality, panel = panel.smooth, main = "airquality data")
```

以 UCBAdmissions 数据集为例，使用 xtabs 函数把数据组织成列联表，先查看数据的内容

```
UCBAdmissions
```

```
## , , Dept = A
##
##           Gender
## Admit      Male Female
##   Admitted  512     89
##   Rejected  313     19
....
```

```
UCBA2DF <- as.data.frame(UCBAdmissions)
UCBA2DF
```

```
##       Admit Gender Dept Freq
## 1  Admitted   Male    A  512
## 2  Rejected   Male    A  313
## 3  Admitted Female    A   89
## 4  Rejected Female    A   19
## 5  Admitted   Male    B  353
....
```

接着将 UCBA2DF 数据集转化为表格的形式

```
UCBA2DF.tab <- xtabs(Freq ~ Gender + Admit + Dept, data = UCBA2DF)
ftable(UCBA2DF.tab)
```

```
##                 Dept   A   B   C   D   E   F
## Gender Admit
## Male   Admitted      512 353 120 138  53  22
```

©黄湘云


```
##         Rejected      313 207 205 279 138 351
## Female Admitted        89  17 202 131  94  24
##         Rejected       19   8 391 244 299 317
```

将录取性别和院系进行对比

```
prop.table(margin.table(UCBA2DF.tab, c(1, 3)), 1)
```

```
##          Dept
## Gender             A          B          C          D          E          F
##   Male   0.30657748 0.20810108 0.12077295 0.15496098 0.07097733 0.13861018
##   Female 0.05885559 0.01362398 0.32316076 0.20435967 0.21416894 0.18583106
```

男生倾向于申请院系 A 和 B，女生倾向于申请院系 C 到 F，院系 A 和 B 是最容易录取的。

6.12 索引访问

which 与引用 [性能比较，在区间 $[0, 1]$ 上生成 10 万个服从均匀分布的随机数，随机抽取其中 $\frac{1}{4}$。

```
n <- 100000
x <- runif(n)
i <- logical(n)
i[sample(n, n / 4)] <- TRUE
microbenchmark::microbenchmark(x[i], x[which(i)], times = 1000)
```

TODO: 使用 `subset` 函数与 [比较

6.13 多维数组

多维数组的行列是怎么定义的 `?array` 轴的概念，画个图表示数组

```
array(1:27, c(3, 3, 3))
```

```
## , , 1
##
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
##
## , , 2
##
##      [,1] [,2] [,3]
## [1,]   10   13   16
## [2,]   11   14   17
## [3,]   12   15   18
##
```

© 黄湘云

```
## , , 3
##
##      [,1] [,2] [,3]
## [1,]   19   22   25
## [2,]   20   23   26
## [3,]   21   24   27
```

垂直于 Z 轴的平面去截三维立方体，3 代表 z 轴，得到三个截面（二维矩阵）

```
asplit(array(1:27, c(3, 3, 3)), 3)
```

```
## [[1]]
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
##
## [[2]]
##      [,1] [,2] [,3]
## [1,]   10   13   16
## [2,]   11   14   17
## [3,]   12   15   18
##
## [[3]]
##      [,1] [,2] [,3]
## [1,]   19   22   25
## [2,]   20   23   26
## [3,]   21   24   27
```

对每个二维矩阵按列求和

```
lapply(asplit(array(1:27, c(3, 3, 3)), 3), apply, 2, sum)
```

```
## [[1]]
## [1]  6 15 24
##
## [[2]]
## [1] 33 42 51
##
## [[3]]
## [1] 60 69 78
```

`asplit` 和 `lapply` 组合处理多维数组的计算问题

三维数组的矩阵运算 abind 包提供更多的数组操作，如合并，替换

数组操作 aperm 数组转置 Array Transposition

asplit 数组拆分其后接 lapply 或者 vapply

apply 数组计算

© 王建平

rray 包 https://github.com/r-lib/rray

6.14 其它操作

成对的数据操作有 `list` 与 `unlist`、`stack` 与 `unstack`、`class` 与 `unclass`、`attach` 与 `detach` 以及 `with` 和 `within`，它们在数据操作过程中有时会起到一定的补充作用。

6.14.1 列表属性

```
# 创建列表
list(...)
pairlist(...)
# 转化列表
as.list(x, ...)
## S3 method for class 'environment'
as.list(x, all.names = FALSE, sorted = FALSE, ...)
as.pairlist(x)
# 检查列表
is.list(x)
is.pairlist(x)

alist(...)
```

`list` 函数用来构造、转化和检查 R 列表对象。下面创建一个临时列表对象 tmp ，它包含两个元素 A 和 B，两个元素都是向量，前者是数值型，后者是字符型

```
(tmp <- list(A = c(1, 2, 3), B = c("a", "b")))
```

```
## $A
## [1] 1 2 3
##
## $B
## [1] "a" "b"
```

```
unlist(x, recursive = TRUE, use.names = TRUE)
```

`unlist` 函数将给定的列表对象 `x` 简化为原子向量 (atomic vector)，我们发现简化之后变成一个字符型向量

```
unlist(tmp)
```

```
##  A1  A2  A3  B1  B2
## "1" "2" "3" "a" "b"
```

```
unlist(tmp, use.names = FALSE)
```

```
## [1] "1" "2" "3" "a" "b"
```

unlist 的逆操作是 relist

©黄天宏

6.14.2 堆叠向量

```
stack(x, ...)
## Default S3 method:
stack(x, drop = FALSE, ...)
## S3 method for class 'data.frame'
stack(x, select, drop = FALSE, ...)

unstack(x, ...)
## Default S3 method:
unstack(x, form, ...)
## S3 method for class 'data.frame'
unstack(x, form, ...)
```

stack 与 unstack 将多个向量堆在一起组成一个向量

```
# 查看数据集 PlantGrowth
class(PlantGrowth)
```

```
## [1] "data.frame"
```

```
head(PlantGrowth)
```

```
##   weight group
## 1   4.17  ctrl
## 2   5.58  ctrl
## 3   5.18  ctrl
## 4   6.11  ctrl
## 5   4.50  ctrl
## 6   4.61  ctrl
```

```
# 检查默认的公式
formula(PlantGrowth)
```

```
## weight ~ group
```

```
# 根据公式解除堆叠
# 下面等价于 unstack(PlantGrowth, form = weight ~ group)
(pg <- unstack(PlantGrowth))
```

```
##    ctrl trt1 trt2
## 1  4.17 4.81 6.31
## 2  5.58 4.17 5.12
## 3  5.18 4.41 5.54
## 4  6.11 3.59 5.50
## 5  4.50 5.87 5.37
## 6  4.61 3.83 5.29
## 7  5.17 6.03 4.92
## 8  4.53 4.89 6.15
```

©黄湘云


```
## 9  5.33 4.32 5.80
## 10 5.14 4.69 5.26
```

现在再将变量 pg 堆叠起来，还可以指定要堆叠的列

```
stack(pg)
```

```
##    values  ind
## 1    4.17 ctrl
## 2    5.58 ctrl
## 3    5.18 ctrl
## 4    6.11 ctrl
## 5    4.50 ctrl
....
```

```
stack(pg, select = -ctrl)
```

```
##    values  ind
## 1    4.81 trt1
## 2    4.17 trt1
## 3    4.41 trt1
## 4    3.59 trt1
## 5    5.87 trt1
....
```

形式上和 reshape 有一些相似之处，数据框可以由长变宽或由宽变长

6.14.3 属性转化

```
class(x)
class(x) <- value
unclass(x)
inherits(x, what, which = FALSE)

oldClass(x)
oldClass(x) <- value
```

class 和 unclass 函数用来查看、设置类属性和取消类属性，常用于面向对象的编程设计中

```
class(iris)
```

```
## [1] "data.frame"
```

```
class(iris$Species)
```

```
## [1] "factor"
```

```
unclass(iris$Species)
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

© 草稿

```
##  [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3
## [112] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [149] 3 3
## attr(,"levels")
....
```

6.14.4 绑定环境

```
attach(what,
  pos = 2L, name = deparse(substitute(what), backtick = FALSE),
  warn.conflicts = TRUE
)
detach(name,
  pos = 2L, unload = FALSE, character.only = FALSE,
  force = FALSE
)
```

attach 和 detach 是否绑定数据框的列名，不推荐操作，推荐使用 with

```
attach(iris)
head(Species)
```

```
## [1] setosa setosa setosa setosa setosa setosa
## Levels: setosa versicolor virginica
```

```
detach(iris)
```

6.14.5 数据环境

```
with(data, expr, ...)
within(data, expr, ...)
## S3 method for class 'list'
within(data, expr, keepAttrs = TRUE, ...)
```

data 参数 data 用来构造表达式计算的环境。其默认值可以是一个环境，列表，数据框。在 within 函数中 data 参数只能是列表或数据框。

expr 参数 expr 包含要计算的表达式。在 within 中通常包含一个复合表达式，比如

```
{
  a <- somefun()
  b <- otherfun()
  ...
  rm(unused1, temp)
}
```

with 和 within 计算一组表达式，计算的环境是由数据构造的，后者可以修改原始的数据

© 基础统计


```
with(mtcars, mpg[cyl == 8 & disp > 350])
```

```
## [1] 18.7 14.3 10.4 10.4 14.7 19.2 15.8
```

和下面计算的结果一样，但是更加简洁漂亮

```
mtcars$mpg[mtcars$cyl == 8 & mtcars$disp > 350]
```

```
## [1] 18.7 14.3 10.4 10.4 14.7 19.2 15.8
```

within 函数可以修改原数据环境中的多个变量，比如删除、修改和添加等

```
# 原数据集 airquality
head(airquality)
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
aq <- within(airquality, {
  lOzone <- log(Ozone) # 取对数
  Month <- factor(month.abb[Month]) # 字符串型转因子型
  cTemp <- round((Temp - 32) * 5 / 9, 1) # 从华氏温度到摄氏温度转化
  S.cT <- Solar.R / cTemp # 使用新创建的变量
  rm(Day, Temp)
})
# 修改后的数据集
head(aq)
```

```
##   Ozone Solar.R Wind Month      S.cT cTemp   lOzone
## 1    41     190  7.4   May  9.793814  19.4 3.713572
## 2    36     118  8.0   May  5.315315  22.2 3.583519
## 3    12     149 12.6   May  6.394850  23.3 2.484907
## 4    18     313 11.5   May 18.742515  16.7 2.890372
## 5    NA      NA 14.3   May        NA  13.3       NA
## 6    28      NA 14.9   May        NA  18.9 3.332205
```

下面再举一个复杂的绘图例子，这个例子来自 boxplot 函数

```
with(ToothGrowth, {
  boxplot(len ~ dose,
    boxwex = 0.25, at = 1:3 - 0.2,
    subset = (supp == "VC"), col = "#4285f4",
    main = "Guinea Pigs' Tooth Growth",
    xlab = "Vitamin C dose mg",
    ylab = "tooth length", ylim = c(0, 35)
  )
```

© 基础

```
  boxplot(len ~ dose,
    add = TRUE, boxwex = 0.25, at = 1:3 + 0.2,
    subset = supp == "OJ", col = "#EA4335"
  )
  legend(2, 9, c("Ascorbic acid", "Orange juice"),
    fill = c("#4285f4", "#EA4335")
  )
})
```

将 boxplot 函数的 subset 参数单独提出来，调用数据子集选择函数 subset ，这里 with 中只包含一个表达式，所以也可以不用大括号

```
with(
  subset(ToothGrowth, supp == "VC"),
  boxplot(len ~ dose,
    boxwex = 0.25, at = 1:3 - 0.2,
    col = "#4285f4", main = "Guinea Pigs' Tooth Growth",
    xlab = "Vitamin C dose mg",
    ylab = "tooth length", ylim = c(0, 35)
  )
)
with(
  subset(ToothGrowth, supp == "OJ"),
  boxplot(len ~ dose,
    add = TRUE, boxwex = 0.25, at = 1:3 + 0.2,
    col = "#EA4335"
  )
```

© 王建硕


```
)
legend(2, 9, c("Ascorbic acid", "Orange juice"),
  fill = c("#4285f4", "#EA4335")
)
```

可以作为数据变换 `transform` 的一种替代，它也比较像 **dplyr** 包的 `mutate` 函数

```
within(mtcars[1:5,1:3],{
  disp.cc <- disp * 2.54^3
  disp.l <- disp.cc / 1e3
})
```

```
##                    mpg cyl disp   disp.l  disp.cc
## Mazda RX4         21.0   6  160 2.621930 2621.930
## Mazda RX4 Wag     21.0   6  160 2.621930 2621.930
## Datsun 710        22.8   4  108 1.769803 1769.803
## Hornet 4 Drive    21.4   6  258 4.227863 4227.863
## Hornet Sportabout 18.7   8  360 5.899343 5899.343
```

```
# 只能使用已有的列，刚生成的列不能用
# transform(
#   mtcars[1:5, 1:3],
#   disp.cc = disp * 2.54^3,
#   disp.l = disp.cc / 1e3
# )
transform(
  mtcars[1:5, 1:3],
  disp.cc = disp * 2.54^3
```

©黄湘云

```
)
```

```
##                    mpg cyl disp  disp.cc
## Mazda RX4         21.0   6  160 2621.930
## Mazda RX4 Wag     21.0   6  160 2621.930
## Datsun 710        22.8   4  108 1769.803
## Hornet 4 Drive    21.4   6  258 4227.863
## Hornet Sportabout 18.7   8  360 5899.343
```

transform 只能使用已有的列，变换中间生成的列不能用，所以相比于 transform 函数，within 显得更为灵活

6.15 apply 族

表 6.1: apply 函数

函数	输入	输出
apply()	矩阵、数据框	向量
lapply()	向量、列表	列表
sapply()	向量、列表	向量、矩阵
mapply()	多个向量	列表
tapply()	数据框、数组	向量
vapply()	列表	矩阵
eapply()	列表	列表
rapply()	嵌套列表	嵌套列表

除此之外，还有 dendrapply() 专门处理层次聚类或分类回归树型结构，而函数 kernapply() 用于时间序列的平滑处理

```
# Reproduce example 10.4.3 from Brockwell and Davis (1991) [@Brockwell_1991_Time]
spectrum(sunspot.year, kernel = kernel("daniell", c(11, 7, 3)), log = "no")
```

将函数应用到多个向量，返回一个列表，生成四组服从正态分布 $\mathcal{N}(\mu_i, \sigma_i)$ 的随机数，它们的均值和方差依次是 $\mu_i = \sigma_i = 1 \ldots 4$

```
means <- 1:4
sds <- 1:4
set.seed(2020)
samples <- mapply(rnorm,
  mean = means, sd = sds,
  MoreArgs = list(n = 10), SIMPLIFY = FALSE
)
samples
```

```
## [[1]]
##  [1]  1.37697212  1.30154837 -0.09802317 -0.13040590 -1.79653432  1.72057350
```

© 黄湘云

图 6.3: 太阳黑子的频谱

```
##  [7]  1.93912102  0.77062225  2.75913135  1.11736679
##
## [[2]]
##  [1]  0.2937544  3.8185184  4.3927459  1.2568322  1.7534795  5.6000862
##  [7]  5.4079918 -4.0775292 -2.5779499  2.1166070
##
## [[3]]
##  [1] 9.523096 6.294548 3.954661 2.780557 5.502806 3.596252 6.893524 5.810155
##  [9] 2.557700 3.331296
##
## [[4]]
##  [1]  0.7499813  1.0251913  8.3813803 13.7414948  5.5524739  5.1625107
##  [7]  2.8576069  4.3040589  1.7588056  5.7887535
```

我们借用图6.4来看一下 mapply 的效果，多组随机数生成非常有助于快速模拟。

```
par(mfrow = c(2, 2), mar = c(2, 2, 2, 2))
invisible(lapply(samples, function(x) {
  plot(x, pch = 16, col = "grey")
  abline(h = mean(x), lwd = 2, col = "darkorange")
}))
```

分别计算每个样本的平均值

```
sapply(samples, mean)
```

```
## [1] 0.8960372 1.7984536 5.0244596 4.9322257
```

©黄湘云

图 6.4: lapply 函数

分别计算每个样本的 1，2，3 分位点

```
lapply(samples, quantile, probs = 1:3 / 4)
```

```
## [[1]]
##       25%       50%       75%
## 0.1191382 1.2094576 1.6346732
##
## [[2]]
##       25%       50%       75%
## 0.5345238 1.9350433 4.2491890
##
## [[3]]
##      25%      50%      75%
## 3.397535 4.728734 6.173450
##
## [[4]]
##      25%      50%      75%
## 2.033506 4.733285 5.729684
```

仅用 sapply() 函数替换上面的 lapply()，我们可以得到一个矩阵，值得注意的是函数 quantile() 和 fivenum() 算出来的结果有一些差异

```
sapply(samples, quantile, probs = 1:3 / 4)
```

```
##          [,1]      [,2]     [,3]     [,4]
## 25% 0.1191382 0.5345238 3.397535 2.033506
```

© 黄湘云


```
## 50% 1.2094576 1.9350433 4.728734 4.733285
## 75% 1.6346732 4.2491890 6.173450 5.729684
```

```
vapply(samples, fivenum, c(Min. = 0, "1st Qu." = 0, Median = 0, "3rd Qu." = 0, Max. = 0))
```

```
##                [,1]       [,2]     [,3]       [,4]
## Min.    -1.79653432 -4.0775292 2.557700  0.7499813
## 1st Qu. -0.09802317  0.2937544 3.331296  1.7588056
## Median   1.20945758  1.9350433 4.728734  4.7332848
## 3rd Qu.  1.72057350  4.3927459 6.294548  5.7887535
## Max.     2.75913135  5.6000862 9.523096 13.7414948
```

vapply 和 sapply 类似，但是预先指定返回值类型，这样可以更加安全，有时也更快。

以数据集 presidents 为例，它是一个 ts 对象类型的时间序列数据，记录了 1945 年至 1974 年每个季度美国总统的支持率，这组数据中存在缺失值，以 NA 表示。支持率的变化趋势见图 6.5。

```
plot(presidents)
```

图 6.5: 1945-1974 美国总统的支持率

计算这 30 年每个季度的平均支持率

```
tapply(presidents, cycle(presidents), mean, na.rm = TRUE)
```

```
##        1        2        3        4
## 58.44828 56.43333 57.22222 53.07143
```

cycle() 函数计算序列中每个观察值在周期中的位置，presidents 的周期为 4，根据位置划分组，然后分组求平均，也可以化作如下计算步骤，虽然看起来复杂，但是数据操作的过程很清晰，不再看起来像是一个黑箱。

©黄湘云

tapply 函数来做分组求和

```
# 一个变量分组求和
tapply(warpbreaks$breaks, warpbreaks[, 3, drop = FALSE], sum)
```

```
## tension
##   L   M   H
## 655 475 390
```

```
# 两个变量分组计数
with(warpbreaks, table(wool, tension))
```

```
##     tension
## wool L M H
##    A 9 9 9
##    B 9 9 9
```

```
# 两个变量分组求和
dat <- aggregate(breaks ~ wool + tension, data = warpbreaks, sum) |>
  reshape(v.names = "breaks", idvar = "wool", timevar = "tension", direction = "wide", sep = "")

`colnames<-`(dat, gsub(pattern = "(breaks)", x = colnames(dat), replacement = ""))
```

```
##   wool   L   M   H
## 1    A 401 216 221
## 2    B 254 259 169
```

6.16 with 选项

注意 data.table 与 Base R 不同的地方

```
# https://github.com/Rdatatable/data.table/issues/4513
# https://d.cosx.org/d/421532-datatable-base-r
library(data.table)
iris <- as.data.table(iris)
```

```
iris[Species == "setosa" & Sepal.Length > 5.5, grepl("Sepal", colnames(iris))]
```

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

需要使用 `with = FALSE` 选项

```
iris[Species == "setosa" & Sepal.Length > 5.5,
  grepl("Sepal", colnames(iris)),
  with = FALSE
]
```

```
##    Sepal.Length Sepal.Width
## 1:          5.8         4.0
## 2:          5.7         4.4
```

© 非洲出版社


```
## 3:          5.7         3.8
```

不使用 with 选项，用函数 mget() 将字符串转变量

```
iris[
  Species == "setosa" & Sepal.Length > 5.5,
  mget(grep("Sepal", colnames(iris), value = TRUE))
]
```

```
##    Sepal.Length Sepal.Width
## 1:          5.8         4.0
## 2:          5.7         4.4
## 3:          5.7         3.8
```

更加 data.table 风格的方式见

```
iris[Species == "setosa" & Sepal.Length > 5.5, .SD, .SDcols = patterns("Sepal")]
```

```
##    Sepal.Length Sepal.Width
## 1:          5.8         4.0
## 2:          5.7         4.4
## 3:          5.7         3.8
```

with 还可以这样用，直接修改、添加一列

```
df <- expand.grid(x = 1:10, y = 1:10)
df$z <- with(df, x^2 + y^2)
df <- subset(df, z < 100)
df <- df[sample(nrow(df)), ]
head(df)
```

```
##    x y  z
## 7  7 1 50
## 8  8 1 65
## 65 5 7 74
## 14 4 2 20
## 37 7 4 65
## 5  5 1 26
```

```
library(ggplot2)
ggplot(df, aes(x, y, z = z)) +
  geom_contour()
```

6.17 分组聚合

```
methods("aggregate")
```

```
## [1] aggregate.data.frame aggregate.default*   aggregate.formula*
## [4] aggregate.ts
## see '?methods' for accessing help and source code
```

© 黄湘云

图 6.6: with 操作

```
args("aggregate.data.frame")
```

```
## function (x, by, FUN, ..., simplify = TRUE, drop = TRUE)
## NULL
```

```
args("aggregate.ts")
```

```
## function (x, nfrequency = 1, FUN = sum, ndeltat = 1, ts.eps = getOption("ts.eps"),
##     ...)
## NULL
```

```
# getAnywhere(aggregate.formula)
```

按 Species 分组，对 Sepal.Length 中大于平均值的数取平均

```
aggregate(Sepal.Length ~ Species, iris, function(x) mean(x[x > mean(x)]))
```

```
##      Species Sepal.Length
## 1     setosa     5.313636
## 2 versicolor     6.375000
## 3  virginica     7.159091
```

```
library(data.table)

dt <- data.table(
  x = rep(1:3, each = 3), y = rep(1:3, 3),
  z = rep(c("A", "B", "C"), 3), w = rep(c("a", "b", "a"), each = 3)
)
```

© 王建硕


```
dt[, .(x_sum = sum(x), y_sum = sum(y)), by = .(z, w)]
```

```
##    z w x_sum y_sum
## 1: A a     4     2
## 2: B a     4     4
## 3: C a     4     6
## 4: A b     2     1
## 5: B b     2     2
## 6: C b     2     3
```

```
dt[, .(x_sum = sum(x), y_sum = sum(y)), by = mget(c("z", "w"))]
```

```
##    z w x_sum y_sum
## 1: A a     4     2
## 2: B a     4     4
## 3: C a     4     6
## 4: A b     2     1
## 5: B b     2     2
## 6: C b     2     3
```

shiny 前端传递字符串向量，借助 `mget()` 函数根据选择的变量分组统计计算，只有一个变量可以使用 `get()` 传递变量给 data.table

```
library(shiny)

ui <- fluidPage(
  fluidRow(
    column(
      6,
      selectInput("input_vars",
        label = "变量", # 给筛选框取名
        choices = c(z = "z", w = "w"), # 待选的值
        selected = "z", # 指定默认值
        multiple = TRUE # 允许多选
      ),
      DT::dataTableOutput("output_table")
    )
  )
)

library(data.table)
library(magrittr)

dt <- data.table(
  x = rep(1:3, each = 3), y = rep(1:3, 3),
  z = rep(c("A", "B", "C"), 3), w = rep(c("a", "b", "a"), each = 3)
)
```

© 统计出版社

```
server <- function(input, output, session) {
  output$output_table <- DT::renderDataTable(
    {
      dt[, .(x_sum = sum(x), y_sum = sum(y)), by = mget(input$input_vars)] |>
        DT::datatable()
    },
    server = FALSE
  )
}

# 执行
shinyApp(ui = ui, server = server)
```

6.18 合并操作

```
dat1 <- data.frame(x = c(0, 0, 10, 10, 20, 20, 30, 30), y = c(1, 1, 2, 2, 3, 3, 4, 4))
dat2 <- data.frame(x = c(0, 10, 20, 30), z = c(3, 4, 5, 6))

data.frame(dat1, z = dat2$z[match(dat1$x, dat2$x)])
```

```
##    x y z
## 1  0 1 3
## 2  0 1 3
## 3 10 2 4
## 4 10 2 4
## 5 20 3 5
## 6 20 3 5
## 7 30 4 6
## 8 30 4 6
```

```
merge(dat1, dat2)
```

```
##    x y z
## 1  0 1 3
## 2  0 1 3
## 3 10 2 4
## 4 10 2 4
## 5 20 3 5
## 6 20 3 5
## 7 30 4 6
## 8 30 4 6
```

保留两个数据集中的所有行

©黄湘云

表 6.2: 不同生长环境下植物的干重

group	1	2	3	4	5	6	7	8	9	10
ctrl	4.17	5.58	5.18	6.11	4.50	4.61	5.17	4.53	5.33	5.14
trt1	4.81	4.17	4.41	3.59	5.87	3.83	6.03	4.89	4.32	4.69
trt2	6.31	5.12	5.54	5.50	5.37	5.29	4.92	6.15	5.80	5.26

6.19 长宽转换

```
args("reshape")
```

```
## function (data, varying = NULL, v.names = NULL, timevar = "time",
##     idvar = "id", ids = 1L:NROW(data), times = seq_along(varying[[1L]]),
##     drop = NULL, direction, new.row.names = NULL, sep = ".",
##     split = if (sep == "") {
##         list(regexp = "[A-Za-z][0-9]", include = TRUE)
##     } else {
##         list(regexp = sep, include = FALSE, fixed = TRUE)
##     })
## NULL
```

PlantGrowth 数据集的重塑操作也可以使用内置的函数 `reshape()` 实现

```
PlantGrowth$id <- rep(1:10, 3)
dat <- reshape(
  data = PlantGrowth, idvar = "group", v.names = "weight",
  timevar = "id", direction = "wide",
  sep = ""
)
knitr::kable(dat,
  caption = "不同生长环境下植物的干重", row.names = FALSE,
  col.names = gsub("(weight)", "", names(dat)),
  align = "c"
)
```

或者，我们也可以使用 **tidyr** 包提供的 `pivot_wider()` 函数

```
tidyr::pivot_wider(
  data = PlantGrowth, id_cols = id,
  names_from = group, values_from = weight
)
```

```
## # A tibble: 10 x 4
##       id  ctrl  trt1  trt2
##    <int> <dbl> <dbl> <dbl>
##  1     1  4.17  4.81  6.31
##  2     2  5.58  4.17  5.12
##  3     3  5.18  4.41  5.54
```

©黄湘云

```
##  4     4  6.11  3.59  5.5
##  5     5  4.5   5.87  5.37
##  6     6  4.61  3.83  5.29
##  7     7  5.17  6.03  4.92
##  8     8  4.53  4.89  6.15
##  9     9  5.33  4.32  5.8
## 10    10  5.14  4.69  5.26
```

或者，我们还可以使用 **data.table** 包提供的 `dcast()` 函数，用于将长格式的数据框重塑为宽格式的

```
PlantGrowth_DT <- as.data.table(PlantGrowth)
# 纵
dcast(PlantGrowth_DT, id ~ group, value.var = "weight")
```

```
##     id ctrl trt1 trt2
##  1:  1 4.17 4.81 6.31
##  2:  2 5.58 4.17 5.12
##  3:  3 5.18 4.41 5.54
##  4:  4 6.11 3.59 5.50
##  5:  5 4.50 5.87 5.37
##  6:  6 4.61 3.83 5.29
##  7:  7 5.17 6.03 4.92
##  8:  8 4.53 4.89 6.15
##  9:  9 5.33 4.32 5.80
## 10: 10 5.14 4.69 5.26
```

```
# 横
dcast(PlantGrowth_DT, group ~ id, value.var = "weight")
```

```
##    group    1    2    3    4    5    6    7    8    9   10
## 1:  ctrl 4.17 5.58 5.18 6.11 4.50 4.61 5.17 4.53 5.33 5.14
## 2:  trt1 4.81 4.17 4.41 3.59 5.87 3.83 6.03 4.89 4.32 4.69
## 3:  trt2 6.31 5.12 5.54 5.50 5.37 5.29 4.92 6.15 5.80 5.26
```

6.20 对符合条件的列操作

```
# 数值型变量的列的位置
which(sapply(iris, is.numeric))
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##            1            2            3            4
```

```
iris[, sapply(iris, is.numeric), with = F][Sepal.Length > 7.5]
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1:          7.6         3.0          6.6         2.1
## 2:          7.7         3.8          6.7         2.2
## 3:          7.7         2.6          6.9         2.3
```

© 王建硕


```
## 4:          7.7         2.8          6.7         2.0
## 5:          7.9         3.8          6.4         2.0
## 6:          7.7         3.0          6.1         2.3
```

```
class(iris)
```

```
## [1] "data.table" "data.frame"
```

用 Base R 提供的管道符号 |> 将 data.table 数据操作与 ggplot2 数据可视化连接起来

```
library(ggplot2)
iris |>
  subset(Species == "setosa" & Sepal.Length > 5.5) |>
  # 行过滤
  # subset(select = grep("Sepal", colnames(iris), value = TRUE)) |> # 列过滤
  subset(select = grepl("Sepal", colnames(iris))) |>
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) + # 绘图
  geom_point()
```

图 6.7: 管道连接数据操作和可视化

6.21 CASE WHEN 和 fcase

`CASE WHEN` 是 SQL 中的条件判断语句，**data.table** 中的函数 `fcase()` 可与之等价。值得注意的是，`fcase()` 需要 **data.table** 版本 1.13.0 及以上。

©黄湘云

```
dat <- data.table(
  weights = c(56.8, 57.2, 46.3, 38.5),
  gender = c("1", "0", "", "0")
)
# 1 表示男，0表示女，空表示未知
transform(dat, gender_cn = fcase(
  gender == "1", "男",
  gender == "0", "女",
  gender == "", "未知"
))
```

```
##    weights gender gender_cn
## 1:    56.8      1        男
## 2:    57.2      0        女
## 3:    46.3             未知
## 4:    38.5      0        女
```

6.22 数据操作实战

Toby Dylan Hocking 在 useR! 2020 大会上分享的幻灯片 https://github.com/tdhock/r-devel-emails

6.23 高频数据操作

以数据集 dat 为例介绍常用的数据操作

```
set.seed(2020)
dat <- data.frame(
  num_a = rep(seq(4), each = 4), num_b = rep(seq(4), times = 4),
  group_a = sample(x = letters[1:3], size = 16, replace = T),
  group_b = sample(x = LETTERS[1:3], size = 16, replace = T)
)
dat <- as.data.table(dat)
dat
```

```
##     num_a num_b group_a group_b
##  1:     1     1       c       B
##  2:     1     2       b       B
##  3:     1     3       a       B
##  4:     1     4       a       C
##  5:     2     1       b       B
##  6:     2     2       b       C
##  7:     2     3       a       B
##  8:     2     4       a       A
##  9:     3     1       b       C
```

© 王建平


```
## 10:     3     2       b       B
## 11:     3     3       b       B
## 12:     3     4       a       B
## 13:     4     1       b       C
## 14:     4     2       c       B
## 15:     4     3       b       C
## 16:     4     4       a       C
```

6.23.1 循环合并

- 问题来源 Faster version of Reduce(merge, list(DT1,DT2,DT3,...)) called mergelist (a la rbindlist)

6.23.2 分组计数

```
dat[, .(length(num_a)), by = .(group_a)] # dat[, .N , by = .(group_a)]
```

```
##    group_a V1
## 1:       c  2
## 2:       b  8
## 3:       a  6
```

```
dat[, .(length(num_a)), by = .(group_b)]
```

```
##    group_b V1
## 1:       B  9
## 2:       C  6
## 3:       A  1
```

```
dat[, .(length(num_a)), by = .(group_a, group_b)]
```

```
##    group_a group_b V1
## 1:       c       B  2
## 2:       b       B  4
## 3:       a       B  3
## 4:       a       C  2
## 5:       b       C  4
## 6:       a       A  1
```

6.23.3 分组抽样

以 group_a 为组别，a、b、c 分别有 6、8、2 条记录

```
# 无放回的抽样
dt_sample_1 <- dat[, .SD[sample(x = .N, size = 2, replace = FALSE)], by = group_a]
# 有放回的随机抽样
dt_sample_2 <- dat[, .SD[sample(x = .N, size = 3, replace = TRUE)], by = group_a]
```

©黄湘云

可能存在该组样本不平衡，有的组的样本量不足你想要的样本量。每个组无放回地抽取 4 个样本，如果该组样本量不足 4，则全部抽取全部样本量。

```
dat[, .SD[sample(x = .N, size = min(4, .N))], by = group_a]
```

```
##     group_a num_a num_b group_b
##  1:       c     1     1       B
##  2:       c     4     2       B
##  3:       b     3     2       B
##  4:       b     2     2       C
##  5:       b     2     1       B
##  6:       b     3     3       B
##  7:       a     1     3       B
##  8:       a     2     3       B
##  9:       a     2     4       A
## 10:       a     1     4       C
```

还可以按照指定的比例抽取样本量[1]

6.23.4 分组排序

data.table 包的分组排序问题 https://d.cosx.org/d/421650-datatable/3

```
dat[with(dat, order(-ave(num_a, group_a, FUN = max), -num_a)), ]
```

```
##     num_a num_b group_a group_b
##  1:     4     1       b       C
##  2:     4     2       c       B
##  3:     4     3       b       C
##  4:     4     4       a       C
##  5:     3     1       b       C
##  6:     3     2       b       B
##  7:     3     3       b       B
##  8:     3     4       a       B
##  9:     2     1       b       B
## 10:     2     2       b       C
## 11:     2     3       a       B
## 12:     2     4       a       A
## 13:     1     1       c       B
## 14:     1     2       b       B
## 15:     1     3       a       B
## 16:     1     4       a       C
```

```
# num_a 降序排列，然后对 group_a 升序排列
dat[with(dat, order(-num_a, group_a)), ]
```

```
##     num_a num_b group_a group_b
```

[1] https://stackoverflow.com/questions/18258690/take-randomly-sample-based-on-groups

© 王建平


```
##  1:     4     4       a       C
##  2:     4     1       b       C
##  3:     4     3       b       C
##  4:     4     2       c       B
##  5:     3     4       a       B
##  6:     3     1       b       C
##  7:     3     2       b       B
##  8:     3     3       b       B
##  9:     2     3       a       B
## 10:     2     4       a       A
## 11:     2     1       b       B
## 12:     2     2       b       C
## 13:     1     3       a       B
## 14:     1     4       a       C
## 15:     1     2       b       B
## 16:     1     1       c       B
```

```
# 简写
dat[order(-num_a, group_a)]
```

```
##     num_a num_b group_a group_b
##  1:     4     4       a       C
##  2:     4     1       b       C
##  3:     4     3       b       C
##  4:     4     2       c       B
##  5:     3     4       a       B
##  6:     3     1       b       C
##  7:     3     2       b       B
##  8:     3     3       b       B
##  9:     2     3       a       B
## 10:     2     4       a       A
## 11:     2     1       b       B
## 12:     2     2       b       C
## 13:     1     3       a       B
## 14:     1     4       a       C
## 15:     1     2       b       B
## 16:     1     1       c       B
```

`setorder()` 函数直接修改原始数据记录的排序

```
setorder(dat, -num_a, group_a)
```

参考多个列分组排序 [2]

[2] https://stackoverflow.com/questions/1296646/how-to-sort-a-dataframe-by-multiple-columns

©黄湘云

提示

如果数据集 dat 包含缺失值，考虑去掉缺失值

```
dat[, .(length(!is.na(num_a))), by = .(group_a)]
```

```
##    group_a V1
## 1:       c  2
## 2:       b  8
## 3:       a  6
```

如果数据集 dat 包含重复值，考虑去掉重复值

```
dat[, .(length(unique(num_a))), by = .(group_a)]
```

```
##    group_a V1
## 1:       c  2
## 2:       b  4
## 3:       a  4
```

按 Species 分组，对 Sepal.Length 降序排列，取 Top 3

```
iris <- as.data.table(iris)
iris[order(-Sepal.Length), .SD[1:3], by = "Species"]
```

```
##       Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1:  virginica          7.9         3.8          6.4         2.0
## 2:  virginica          7.7         3.8          6.7         2.2
## 3:  virginica          7.7         2.6          6.9         2.3
## 4: versicolor          7.0         3.2          4.7         1.4
## 5: versicolor          6.9         3.1          4.9         1.5
## 6: versicolor          6.8         2.8          4.8         1.4
## 7:     setosa          5.8         4.0          1.2         0.2
## 8:     setosa          5.7         4.4          1.5         0.4
## 9:     setosa          5.7         3.8          1.7         0.3
```

对 iris 各个列排序

```
dat <- head(iris)
ind <- do.call(what = "order", args = dat[, c(5, 1, 2, 3)])
dat[ind, ]
```

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

按 Species 分组，对 Sepal.Length 降序排列，取 Top 3

```
iris = as.data.table(iris)
iris[order(-Sepal.Length), .SD[1:3], by="Species"]
```

© 黄湘云

表 6.3: iris 数据集原顺序（左）和新顺序（右）

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
4.3	3.0	1.1	0.1	setosa
4.4	2.9	1.4	0.2	setosa
4.4	3.0	1.3	0.2	setosa
4.4	3.2	1.3	0.2	setosa
4.5	2.3	1.3	0.3	setosa
4.6	3.1	1.5	0.2	setosa

```
##       Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1:  virginica          7.9         3.8          6.4         2.0
## 2:  virginica          7.7         3.8          6.7         2.2
## 3:  virginica          7.7         2.6          6.9         2.3
## 4: versicolor          7.0         3.2          4.7         1.4
## 5: versicolor          6.9         3.1          4.9         1.5
## 6: versicolor          6.8         2.8          4.8         1.4
## 7:     setosa          5.8         4.0          1.2         0.2
## 8:     setosa          5.7         4.4          1.5         0.4
## 9:     setosa          5.7         3.8          1.7         0.3
```

对 iris 各个列排序，依次对第 5、1、2、3 列升序排列

```
ind <- do.call(what = "order", args = iris[,c(5,1,2,3)])
head(iris[ind, ])
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1:          4.3         3.0          1.1         0.1  setosa
## 2:          4.4         2.9          1.4         0.2  setosa
## 3:          4.4         3.0          1.3         0.2  setosa
## 4:          4.4         3.2          1.3         0.2  setosa
## 5:          4.5         2.3          1.3         0.3  setosa
## 6:          4.6         3.1          1.5         0.2  setosa
```

©黄湘云

第七章　高级数据操作

```
library(data.table)
library(magrittr)
```

介绍 data.table 处理数据的方式，对标 dplyr 的基本操作

7.1　基础介绍

```
# 用一个真实的数据集替换，让每一个操作都有实际含义和价值 mtcars
DT <- data.table(
  x = rep(c("b", "a", "c"), each = 3),
  v = c(1, 1, 1, 2, 2, 1, 1, 2, 2),
  y = c(1, 3, 6), a = 1:9, b = 9:1
)
DT
```

```
##    x v y a b
## 1: b 1 1 1 9
## 2: b 1 3 2 8
## 3: b 1 6 3 7
## 4: a 2 1 4 6
## 5: a 2 3 5 5
## 6: a 1 6 6 4
## 7: c 1 1 7 3
## 8: c 2 3 8 2
## 9: c 2 6 9 1
```

```
# 分组求和
DT[, sum(v), by = .(y %% 2)]
```

```
##    y V1
## 1: 1  9
## 2: 0  4
```

```
DT[, sum(v), by = .(bool = y %% 2)]
```

```
##    bool V1
## 1:    1  9
```

© 黄湘云


```
## 2:    0  4
```

```
DT[, .SD[2], by = x] # 每组第二行
```

```
##    x v y a b
## 1: b 1 3 2 8
## 2: a 2 3 5 5
## 3: c 2 3 8 2
```

```
DT[, tail(.SD, 2), by = x] # 每组最后两行
```

```
##    x v y a b
## 1: b 1 3 2 8
## 2: b 1 6 3 7
## 3: a 2 3 5 5
## 4: a 1 6 6 4
## 5: c 2 3 8 2
## 6: c 2 6 9 1
```

```
# 除了 x 列外，所有列都按 x 分组求和
DT[, lapply(.SD, sum), by = x]
```

```
##    x v  y  a  b
## 1: b 3 10  6 24
## 2: a 5 10 15 15
## 3: c 5 10 24  6
```

```
# 各个列都按 x 分组取最小
DT[, .SD[which.min(v)], by = x] # 分组嵌套查询
```

```
##    x v y a b
## 1: b 1 1 1 9
## 2: a 1 6 6 4
## 3: c 1 1 7 3
```

```
DT[, list(MySum = sum(v), MyMin = min(v), MyMax = max(v)), by = .(x, y %% 2)] # 表达式嵌套
```

```
##    x y MySum MyMin MyMax
## 1: b 1     2     1     1
## 2: b 0     1     1     1
## 3: a 1     4     2     2
## 4: a 0     1     1     1
## 5: c 1     3     1     2
## 6: c 0     2     2     2
```

```
DT[, .(a = .(a), b = .(b)), by = x] # 按 x 分组，将 a,b 两列的值列出来
```

```
##    x     a     b
## 1: b 1,2,3 9,8,7
## 2: a 4,5,6 6,5,4
## 3: c 7,8,9 3,2,1
```

© 基础

```
DT[, .(seq = min(a):max(b)), by = x] # 列操作不仅仅是聚合
```

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

```
# 按 x 分组对 v 求和，然后过滤出和小于 20 的行
DT[, sum(v), by = x][V1 < 20] # 组合查询
```

```
##    x V1
## 1: b  3
## 2: a  5
## 3: c  5
```

```
DT[, sum(v), by = x][order(-V1)] # 对结果排序
```

```
##    x V1
## 1: a  5
## 2: c  5
## 3: b  3
```

```
DT[, c(.N, lapply(.SD, sum)), by = x] # 计算每一组的和，每一组的观测数
```

```
##    x N v  y  a  b
## 1: b 3 3 10  6 24
## 2: a 3 5 10 15 15
## 3: c 3 5 10 24  6
```

```
# 两个复杂的操作，还没弄清楚这个技术存在的意义
DT[,
  {
    tmp <- mean(y)
    .(a = a - tmp, b = b - tmp)
```

© 数据科学


```
  },
  by = x
] # anonymous lambda in 'j', j accepts any valid
```

```
##    x          a          b
## 1: b -2.3333333  5.6666667
## 2: b -1.3333333  4.6666667
## 3: b -0.3333333  3.6666667
## 4: a  0.6666667  2.6666667
## 5: a  1.6666667  1.6666667
## 6: a  2.6666667  0.6666667
## 7: c  3.6666667 -0.3333333
## 8: c  4.6666667 -1.3333333
## 9: c  5.6666667 -2.3333333
```

```
# using rleid, get max(y) and min of all cols in .SDcols for each consecutive run of 'v'
DT[, c(.(y = max(y)), lapply(.SD, min)), by = rleid(v), .SDcols = v:b]
```

```
##    rleid y v y a b
## 1:     1 6 1 1 1 7
## 2:     2 3 2 1 4 5
## 3:     3 6 1 1 6 3
## 4:     4 6 2 3 8 1
```

7.1.1 过滤

```
mtcars_df = as.data.table(mtcars)
```

过滤 cyl = 6 并且 gear = 4 的记录

```
mtcars_df[cyl == 6 & gear == 4]
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1: 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2: 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 3: 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## 4: 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
```

过滤操作是针对数据框的行（记录）

```
mtcars_df[cyl == 6 & gear == 4, .(mpg, disp)]
```

```
##     mpg  disp
## 1: 21.0 160.0
## 2: 21.0 160.0
## 3: 19.2 167.6
## 4: 17.8 167.6
```

©清华大学出版社

```
subset(x = mtcars_df, subset = cyl == 6 & gear == 4, select = c(mpg, disp))
```

```
##     mpg  disp
## 1: 21.0 160.0
## 2: 21.0 160.0
## 3: 19.2 167.6
## 4: 17.8 167.6
```

```
mtcars %>%
  dplyr::filter(cyl == 6 & gear == 4)  %>%
  dplyr::select(mpg, disp)
```

```
##                mpg  disp
## Mazda RX4     21.0 160.0
## Mazda RX4 Wag 21.0 160.0
## Merc 280      19.2 167.6
## Merc 280C     17.8 167.6
```

7.1.2 变换

根据已有的列生成新的列，或者修改已有的列，一次只能修改一列

```
mtcars_df[, mean_mpg := mean(mpg)
          ][, mean_disp := mean(disp)]
mtcars_df[1:6, ]
```

```
##     mpg cyl disp  hp drat    wt  qsec vs am gear carb mean_mpg mean_disp
## 1: 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4 20.09062  230.7219
## 2: 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4 20.09062  230.7219
## 3: 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1 20.09062  230.7219
## 4: 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1 20.09062  230.7219
## 5: 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2 20.09062  230.7219
## 6: 18.1   6  225 105 2.76 3.460 20.22  1  0    3    1 20.09062  230.7219
```

```
mtcars_df[, .(mean_mpg = mean(mpg), mean_disp = mean(disp))]
```

```
##    mean_mpg mean_disp
## 1: 20.09062  230.7219
```

```
# mtcars_df[, .(mean_mpg := mean(mpg), mean_disp := mean(disp))] # 报错
# 正确的姿势
mtcars_df[, `:=`(mean_mpg = mean(mpg), mean_disp = mean(disp))
          ][, .(mpg, disp, mean_mpg, mean_disp)] %>% head()
```

```
##     mpg disp mean_mpg mean_disp
## 1: 21.0  160 20.09062  230.7219
## 2: 21.0  160 20.09062  230.7219
## 3: 22.8  108 20.09062  230.7219
## 4: 21.4  258 20.09062  230.7219
```

© 基础


```
## 5: 18.7  360 20.09062  230.7219
## 6: 18.1  225 20.09062  230.7219
```

```
mtcars %>%
  dplyr::summarise(mean_mpg = mean(mpg), mean_disp = mean(disp))
```

```
##   mean_mpg mean_disp
## 1 20.09062  230.7219
```

```
mtcars %>%
  dplyr::mutate(mean_mpg = mean(mpg), mean_disp = mean(disp)) %>%
  dplyr::select(mpg, disp, mean_mpg, mean_disp) %>% head()
```

```
##                    mpg disp mean_mpg mean_disp
## Mazda RX4         21.0  160 20.09062  230.7219
## Mazda RX4 Wag     21.0  160 20.09062  230.7219
## Datsun 710        22.8  108 20.09062  230.7219
## Hornet 4 Drive    21.4  258 20.09062  230.7219
## Hornet Sportabout 18.7  360 20.09062  230.7219
## Valiant           18.1  225 20.09062  230.7219
```

7.1.3 聚合

分组统计多个分组变量

```
dcast(mtcars_df, cyl ~ gear, value.var = "mpg", fun = mean)
```

```
##    cyl     3      4    5
## 1:   4 21.50 26.925 28.2
## 2:   6 19.75 19.750 19.7
## 3:   8 15.05    NaN 15.4
```

```
tapply(mtcars$mpg, list(mtcars$cyl, mtcars$gear), mean)
```

```
##       3      4    5
## 4 21.50 26.925 28.2
## 6 19.75 19.750 19.7
## 8 15.05     NA 15.4
```

```
mtcars_df[, .(mean_mpg = mean(mpg)), by = .(cyl, gear)]
```

```
##    cyl gear mean_mpg
## 1:   6    4   19.750
## 2:   4    4   26.925
## 3:   6    3   19.750
## 4:   8    3   15.050
## 5:   4    3   21.500
## 6:   4    5   28.200
## 7:   8    5   15.400
## 8:   6    5   19.700
```

© 清华大学出版社

```
aggregate(data = mtcars_df, mpg ~ cyl + gear, FUN = mean)
```

```
##   cyl gear    mpg
## 1   4    3 21.500
## 2   6    3 19.750
## 3   8    3 15.050
## 4   4    4 26.925
## 5   6    4 19.750
## 6   4    5 28.200
## 7   6    5 19.700
## 8   8    5 15.400
```

```
mtcars %>%
  dplyr::group_by(cyl, gear) %>%
  dplyr::summarise(mean_mpg = mean(mpg))
```

```
## # A tibble: 8 x 3
## # Groups:   cyl [3]
##     cyl  gear mean_mpg
##   <dbl> <dbl>    <dbl>
## 1     4     3     21.5
## 2     4     4     26.9
## 3     4     5     28.2
## 4     6     3     19.8
## 5     6     4     19.8
## 6     6     5     19.7
## 7     8     3     15.0
## 8     8     5     15.4
```

7.1.4 命名

修改列名，另存一份生效

```
sub_mtcars_df <- mtcars_df[, .(mean_mpg = mean(mpg)), by = .(cyl, gear)]
setNames(sub_mtcars_df, c("cyl", "gear", "ave_mpg"))
```

```
##    cyl gear ave_mpg
## 1:   6    4  19.750
## 2:   4    4  26.925
## 3:   6    3  19.750
## 4:   8    3  15.050
## 5:   4    3  21.500
## 6:   4    5  28.200
## 7:   8    5  15.400
## 8:   6    5  19.700
```

©黄湘云


```
# 注意 sub_mtcars_df 并没有修改列名
sub_mtcars_df
```

```
##    cyl gear mean_mpg
## 1:   6    4   19.750
## 2:   4    4   26.925
## 3:   6    3   19.750
## 4:   8    3   15.050
## 5:   4    3   21.500
## 6:   4    5   28.200
## 7:   8    5   15.400
## 8:   6    5   19.700
```

修改列名并直接起作用，在原来的数据集上生效

```
setnames(sub_mtcars_df, old = c("mean_mpg"), new = c("ave_mpg"))
# sub_mtcars_df 已经修改了列名
sub_mtcars_df
```

```
##    cyl gear ave_mpg
## 1:   6    4  19.750
## 2:   4    4  26.925
## 3:   6    3  19.750
## 4:   8    3  15.050
## 5:   4    3  21.500
## 6:   4    5  28.200
## 7:   8    5  15.400
## 8:   6    5  19.700
```

修改列名最好使用 **data.table** 包的函数 `setnames()` 明确指出了要修改的列名，

7.1.5 排序

按照某（些）列从大到小或从小到大的顺序排列，先按 cyl 升序，然后按 gear 降序

```
mtcars_df[, .(mpg, cyl, gear)
          ][cyl == 4
            ][order(cyl, -gear)]
```

```
##      mpg cyl gear
##  1: 26.0   4    5
##  2: 30.4   4    5
##  3: 22.8   4    4
##  4: 24.4   4    4
##  5: 22.8   4    4
##  6: 32.4   4    4
##  7: 30.4   4    4
##  8: 33.9   4    4
```

©清华大学出版社

```
##  9: 27.3   4    4
## 10: 21.4   4    4
## 11: 21.5   4    3
```

```
mtcars %>%
  dplyr::select(mpg, cyl, gear) %>%
  dplyr::filter(cyl == 4) %>%
  dplyr::arrange(cyl, desc(gear))
```

```
##                 mpg cyl gear
## Porsche 914-2  26.0   4    5
## Lotus Europa   30.4   4    5
## Datsun 710     22.8   4    4
## Merc 240D      24.4   4    4
## Merc 230       22.8   4    4
## Fiat 128       32.4   4    4
## Honda Civic    30.4   4    4
## Toyota Corolla 33.9   4    4
## Fiat X1-9      27.3   4    4
## Volvo 142E     21.4   4    4
## Toyota Corona  21.5   4    3
```

7.1.6 变形

melt 宽的变长的

```
DT <- data.table(
  i_1 = c(1:5, NA),
  i_2 = c(NA, 6, 7, 8, 9, 10),
  f_1 = factor(sample(c(letters[1:3], NA), 6, TRUE)),
  f_2 = factor(c("z", "a", "x", "c", "x", "x"), ordered = TRUE),
  c_1 = sample(c(letters[1:3], NA), 6, TRUE),
  d_1 = as.Date(c(1:3, NA, 4:5), origin = "2013-09-01"),
  d_2 = as.Date(6:1, origin = "2012-01-01")
)
```

```
DT[, .(i_1, i_2, f_1, f_2)]
```

```
##    i_1 i_2  f_1 f_2
## 1:   1  NA    a   z
## 2:   2   6 <NA>   a
## 3:   3   7 <NA>   x
## 4:   4   8 <NA>   c
## 5:   5   9    c   x
## 6:  NA  10 <NA>   x
```

```
melt(DT, id = 1:2, measure = c("f_1", "f_2"))
```

©黄海广


```
##     i_1 i_2 variable value
##  1:   1  NA      f_1     a
##  2:   2   6      f_1  <NA>
##  3:   3   7      f_1  <NA>
##  4:   4   8      f_1  <NA>
##  5:   5   9      f_1     c
##  6:  NA  10      f_1  <NA>
##  7:   1  NA      f_2     z
##  8:   2   6      f_2     a
##  9:   3   7      f_2     x
## 10:   4   8      f_2     c
## 11:   5   9      f_2     x
## 12:  NA  10      f_2     x
```

dcast 长的变宽的

```
sleep <- as.data.table(sleep)
dcast(sleep, group ~ ID, value.var = "extra")
```

```
##    group   1    2    3    4    5   6   7   8   9  10
## 1:     1 0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0.0 2.0
## 2:     2 1.9  0.8  1.1  0.1 -0.1 4.4 5.5 1.6 4.6 3.4
```

```
# 如果有多个值
dcast(mtcars_df, cyl ~ gear, value.var = "mpg")
```

```
##    cyl  3 4 5
## 1:   4  1 8 2
## 2:   6  2 4 1
## 3:   8 12 0 2
```

```
dcast(mtcars_df, cyl ~ gear, value.var = "mpg", fun = mean)
```

```
##    cyl     3      4    5
## 1:   4 21.50 26.925 28.2
## 2:   6 19.75 19.750 19.7
## 3:   8 15.05    NaN 15.4
```

tidyr 包提供数据变形的函数 `tidyr::pivot_longer()` 和 `tidyr::pivot_wider()` 相比于 Base R 提供的 `reshape()` 和 data.table 提供的 `melt()` 和 `dcast()` 更加形象的命名

```
tidyr::pivot_wider(data = sleep, names_from = "ID", values_from = "extra")
```

```
## # A tibble: 2 x 11
##   group   `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `9`  `10`
##   <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 1       0.7  -1.6  -0.2  -1.2  -0.1   3.4   3.7   0.8   0     2
## 2 2       1.9   0.8   1.1   0.1  -0.1   4.4   5.5   1.6   4.6   3.4
```

```
reshape(data = sleep, v.names = "extra", idvar = "group", timevar = "ID", direction = "wide")
```

```
##    group extra.1 extra.2 extra.3 extra.4 extra.5 extra.6 extra.7 extra.8
```

©黄天祥

```
## 1:     1     0.7    -1.6    -0.2    -1.2    -0.1     3.4     3.7     0.8
## 2:     2     1.9     0.8     1.1     0.1    -0.1     4.4     5.5     1.6
##    extra.9 extra.10
## 1:     0.0      2.0
## 2:     4.6      3.4
```

- idvar 分组变量
- timevar 组内编号
- v.names 个体观察值

注意

sep 新的列名是由参数 v.names (extra) 和参数值 timevar (ID) 拼接起来的，默认 sep = "." 推荐使用下划线来做分割 sep = "_"

```
ToothGrowth %>% head
```

```
##    len supp dose
## 1  4.2   VC  0.5
## 2 11.5   VC  0.5
## 3  7.3   VC  0.5
## 4  5.8   VC  0.5
## 5  6.4   VC  0.5
## 6 10.0   VC  0.5
```

```
ToothGrowth$time <- rep(1:10, 6)
reshape(ToothGrowth, v.names = "len", idvar = c("supp", "dose"),
        timevar = "time", direction = "wide")
```

```
##    supp dose len.1 len.2 len.3 len.4 len.5 len.6 len.7 len.8 len.9 len.10
## 1    VC  0.5   4.2  11.5   7.3   5.8   6.4  10.0  11.2  11.2   5.2    7.0
## 11   VC  1.0  16.5  16.5  15.2  17.3  22.5  17.3  13.6  14.5  18.8   15.5
## 21   VC  2.0  23.6  18.5  33.9  25.5  26.4  32.5  26.7  21.5  23.3   29.5
## 31   OJ  0.5  15.2  21.5  17.6   9.7  14.5  10.0   8.2   9.4  16.5    9.7
## 41   OJ  1.0  19.7  23.3  23.6  26.4  20.0  25.2  25.8  21.2  14.5   27.3
## 51   OJ  2.0  25.5  26.4  22.4  24.5  24.8  30.9  26.4  27.3  29.4   23.0
```

以数据集 ToothGrowth 为例，变量 supp（大组），dose（小组）和 time（组内个体编号）一起决定唯一的一个数据 len，特别适合纵向数据的变形操作

7.1.7 分组

分组切片，取每组第一个和最后一个值

```
Loblolly %>%
  dplyr::group_by(Seed) %>%
  dplyr::arrange(height, age, Seed) %>%
  dplyr::slice(1, dplyr::n())
```

```
## # A tibble: 28 x 3
```

©黄湘云


```
## # Groups:   Seed [14]
##    height   age Seed
##     <dbl> <dbl> <ord>
##  1   3.93     3 329
##  2  56.4     25 329
##  3   4.12     3 327
##  4  56.8     25 327
##  5   4.38     3 325
##  6  58.5     25 325
##  7   3.91     3 307
##  8  59.1     25 307
##  9   3.46     3 331
## 10  59.5     25 331
## # ... with 18 more rows
```

`dplyr::slice()` 和函数 `slice.index()` 有关系吗?

7.1.8 合并

合并操作对应于数据库中的连接操作，dplyr 包的哲学就来源于对数据库操作的进一步抽象，data.table 包的 merge 函数就对应为 dplyr 包的 join 函数

`data.table::merge` 和 `dplyr::join`

给出一个表格，数据操作，data.table 实现，dplyr 实现

```
dt1 <- data.table(A = letters[1:10], X = 1:10, key = "A")
dt2 <- data.table(A = letters[5:14], Y = 1:10, key = "A")
merge(dt1, dt2) # 内连接
```

```
##    A  X Y
## 1: e  5 1
## 2: f  6 2
## 3: g  7 3
## 4: h  8 4
## 5: i  9 5
## 6: j 10 6
```

参数 key 的作用相当于建立一个索引，通过它实现更快的数据操作速度

`key = c("x","y","z")` 或者 `key = "x,y,z"` 其中 x,y,z 是列名

```
data(band_members, band_instruments, package = "dplyr")
band_members
```

```
## # A tibble: 3 x 2
##   name  band
##   <chr> <chr>
## 1 Mick  Stones
## 2 John  Beatles
```

内部使用 © 黄


```
## 3 Paul  Beatles
```

```
band_instruments
```

```
## # A tibble: 3 x 2
##   name  plays
##   <chr> <chr>
## 1 John  guitar
## 2 Paul  bass
## 3 Keith guitar
```

```
dplyr::inner_join(band_members, band_instruments)
```

```
## # A tibble: 2 x 3
##   name  band    plays
##   <chr> <chr>   <chr>
## 1 John  Beatles guitar
## 2 Paul  Beatles bass
```

list 列表里每个元素都是 data.frame 时，最适合用 `data.table::rbindlist` 合并

```
# 合并列表 https://recology.info/2018/10/limiting-dependencies/
function(x) {
  tibble::as_tibble((x <- data.table::setDF(
    data.table::rbindlist(x, use.names = TRUE, fill = TRUE, idcol = "id"))
  ))
}
```

```
## function(x) {
##   tibble::as_tibble((x <- data.table::setDF(
##     data.table::rbindlist(x, use.names = TRUE, fill = TRUE, idcol = "id"))
##   ))
## }
```

7.2 高频操作

以面向问题的方式介绍 Base R 提供的数据操作，然后过渡到 data.table，它是加强版的 Base R。

表 7.1: 单表的操作

base	dplyr
`df[order(x), , drop = FALSE]`	`arrange(df, x)`
`df[!duplicated(x), , drop = FALSE],unique()`	`distinct(df, x)`
`df[x & !is.na(x), , drop = FALSE],subset()`	`filter(df, x)`
`df$z <- df$x + df$y,transform()`	`mutate(df, z = x + y)`
`df$x`	`pull(df, x)`
N/A	`rename(df, y = x)`
`df[c("x", "y")],subset()`	`select(df, x, y)`

©黄湘云

base	dplyr
`df[grepl(names(df), "^x")]`	`select(df, starts_with("x")`
`mean(df$x)`	`summarise(df, mean(x))`
`df[c(1, 2, 5), , drop = FALSE]`	`slice(df, c(1, 2, 5))`

表 7.2: 两表的操作

base	dplyr
`merge(df1, df2)`	`inner_join(df1, df2)`
`merge(df1, df2, all.x = TRUE)`	`left_join(df1, df2)`
`merge(df1, df2, all.y = TRUE)`	`right_join(df1, df2)`
`merge(df1, df2, all = TRUE)`	`full_join(df1, df2)`
`df1[df1$x %in% df2$x, , drop = FALSE]`	`semi_join(df1, df2)`
`df1[!df1$x %in% df2$x, , drop = FALSE]`	`anti_join(df1, df2)`

```
library(magrittr)
class(mtcars)
```

```
## [1] "data.frame"
```

```
library(data.table)
mtcars <- as.data.table(mtcars)
class(mtcars)
```

```
## [1] "data.table" "data.frame"
```

7.2.1 选择多列

```
# base
mtcars[, c("cyl", "gear")] %>% head(3)
```

```
##    cyl gear
## 1:   6    4
## 2:   6    4
## 3:   4    4
```

```
# data.table
mtcars[, c("cyl", "gear")] %>% head(3)
```

```
##    cyl gear
## 1:   6    4
## 2:   6    4
## 3:   4    4
```

```
# dplyr
dplyr::select(mtcars, cyl, gear) %>% head(3)
```

©清华大学出版社

```
##    cyl gear
## 1:   6    4
## 2:   6    4
## 3:   4    4
```

反选多列，选择除了 cyl 和 gear 的列

```
## 或者 mtcars[, setdiff(names(mtcars), c("cyl", "gear"))]
mtcars[ , !(names(mtcars) %in% c("cyl","gear"))]  %>% head(3)
```

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

```
subset(mtcars, select = -c(cyl, gear)) %>% head(3)
```

```
##     mpg disp  hp drat    wt  qsec vs am carb
## 1: 21.0  160 110 3.90 2.620 16.46  0  1    4
## 2: 21.0  160 110 3.90 2.875 17.02  0  1    4
## 3: 22.8  108  93 3.85 2.320 18.61  1  1    1
```

7.2.2 过滤多行

```
# base
mtcars[mtcars$cyl == 6 & mtcars$gear == 4,]
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1: 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2: 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 3: 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## 4: 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
```

```
subset(mtcars, subset = cyl == 6 & gear == 4)
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1: 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2: 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 3: 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## 4: 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
```

```
# data.table
mtcars[cyl == 6 & gear == 4,]
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1: 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2: 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 3: 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## 4: 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
```

```
# dplyr
dplyr::filter(mtcars, cyl == 6 & gear == 4)
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
```

©黄海广


```
## 1: 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2: 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## 3: 19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## 4: 17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
```

7.2.3 去重多行

```
# base
mtcars[!duplicated(mtcars[, c("cyl", "gear")])]
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1: 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2: 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 3: 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 4: 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 5: 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## 6: 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## 7: 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## 8: 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
```

```
# data.table
mtcars[!duplicated(mtcars, by = c("cyl", "gear")), ]
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1: 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2: 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 3: 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 4: 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 5: 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## 6: 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## 7: 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## 8: 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
```

```
unique(mtcars, by = c("cyl", "gear"))
```

```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1: 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2: 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 3: 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 4: 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 5: 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## 6: 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## 7: 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## 8: 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
```

```
# dplyr
dplyr::distinct(mtcars, cyl, gear, .keep_all = TRUE)
```

© 黄湘云


```
##     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 1: 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## 2: 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 3: 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## 4: 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## 5: 21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## 6: 26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## 7: 15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## 8: 19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
```

7.2.4 合并操作

在数据库的操作中，合并又称为连接

7.2.4.1 左合并

```
# dplyr::inner_join()
# dplyr::left_join()
# dplyr::right_join()
# dplyr::full_join()
```

7.2.4.2 右合并

7.2.5 新添多列

```
mtcars[cyl == 6, `:=`(disp_mean = mean(disp), hp_mean = mean(hp))][cyl == 6, .(cyl, disp, hp, disp_mean, h
```

```
##    cyl  disp  hp disp_mean  hp_mean
## 1:   6 160.0 110  183.3143 122.2857
## 2:   6 160.0 110  183.3143 122.2857
## 3:   6 258.0 110  183.3143 122.2857
## 4:   6 225.0 105  183.3143 122.2857
## 5:   6 167.6 123  183.3143 122.2857
## 6:   6 167.6 123  183.3143 122.2857
## 7:   6 145.0 175  183.3143 122.2857
```

7.2.6 删除多列

删除列就是将该列的值清空，置为 NULL，下面将新添的两个列删除，根据列名的特点用正则表达式匹配

```
mtcars[, colnames(mtcars)[grep('_mean$', colnames(mtcars))] := NULL]
```

©黄天宏

7.2.7 修改多列类型

```
mtcars[, (c("cyl", "disp")) := lapply(.SD, as.integer), .SDcols = c("cyl", "disp")]
str(mtcars)
```

```
## Classes 'data.table' and 'data.frame':   32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : int  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: int  160 160 108 258 360 225 360 146 140 167 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "index")= int(0)
```

7.2.8 取每组第一行

先将 mtcars 按 cyl 升序，gear 降序排列，然后按 cyl, gear 和 am 分组取第一行

```
mtcars[order(cyl, - gear)][, head(.SD, 1), by = list(cyl, gear, am)]
```

```
##     cyl gear am  mpg disp  hp drat    wt  qsec vs carb
##  1:   4    5  1 26.0  120  91 4.43 2.140 16.70  0    2
##  2:   4    4  1 22.8  108  93 3.85 2.320 18.61  1    1
##  3:   4    4  0 24.4  146  62 3.69 3.190 20.00  1    2
##  4:   4    3  0 21.5  120  97 3.70 2.465 20.01  1    1
##  5:   6    5  1 19.7  145 175 3.62 2.770 15.50  0    6
##  6:   6    4  1 21.0  160 110 3.90 2.620 16.46  0    4
##  7:   6    4  0 19.2  167 123 3.92 3.440 18.30  1    4
##  8:   6    3  0 21.4  258 110 3.08 3.215 19.44  1    1
##  9:   8    5  1 15.8  351 264 4.22 3.170 14.50  0    4
## 10:   8    3  0 18.7  360 175 3.15 3.440 17.02  0    2
```

```
# 或者
mtcars[order(cyl, - gear)][, .SD[1], by = list(cyl, gear, am)]
```

```
##     cyl gear am  mpg disp  hp drat    wt  qsec vs carb
##  1:   4    5  1 26.0  120  91 4.43 2.140 16.70  0    2
##  2:   4    4  1 22.8  108  93 3.85 2.320 18.61  1    1
##  3:   4    4  0 24.4  146  62 3.69 3.190 20.00  1    2
##  4:   4    3  0 21.5  120  97 3.70 2.465 20.01  1    1
##  5:   6    5  1 19.7  145 175 3.62 2.770 15.50  0    6
```

© 黄湘云

表 7.3: 1949-1960 年国际航班乘客数量变化

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1949	112	118	132	129	121	135	148	148	136	119	104	118
1950	115	126	141	135	125	149	170	170	158	133	114	140
1951	145	150	178	163	172	178	199	199	184	162	146	166
1952	171	180	193	181	183	218	230	242	209	191	172	194
1953	196	196	236	235	229	243	264	272	237	211	180	201
1954	204	188	235	227	234	264	302	293	259	229	203	229
1955	242	233	267	269	270	315	364	347	312	274	237	278
1956	284	277	317	313	318	374	413	405	355	306	271	306
1957	315	301	356	348	355	422	465	467	404	347	305	336
1958	340	318	362	348	363	435	491	505	404	359	310	337
1959	360	342	406	396	420	472	548	559	463	407	362	405
1960	417	391	419	461	472	535	622	606	508	461	390	432

```
##  6:   6    4  1 21.0  160 110 3.90 2.620 16.46  0    4
##  7:   6    4  0 19.2  167 123 3.92 3.440 18.30  1    4
##  8:   6    3  0 21.4  258 110 3.08 3.215 19.44  1    1
##  9:   8    5  1 15.8  351 264 4.22 3.170 14.50  0    4
## 10:   8    3  0 18.7  360 175 3.15 3.440 17.02  0    2
```

7.2.9 计算环比同比

以数据集 AirPassengers 为例，重新整理后见表 7.3

```
library(magrittr)
dat <- data.frame(
  year = rep(1949:1960, each = 12),
  month = month.abb, num = AirPassengers
) %>%
  reshape(.,
    v.names = "num", idvar = "year", timevar = "month",
    direction = "wide", sep = ""
  ) %>%
  setNames(., gsub(pattern = "(num)", replacement = "", x = colnames(.)))

rownames(dat) <- subset(dat, select = year, drop = TRUE)
air_passengers <- subset(dat, select = -year)

knitr::kable(air_passengers,
    caption = "1949-1960年国际航班乘客数量变化",
    align = "c", row.names = TRUE
  )
```

©黄湘云

横向计算环比，如 1949 年 2 月相比 1 月增长多少、3 月相比 2 月增长多少，以此类推，就是计算环比？纵向计算同比，如 1950 年 1 月相比 1949 年 1 月增长多少、1951 年相比 1950 年 1 月增长多少？

```
# 环比横向/同比纵向
mom <- function(x) diff(x, lag = 1)/x[-length(x)] # month to month
# 格式化输出
format_mom <- function(x) formatC(mom(x), format = "f", digits = 4)
```

```
library(formattable)
# 同比变化
air_passengers %>%
  apply(., 2, format_mom) %>% as.data.frame() %>%
  formattable(., list(
        Jan = color_tile("white", "pink"),
        Feb = color_tile("white", "springgreen4"),
        Mar = percent
      ))

library(DT)
datatable(air_passengers)
```

7.2.10 合并多个数据框

将所有列都保留，以 full_join() 方式合并

```
df1 <- iris[1:10, c(1, 5)]
df2 <- iris[11:15, c(1, 2, 5)]
df3 <- iris[16:30, c(1, 3, 5)]
all_dfs <- list(df1, df2, df3)
# base
Reduce(function(x, y, ...) merge(x, y, ..., all = TRUE), all_dfs)
```

```
##    Sepal.Length Species Sepal.Width Petal.Length
## 1           4.3  setosa         3.0           NA
## 2           4.4  setosa          NA           NA
## 3           4.6  setosa          NA          1.0
## 4           4.6  setosa          NA          1.0
## 5           4.7  setosa          NA          1.6
## 6           4.8  setosa         3.0          1.9
## 7           4.8  setosa         3.4          1.9
## 8           4.9  setosa          NA           NA
## 9           4.9  setosa          NA           NA
## 10          5.0  setosa          NA          1.6
## 11          5.0  setosa          NA          1.6
## 12          5.0  setosa          NA          1.6
## 13          5.0  setosa          NA          1.6
## 14          5.1  setosa          NA          1.5
```

©黄天源

```
## 15          5.1  setosa          NA          1.4
## 16          5.1  setosa          NA          1.7
## 17          5.1  setosa          NA          1.5
## 18          5.2  setosa          NA          1.4
## 19          5.2  setosa          NA          1.5
## 20          5.4  setosa         3.7          1.3
## 21          5.4  setosa         3.7          1.7
## 22          5.7  setosa          NA          1.5
## 23          5.7  setosa          NA          1.7
## 24          5.8  setosa         4.0           NA
```

```
# dplyr
Reduce(function(x, y, ...) dplyr::full_join(x, y, ...), all_dfs)
```

```
##    Sepal.Length Species Sepal.Width Petal.Length
## 1           5.1  setosa          NA          1.4
## 2           5.1  setosa          NA          1.5
## 3           5.1  setosa          NA          1.5
## 4           5.1  setosa          NA          1.7
## 5           4.9  setosa          NA           NA
## 6           4.7  setosa          NA          1.6
## 7           4.6  setosa          NA          1.0
## 8           5.0  setosa          NA          1.6
## 9           5.0  setosa          NA          1.6
## 10          5.4  setosa         3.7          1.3
## 11          5.4  setosa         3.7          1.7
## 12          4.6  setosa          NA          1.0
## 13          5.0  setosa          NA          1.6
## 14          5.0  setosa          NA          1.6
## 15          4.4  setosa          NA           NA
## 16          4.9  setosa          NA           NA
## 17          4.8  setosa         3.4          1.9
## 18          4.8  setosa         3.0          1.9
## 19          4.3  setosa         3.0           NA
## 20          5.8  setosa         4.0           NA
## 21          5.7  setosa          NA          1.5
## 22          5.7  setosa          NA          1.7
## 23          5.2  setosa          NA          1.5
## 24          5.2  setosa          NA          1.4
```

合并完应该有 30 行，为啥只有 24 行？这是因为 `merge()` 函数对主键 key 相同的记录会合并，要想不合并，需要调用 `rbindlist()` 函数 https://d.cosx.org/d/421235

`rbind()` 列数相同的两个 data.frame 按行合并，`cbind()` 行数相同的两个 data.frame 按列合并，`merge()` 对行、列数没有要求

```
rbindlist(all_dfs, fill = TRUE)
```

©黄海广

```
##     Sepal.Length Species Sepal.Width Petal.Length
##  1:          5.1  setosa          NA           NA
##  2:          4.9  setosa          NA           NA
##  3:          4.7  setosa          NA           NA
##  4:          4.6  setosa          NA           NA
##  5:          5.0  setosa          NA           NA
##  6:          5.4  setosa          NA           NA
##  7:          4.6  setosa          NA           NA
##  8:          5.0  setosa          NA           NA
##  9:          4.4  setosa          NA           NA
## 10:          4.9  setosa          NA           NA
## 11:          5.4  setosa         3.7           NA
## 12:          4.8  setosa         3.4           NA
## 13:          4.8  setosa         3.0           NA
## 14:          4.3  setosa         3.0           NA
## 15:          5.8  setosa         4.0           NA
## 16:          5.7  setosa          NA          1.5
## 17:          5.4  setosa          NA          1.3
## 18:          5.1  setosa          NA          1.4
## 19:          5.7  setosa          NA          1.7
## 20:          5.1  setosa          NA          1.5
## 21:          5.4  setosa          NA          1.7
## 22:          5.1  setosa          NA          1.5
## 23:          4.6  setosa          NA          1.0
## 24:          5.1  setosa          NA          1.7
## 25:          4.8  setosa          NA          1.9
## 26:          5.0  setosa          NA          1.6
## 27:          5.0  setosa          NA          1.6
## 28:          5.2  setosa          NA          1.5
## 29:          5.2  setosa          NA          1.4
## 30:          4.7  setosa          NA          1.6
##     Sepal.Length Species Sepal.Width Petal.Length
```

```
# dplyr
dplyr::bind_rows(all_dfs)
```

```
##    Sepal.Length Species Sepal.Width Petal.Length
## 1           5.1  setosa          NA           NA
## 2           4.9  setosa          NA           NA
## 3           4.7  setosa          NA           NA
## 4           4.6  setosa          NA           NA
## 5           5.0  setosa          NA           NA
## 6           5.4  setosa          NA           NA
## 7           4.6  setosa          NA           NA
## 8           5.0  setosa          NA           NA
## 9           4.4  setosa          NA           NA
```

©黄湘云

```
## 10          4.9  setosa          NA           NA
## 11          5.4  setosa         3.7           NA
## 12          4.8  setosa         3.4           NA
## 13          4.8  setosa         3.0           NA
## 14          4.3  setosa         3.0           NA
## 15          5.8  setosa         4.0           NA
## 16          5.7  setosa          NA          1.5
## 17          5.4  setosa          NA          1.3
## 18          5.1  setosa          NA          1.4
## 19          5.7  setosa          NA          1.7
## 20          5.1  setosa          NA          1.5
## 21          5.4  setosa          NA          1.7
## 22          5.1  setosa          NA          1.5
## 23          4.6  setosa          NA          1.0
## 24          5.1  setosa          NA          1.7
## 25          4.8  setosa          NA          1.9
## 26          5.0  setosa          NA          1.6
## 27          5.0  setosa          NA          1.6
## 28          5.2  setosa          NA          1.5
## 29          5.2  setosa          NA          1.4
## 30          4.7  setosa          NA          1.6
```

7.2.11 分组聚合多个指标

https://stackoverflow.com/questions/24151602/calculate-multiple-aggregations-with-lapply-sd

```r
# base
aggregate(
  data = mtcars, cbind(mpg, hp) ~ cyl,
  FUN = function(x) c(mean = mean(x), median = median(x))
)
```

```
##   cyl mpg.mean mpg.median   hp.mean hp.median
## 1   4 26.66364   26.00000  82.63636  91.00000
## 2   6 19.74286   19.70000 122.28571 110.00000
## 3   8 15.10000   15.20000 209.21429 192.50000
```

```r
# 数据一致性 https://d.cosx.org/d/420763-base-r
with(
  aggregate(cbind(mpg, hp) ~ cyl, mtcars,
    FUN = function(x) c(mean = mean(x), median = median(x))
  ),
  cbind.data.frame(cyl, mpg, hp)
)
```

```
##   cyl     mean median      mean median
## 1   4 26.66364   26.0  82.63636   91.0
```

© 王建硕


```
## 2   6 19.74286   19.7 122.28571  110.0
## 3   8 15.10000   15.2 209.21429  192.5
```

```
# data.table
mtcars[, as.list(unlist(lapply(.SD, function(x) {
  list(
    mean = mean(x),
    median = median(x)
  )
}))),
by = "cyl", .SDcols = c("mpg", "hp")
]
```

```
##    cyl mpg.mean mpg.median   hp.mean hp.median
## 1:   6 19.74286       19.7 122.28571     110.0
## 2:   4 26.66364       26.0  82.63636      91.0
## 3:   8 15.10000       15.2 209.21429     192.5
```

```
# dplyr
mtcars %>%
  dplyr::group_by(cyl) %>%
  dplyr::summarise(
    mean_mpg = mean(mpg), mean_hp = mean(hp),
    median_mpg = mean(mpg), median_hp = mean(hp)
  )
```

```
## # A tibble: 3 x 5
##     cyl mean_mpg mean_hp median_mpg median_hp
##   <int>    <dbl>   <dbl>      <dbl>     <dbl>
## 1     4     26.7    82.6       26.7      82.6
## 2     6     19.7   122.        19.7     122.
## 3     8     15.1   209.        15.1     209.
```

7.2.12 重命名多个列

```
tmp <- aggregate(data = mtcars, cbind(mpg, hp) ~ cyl,
          FUN = median)
tmp <- as.data.table(tmp)
setnames(tmp, old = c("mpg", "hp"), new = c("median_mpg", "median_hp"))
tmp
```

```
##    cyl median_mpg median_hp
## 1:   4       26.0      91.0
## 2:   6       19.7     110.0
## 3:   8       15.2     192.5
```

© 非洲出版社

7.2.13 对多个列依次排序

https://stackoverflow.com/questions/1296646/how-to-sort-a-dataframe-by-multiple-columns

```
# base
tmp[order(median_mpg, -median_hp), ]
```

```
##    cyl median_mpg median_hp
## 1:   8       15.2     192.5
## 2:   6       19.7     110.0
## 3:   4       26.0      91.0
```

```
# data.table
setorder(tmp, median_mpg, -median_hp)
# dplyr
dplyr::arrange(tmp, median_mpg, desc(median_hp))
```

```
##    cyl median_mpg median_hp
## 1:   8       15.2     192.5
## 2:   6       19.7     110.0
## 3:   4       26.0      91.0
```

7.2.14 重排多个列的位置

```
# https://stackoverflow.com/questions/19619666/change-column-position-of-data-table
setcolorder(tmp, c("median_mpg", setdiff(names(tmp), "median_mpg")))
tmp
```

```
##    median_mpg cyl median_hp
## 1:       15.2   8     192.5
## 2:       19.7   6     110.0
## 3:       26.0   4      91.0
```

```
# dplyr
dplyr::select(tmp, "median_mpg", setdiff(names(tmp), "median_mpg"))
```

```
##    median_mpg cyl median_hp
## 1:       15.2   8     192.5
## 2:       19.7   6     110.0
## 3:       26.0   4      91.0
```

7.2.15 整理回归结果

```
dat <- split(iris, iris$Species)
mod <- lapply(dat, function(x) lm(Petal.Length ~ Sepal.Length, x))
mod <- lapply(mod, function(x) coef(summary(x)))
mod <- Map(function(x, y) {x <- as.data.frame(x) ; x$Species = y; x}, mod, names(dat))
```

© 黄湘云


```
mod <- do.call(rbind, mod)
mod
```

```
##                          Estimate Std. Error    t value     Pr(>|t|)    Species
## setosa.(Intercept)      0.8030518 0.34387807  2.3352806 2.375647e-02     setosa
## setosa.Sepal.Length     0.1316317 0.06852690  1.9208760 6.069778e-02     setosa
## versicolor.(Intercept)  0.1851155 0.51421351  0.3599974 7.204283e-01 versicolor
## versicolor.Sepal.Length 0.6864698 0.08630708  7.9538056 2.586190e-10 versicolor
## virginica.(Intercept)   0.6104680 0.41710685  1.4635770 1.498279e-01  virginica
## virginica.Sepal.Length  0.7500808 0.06302606 11.9011203 6.297786e-16  virginica
```

```
# 管道操作
split(iris, iris$Species) %>%
  lapply(., function(x) coef(summary(lm(Petal.Length ~ Sepal.Length, x)))) %>%
  Map(function(x, y) {
    x <- as.data.frame(x)
    x$Species <- y
    x
  }, ., levels(iris$Species)) %>%
  do.call(rbind, .)
```

```
##                          Estimate Std. Error    t value     Pr(>|t|)    Species
## setosa.(Intercept)      0.8030518 0.34387807  2.3352806 2.375647e-02     setosa
## setosa.Sepal.Length     0.1316317 0.06852690  1.9208760 6.069778e-02     setosa
## versicolor.(Intercept)  0.1851155 0.51421351  0.3599974 7.204283e-01 versicolor
## versicolor.Sepal.Length 0.6864698 0.08630708  7.9538056 2.586190e-10 versicolor
## virginica.(Intercept)   0.6104680 0.41710685  1.4635770 1.498279e-01  virginica
## virginica.Sepal.Length  0.7500808 0.06302606 11.9011203 6.297786e-16  virginica
```

```
# dplyr 操作，需要 dplyr >= 1.0.0 或者开发版
iris %>%
  dplyr::group_by(Species) %>%
  dplyr::summarise(broom::tidy(lm(Petal.Length ~ Sepal.Length)))
```

```
## # A tibble: 6 x 6
## # Groups:   Species [3]
##   Species    term         estimate std.error statistic  p.value
##   <fct>      <chr>           <dbl>     <dbl>     <dbl>    <dbl>
## 1 setosa     (Intercept)     0.803    0.344      2.34  2.38e- 2
## 2 setosa     Sepal.Length    0.132    0.0685     1.92  6.07e- 2
## 3 versicolor (Intercept)     0.185    0.514      0.360 7.20e- 1
## 4 versicolor Sepal.Length    0.686    0.0863     7.95  2.59e-10
## 5 virginica  (Intercept)     0.610    0.417      1.46  1.50e- 1
## 6 virginica  Sepal.Length    0.750    0.0630    11.9   6.30e-16
```

©清华大学出版社

7.2.16 := 和 .()

```
mtcars[, mpg_rate := round(mpg/sum(mpg) * 100, digits = 2), by = .(cyl, vs, am)]
mtcars[, .(mpg_rate, mpg, cyl, vs, am)]
```

```
##     mpg_rate  mpg cyl vs am
##  1:    34.04 21.0   6  0  1
##  2:    34.04 21.0   6  0  1
##  3:    11.48 22.8   4  1  1
##  4:    27.97 21.4   6  1  0
##  5:    10.35 18.7   8  0  0
##  6:    23.66 18.1   6  1  0
##  7:     7.92 14.3   8  0  0
##  8:    35.52 24.4   4  1  0
##  9:    33.19 22.8   4  1  0
## 10:    25.10 19.2   6  1  0
## 11:    23.27 17.8   6  1  0
## 12:     9.08 16.4   8  0  0
## 13:     9.58 17.3   8  0  0
## 14:     8.42 15.2   8  0  0
## 15:     5.76 10.4   8  0  0
## 16:     5.76 10.4   8  0  0
## 17:     8.14 14.7   8  0  0
## 18:    16.31 32.4   4  1  1
## 19:    15.31 30.4   4  1  1
## 20:    17.07 33.9   4  1  1
## 21:    31.30 21.5   4  1  0
## 22:     8.58 15.5   8  0  0
## 23:     8.42 15.2   8  0  0
## 24:     7.36 13.3   8  0  0
## 25:    10.63 19.2   8  0  0
## 26:    13.75 27.3   4  1  1
## 27:   100.00 26.0   4  0  1
## 28:    15.31 30.4   4  1  1
## 29:    51.30 15.8   8  0  1
## 30:    31.93 19.7   6  0  1
## 31:    48.70 15.0   8  0  1
## 32:    10.78 21.4   4  1  1
##     mpg_rate  mpg cyl vs am
```

```
mtcars[, .(mpg_rate = round(mpg/sum(mpg) * 100, digits = 2)), by = .(cyl, vs, am)]
```

```
##     cyl vs am mpg_rate
##  1:   6  0  1    34.04
##  2:   6  0  1    34.04
##  3:   6  0  1    31.93
```

©黄湘云

```
##  4:   4  1  1    11.48
##  5:   4  1  1    16.31
##  6:   4  1  1    15.31
##  7:   4  1  1    17.07
##  8:   4  1  1    13.75
##  9:   4  1  1    15.31
## 10:   4  1  1    10.78
## 11:   6  1  0    27.97
## 12:   6  1  0    23.66
## 13:   6  1  0    25.10
## 14:   6  1  0    23.27
## 15:   8  0  0    10.35
## 16:   8  0  0     7.92
## 17:   8  0  0     9.08
## 18:   8  0  0     9.58
## 19:   8  0  0     8.42
## 20:   8  0  0     5.76
## 21:   8  0  0     5.76
## 22:   8  0  0     8.14
## 23:   8  0  0     8.58
## 24:   8  0  0     8.42
## 25:   8  0  0     7.36
## 26:   8  0  0    10.63
## 27:   4  1  0    35.52
## 28:   4  1  0    33.19
## 29:   4  1  0    31.30
## 30:   4  0  1   100.00
## 31:   8  0  1    51.30
## 32:   8  0  1    48.70
##     cyl vs am mpg_rate
```

7.2.17 去掉含有缺失值的记录

```
airquality[complete.cases(airquality), ] %>% head
```

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 7    23     299  8.6   65     5   7
## 8    19      99 13.8   59     5   8
```

```
# 或着
airquality[!apply(airquality, 1, anyNA), ] %>%  head
```

©黄海广

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 7    23     299  8.6   65     5   7
## 8    19      99 13.8   59     5   8
```

7.2.18 集合操作

match 和 `%in%` https://d.cosx.org/d/421314

```
`%nin%` <- Negate('%in%')
# `%in%` <- function(x, table) match(x, table, nomatch = 0) > 0 # %in% 函数的定义
x <- letters[1:5]
y <- letters[3:8]

x %in% y
```

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

```
x %nin% y
```

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

返回一个逻辑向量，x 中的元素匹配到了就返回 TRUE，否则 FALSE，`%nin%` 是 `%in%` 的取反效果

```
match(x,y)
```

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

x 在 y 中的匹配情况，匹配到了，就返回在 y 中匹配的位置，没有匹配到就返回 NA

```
setdiff(x, y)
```

```
## [1] "a" "b"
```

```
intersect(x, y)
```

```
## [1] "c" "d" "e"
```

```
union(x, y)
```

```
## [1] "a" "b" "c" "d" "e" "f" "g" "h"
```

7.3 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
```

© 黄湘云


```
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] magrittr_2.0.3    data.table_1.14.2
##
## loaded via a namespace (and not attached):
##  [1] knitr_1.38       sysfonts_0.8.8   tidyselect_1.1.2 R6_2.5.1
##  [5] rlang_1.0.2      fastmap_1.1.0    fansi_1.0.3      stringr_1.4.0
##  [9] dplyr_1.0.8      tools_4.1.3      broom_0.7.12     xfun_0.30
## [13] utf8_1.2.2       DBI_1.1.2        cli_3.2.0        htmltools_0.5.2
## [17] ellipsis_0.3.2   assertthat_0.2.1 yaml_2.3.5       digest_0.6.29
## [21] tibble_3.1.6     lifecycle_1.0.1  crayon_1.5.1     bookdown_0.25
## [25] tidyr_1.2.0      purrr_0.3.4      vctrs_0.4.0      curl_4.3.2
## [29] glue_1.6.2       evaluate_0.15    rmarkdown_2.13   stringi_1.7.6
## [33] compiler_4.1.3   pillar_1.7.0     backports_1.4.1  generics_0.1.2
## [37] pkgconfig_2.0.3
```

© 王建国

第八章　并行化操作

向量化运算、并行运算和分布式运算

- future 在 R 语言中提供统一的并行和分布式处理框架
- future.apply 可以替代 base R 提供的 apply 族函数
- future.batchtools 使用 batchtools 实现并行和分布式处理
- batchtools Map 函数的并行实现，用于高性能计算系统和分布式处理，可以单机多核并行也可以多机并行，还提供了一种抽象的机制去定义大规模计算机实验。
- multidplyr 是 dplyr 的后端，多核环境下实现数据分块，提高并行处理性能
- disk.frame 是基于磁盘的超出内存容量的快速并行数据操作框架
- parallelMap R package to interface some popular parallelization back-ends with a unified interface
- big.data.table 基于 data.table 的分布式并行计算

8.1　apply

apply 家族和 `do.call`

8.2　MapReduce

高阶函数，简单来说，就是参数为函数，返回值也是函数。Base R 提供了 `Reduce` 、`Filter` 、`Find` 、`Map` 、`Negate` 和 `Position` 等常用函数，此外还有 `*apply` 族。

与 `purrr::map` 比较

在 R 语言里玩转 `apply`，`Map()` 和 `Reduce()`[1]，下面分别以提取合并多张 XLSX 表格[2]，分组计算[3] 和子集操作 [4] 为例，从函数式编程到 MapReduce [5]，制作数据透视表[6]，用于数据处理的函数式编程和单元测试 Functional programming and unit testing for data munging with R 特别是第三章 https://b-rodrigues.github.io/fput/，然后是函数式编程与数据建模 Modeling data with functional programming in R[7]

[1] https://stackoverflow.com/questions/3505701/grouping-functions-tapply-by-aggregate-and-the-apply-family
[2] https://trinkerrstuff.wordpress.com/2018/02/14/easily-make-multi-tabbed-xlsx-files-with-openxlsx/
[3] https://statcompute.wordpress.com/2018/09/03/playing-map-and-reduce-in-r-by-group-calculation/
[4] https://statcompute.wordpress.com/2018/09/08/playing-map-and-reduce-in-r-subsetting/
[5] https://cartesianfaith.com/2015/09/17/from-functional-programming-to-mapreduce-in-r/
[6] https://digitheadslabnotebook.blogspot.com/2010/01/pivot-tables-in-r.html
[7] https://cartesianfaith.files.wordpress.com/2015/12/rowe-modeling-data-with-functional-programming-in-r.pdf

© 黄湘云


```
add <- function(x) Reduce("+", x)
add(list(1, 2, 3))
```

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

```
add_accuml <- function(x) Reduce("+", x, accumulate = TRUE)
add_accuml(list(1, 2, 3))
```

```
## [1] 1 3 6
```

8.3 parallel

并行计算小抄 将共享内存的 R 包整理在一起

```
library(parallel)
```

8.4 Rmpi

Rmpi 由卡尔顿大学的 Hao Yu 开发和维护

首先安装 openmpi-devel 开发环境（以 Fedora 30 为例）

```
yum install -y openmpi-devel
echo "export ORTED=/usr/lib64/openmpi/bin" >> ~/.bashrc
# 或者
echo "PATH=/usr/lib64/openmpi/bin:$PATH; export PATH" | tee -a ~/.bashrc
source ~/.bashrc
```

然后进入 R 安装 R 包 Rmpi

```
install.packages('Rmpi')
```

使用 Rmpi 包生成两组服从均匀分布的随机数

```
# 加载 R 包
library(Rmpi)
# 检测可用的逻辑 CPU 核心数
parallel::detectCores()
# 虚拟机分配四个逻辑CPU核
# 1个 master 2个 worker 主机 cloud
mpi.spawn.Rslaves(nslaves=2)
```

```
#         2 slaves are spawned successfully. 0 failed.
# master (rank 0, comm 1) of size 3 is running on: cloud
# slave1 (rank 1, comm 1) of size 3 is running on: cloud
# slave2 (rank 2, comm 1) of size 3 is running on: cloud
```

调用 `mpi.apply` 函数

©黄湘云

```
set.seed(1234)
mpi.apply(c(10, 20), runif)
```

```
[[1]]
 [1] 0.33684269 0.84638494 0.82776590 0.23707947 0.07593769 0.27981368
 [7] 0.45307675 0.02878214 0.32807421 0.92854275

[[2]]
 [1] 0.63474442 0.04025071 0.01996498 0.01922093 0.41258827 0.84150414
 [7] 0.74705002 0.07635368 0.32807392 0.94570363 0.89187667 0.67069020
[13] 0.92996997 0.22486589 0.22118236 0.15807970 0.65619450 0.16473730
[19] 0.85833484 0.11416449
```

用完要关闭

```
mpi.close.Rslaves()
```

pbdMPI 包处于活跃维护状态，是 pbdR 项目 的核心组件，能够以分布式计算的方式轻松处理 TB 级数据[8]

Rhpc 包同样基于 MPI 方式，但是集 Rmpi 和 snow 两个包的优点于一身，在保持 apply 编程风格的同时，能够提供更好的高性能计算环境，支持长向量，能够处理一些大数据。

8.5 gpuR

Charles Determan 开发的 gpuR 基于 OpenCL 加速，目前处于活跃维护状态。而 Charles Determan 开发的另一个 gpuRcuda 包是基于 CUDA 加速

赵鹏 的博客 ParallelR 关注基于 CUDA 的 GPU 加速

此外还有 gputools

```
library(gpuR)
set.seed(2019)
gpuA <- gpuMatrix(rnorm(16), nrow = 4, ncol = 4)
gpuA
```

```
An object of class "fgpuMatrix"
Slot "address":
<pointer: 0x000000000fbe9760>

Slot ".context_index":
[1] 1

Slot ".platform_index":
[1] 1

Slot ".platform":
```

[8]2016 年国际 R 语言大会上的介绍https://github.com/snoweye/user2016.demo 和 2018 年 JSM 会上的介绍 https://github.com/RBigData/R_JSM2018

© 高等教育出版社


```
[1] "Intel(R) OpenCL"

Slot ".device_index":
[1] 1

Slot ".device":
[1] "Intel(R) HD Graphics 4600"
```

```
gpuB <- gpuA %*% gpuA
print(gpuB)
```

```
Source: gpuR Matrix [4 x 4]

            [,1]      [,2]      [,3]       [,4]
[1,]  2.61787200 -1.274909 -2.150301 -2.0073860
[2,] -0.02231596  1.566433  0.986027  0.7339008
[3,] -0.12862393  1.848340  3.261899  1.6919358
[4,] -1.90084898 -1.863014 -1.312350 -0.2553876
```

8.6 运行环境

```
xfun::session_info()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Locale:
##   LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##   LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##   LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##   LC_PAPER=en_US.UTF-8       LC_NAME=C
##   LC_ADDRESS=C               LC_TELEPHONE=C
##   LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## Package version:
##   base64enc_0.1.3 bookdown_0.25   bslib_0.3.1     cli_3.2.0
##   compiler_4.1.3  curl_4.3.2      digest_0.6.29   evaluate_0.15
##   fastmap_1.1.0   fs_1.5.2        glue_1.6.2      graphics_4.1.3
##   grDevices_4.1.3 highr_0.9       htmltools_0.5.2 jquerylib_0.1.4
##   jsonlite_1.8.0  knitr_1.38      magrittr_2.0.3  methods_4.1.3
##   R6_2.5.1        rappdirs_0.3.3  rlang_1.0.2     rmarkdown_2.13
##   sass_0.4.1      stats_4.1.3     stringi_1.7.6   stringr_1.4.0
##   sysfonts_0.8.8  tinytex_0.38    tools_4.1.3     utils_4.1.3
##   xfun_0.30       yaml_2.3.5
```

create-an-empty-data-frame pipe-r

© 基础设施

© 黄湘云

第九章　净土化操作

> 常用操作和高频问题需要合并进之前的 data-manipulation，本章只介绍向量化计算以 dplyr 为核心的 tidyverse 风数据操作管道风操作

在不同规模的数据集上，Base R，dplyr 和 data.table 的处理性能应该属于低、中、高档搭配的情形

更加高级的数据变形操作，特别是数据类型的一致性，方便后续的可视化和建模，引入 tidyverse，数据处理或者叫特征工程 Base R vs data.table vs dplyr 它们各有优点，所以都加以介绍参考 Jozef Hajnala 博文。

关于 tidyverse 提供的数据操作不要移动到 Base R 对应的章节，这二者已经越行越远，本章主要讲并行或分布式数据操作工具，如 sparklyr 针对大数据集上的操

Base R 的数据操作的一致性问题参见统计之都帖子 https://d.cosx.org/d/420763

Malcolm Barrett 以幻灯片的形式呈现 dplyr 和 purrr 的基础用法

Charlotte Wickham 的课程 A introduction to purrr purrr-tutorial

关于引用 quotation

相比于 SQL，dplyr 在数据库操作的不足，这是一些比较难的部分 https://dbi.r-dbi.org/articles/dbi-1#sec:open-issues

函数式编程 Functional Programming Languages 用于数据处理

- rpivotTable 动态数据透视表
- fuzzyjoin Join tables together on inexact matching
- dtplyr dtplyr is the data.table backend for dplyr. It provides S3 methods for data.table objects so that dplyr works the way you expect.
- bplyr basic dplyr and tidyr functionality without the tidyverse dependencies
- SqlRender 基于 Java 语言，借助 rJava 包支持参数化的 SQL 语句，并且可以将一种 SQL 语句（如 Microsoft SQL Server）转化为多种 SQL 语句（如 Oracle, PostgreSQL, Amazon RedShift, Impala, IBM Netezza, Google BigQuery, Microsoft PDW, and SQLite）
- fastmap 实现键值存储，提供新的数据结构
- Roaring bitmaps Bitsets, also called bitmaps, are commonly used as fast data structures.

```
library(tidyverse)
```

数据操作的语法

第一代

1. Base R 数据操作已在第 六 章详细介绍

©黄湘云

第二代

1. reshape （退休）使用函数 `melt` 和 `cast` 重构 (restructure) 和聚合 (aggregate) 数据
2. reshape2 （退休）是 reshape 的继任者，功能和 reshape 类似，提供两个函数 `melt` 和 `cast` 聚合数据，因此不再介绍 reshape，而鉴于 reshape2 还在活跃使用中，故而以它为例介绍 `melt` 和 `cast` 函数
3. plyr（退休）统一拆分 (split)，计算 (apply)，合并 (combine) 的数据处理流，由 dplyr（用于 data.frame）和 purrr （用于 list）继任

第三代

1. dplyr 操作数据的语法及其扩展
2. sparklyr 给 dplyr 提供 Spark 接口支持
3. dbplyr 给 dplyr 提供 DBI 数据库接口支持
4. dtplyr 给 dplyr 提供 data.table 支持
5. tidyr 提供 `spread` 和 `gather` 两个函数清洗数据

Garrett Grolemund 在 RStudio 主要从事教育教学，参考 Materials for the Tidyverse Train-the-trainer workshop 和 The Tidyverse Cookbook

Dirk Eddelbuettel 的 Getting Started in R – Tinyverse Edition

9.1 常用操作

dplyr 由 Hadley Wickham 主要由开发和维护，是 Rstudio 公司开源的用于数据处理的一大利器，该包号称“数据操作的语法”，与 ggplot2 对应，也就是说数据处理那一套已经建立完整的和 SQL 一样的功能。它们都遵循同样的处理逻辑，只不过一个用 SQL 写，一个用 R 语言写，处理效率差不多，R 语言写的 SQL 会被翻译为 SQL 语句，再传至数据库查询，当然它也支持内存内的数据操作。目前 dplyr 以 dbplyr 为后端支持的数据库有：MySQL、PostgreSQL，SQLite 等，完整的支持列表请看 这里，连接特定数据库，都是基于 DBI，DBI 即 Database Interface，是使用 C/C++ 开发的底层数据库接口，是一个统一的关系型数据库连接框架，需要根据不同的具体的数据库进行实例化，才可使用。

dplyr 常用的函数是 7 个：`arrange` 排序 `filter` 过滤行 `select` 选择列 `mutate` 变换 `summarise` 汇总 `group_by` 分组 `distinct` 去重

以 ggplot2 包自带的钻石数据集 diamonds 为例介绍

9.1.1 查看

除了直接打印数据集的前几行，tibble 包还提供 glimpse 函数查看数据集，而 Base R 默认查看方式是调用 str 函数

```
diamonds
```

```
## # A tibble: 53,940 x 10
##    carat cut       color clarity depth table price     x     y     z
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
##  2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
```

©黄湘云


```
## 3 0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4 0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5 0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6 0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
## 7 0.24 Very Good I     VVS1     62.3    57   336  3.95  3.98  2.47
## 8 0.26 Very Good H     SI1      61.9    55   337  4.07  4.11  2.53
## 9 0.22 Fair      E     VS2      65.1    61   337  3.87  3.78  2.49
## 10 0.23 Very Good H     VS1      59.4    61   338  4     4.05  2.39
## # ... with 53,930 more rows
```

```
glimpse(diamonds)
```

```
## Rows: 53,940
## Columns: 10
## $ carat   <dbl> 0.23, 0.21, 0.23, 0.29, 0.31, 0.24, 0.24, 0.26, 0.22, 0.23, 0.~
## $ cut     <ord> Ideal, Premium, Good, Premium, Good, Very Good, Very Good, Ver~
## $ color   <ord> E, E, E, I, J, J, I, H, E, H, J, J, F, J, E, E, I, J, J, J, I,~
## $ clarity <ord> SI2, SI1, VS1, VS2, SI2, VVS2, VVS1, SI1, VS2, VS1, SI1, VS1, ~
## $ depth   <dbl> 61.5, 59.8, 56.9, 62.4, 63.3, 62.8, 62.3, 61.9, 65.1, 59.4, 64~
## $ table   <dbl> 55, 61, 65, 58, 58, 57, 57, 55, 61, 61, 55, 56, 61, 54, 62, 58~
## $ price   <int> 326, 326, 327, 334, 335, 336, 336, 337, 337, 338, 339, 340, 34~
## $ x       <dbl> 3.95, 3.89, 4.05, 4.20, 4.34, 3.94, 3.95, 4.07, 3.87, 4.00, 4.~
## $ y       <dbl> 3.98, 3.84, 4.07, 4.23, 4.35, 3.96, 3.98, 4.11, 3.78, 4.05, 4.~
## $ z       <dbl> 2.43, 2.31, 2.31, 2.63, 2.75, 2.48, 2.47, 2.53, 2.49, 2.39, 2.~
```

表 9.1: dplyr 定义的数据对象类型

类型	含义
int	整型 integer
dbl	（单）双精度浮点类型
chr	字符（串）类型
dttm	data-time 类型
lgl	布尔类型
fctr	因子类型 factor
date	日期类型

表 9.1 中 dttm 和 date 类型代指 lubridate 包指定的日期对象 POSIXct、POSIXlt、Date、chron、yearmon、yearqtr、zoo、zooreg、timeDate、xts、its、ti、jul、timeSeries 和 fts。

9.1.2 筛选

按条件筛选数据的子集，按行筛选

```
diamonds %>% filter(cut == "Ideal" , carat >= 3)
```

```
## # A tibble: 4 x 10
##   carat cut   color clarity depth table price     x     y     z
```

```
##   <dbl> <ord> <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  3.22 Ideal I     I1       62.6    55 12545  9.49  9.42  5.92
## 2  3.5  Ideal H     I1       62.8    57 12587  9.65  9.59  6.03
## 3  3.01 Ideal J     SI2      61.7    58 16037  9.25  9.2   5.69
## 4  3.01 Ideal J     I1       65.4    60 16538  8.99  8.93  5.86
```

先按行，再按列筛选

```
diamonds %>%
  filter(carat >= 3, color == "I") %>%
  select(cut, carat)
```

```
## # A tibble: 16 x 2
##    cut       carat
##    <ord>     <dbl>
##  1 Premium    3.01
##  2 Fair       3.02
##  3 Good       3
##  4 Ideal      3.22
##  5 Premium    4.01
##  6 Very Good  3.04
##  7 Very Good  4
##  8 Premium    3.67
##  9 Premium    3
## 10 Fair       3
## 11 Premium    3.01
## 12 Fair       3.01
## 13 Fair       3.01
## 14 Good       3.01
## 15 Good       3.01
## 16 Premium    3.04
```

9.1.3 排序

arrange 默认升序排列，按钻石重量升序，按价格降序

```
diamonds %>%
  filter(cut == "Ideal" , carat >= 3) %>%
  arrange(carat, desc(price))
```

```
## # A tibble: 4 x 10
##   carat cut   color clarity depth table price     x     y     z
##   <dbl> <ord> <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  3.01 Ideal J     I1       65.4    60 16538  8.99  8.93  5.86
## 2  3.01 Ideal J     SI2      61.7    58 16037  9.25  9.2   5.69
## 3  3.22 Ideal I     I1       62.6    55 12545  9.49  9.42  5.92
## 4  3.5  Ideal H     I1       62.8    57 12587  9.65  9.59  6.03
```

©黄湘云

9.1.4 聚合

分组求和，求平均，计数

```
diamonds %>%
  filter(carat > 3, color == "I") %>%
  group_by(cut, clarity) %>%
  summarise(sum_carat = sum(carat), mean_carat = mean(carat), n_count = n())
```

```
## # A tibble: 8 x 5
## # Groups:   cut [5]
##   cut       clarity sum_carat mean_carat n_count
##   <ord>     <ord>       <dbl>      <dbl>   <int>
## 1 Fair      I1           3.02       3.02       1
## 2 Fair      SI2          6.02       3.01       2
## 3 Good      SI2          6.02       3.01       2
## 4 Very Good I1           4          4          1
## 5 Very Good SI2          3.04       3.04       1
## 6 Premium   I1          10.7        3.56       3
## 7 Premium   SI2          6.05       3.02       2
## 8 Ideal     I1           3.22       3.22       1
```

9.1.5 合并

按行合并

```
set.seed(2018)
one <- diamonds %>%
  filter(color == "I") %>%
  sample_n(5)
two <- diamonds %>%
  filter(color == "J") %>%
  sample_n(5)
# 按行合并数据框 one 和 two
bind_rows(one, two)
```

```
## # A tibble: 10 x 10
##    carat cut       color clarity depth table price     x     y     z
##    <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
##  1  0.42 Ideal     I     VVS1     62.5  57     884  4.77  4.8   2.99
##  2  0.3  Ideal     I     VVS2     62.5  53.6   532  4.29  4.33  2.69
##  3  2.02 Good      I     VS1      57.9  63   17533  8.13  8.21  4.73
##  4  0.9  Premium   I     VS2      61.9  58    3398  6.18  6.23  3.84
##  5  1.98 Very Good I     VS2      62.7  60   15083  7.9   7.96  4.98
##  6  1.51 Very Good J     VVS2     62.6  63    8706  7.29  7.24  4.55
##  7  0.7  Very Good J     SI1      61.7  57    1979  5.65  5.69  3.5
##  8  1.16 Premium   J     VS2      62.2  59    4702  6.74  6.69  4.18
```

©黄海广

```
## 9  1.5  Premium   J     VVS2     61.8  60    8760  7.36  7.33  4.54
## 10  1.51 Premium   J     SI1      60.4  62    6680  7.42  7.32  4.45
```

按列合并

```
set.seed(2018)
three <- diamonds %>%
  select(carat, color) %>%
  sample_n(5)
four <- diamonds %>%
  select(carat, color) %>%
  sample_n(5)
bind_cols(three, four)
```

```
## # A tibble: 5 x 4
##   carat...1 color...2 carat...3 color...4
##       <dbl> <ord>         <dbl> <ord>
## 1      0.33 H              0.52 F
## 2      1.09 F              0.51 F
## 3      1.52 I              0.5  G
## 4      0.95 G              0.38 E
## 5      0.35 E              0.51 J
```

9.1.6 变换

添加一列，新的列或者改变原来的列

```
diamonds %>%
  filter(carat > 3, color == "I") %>%
  select(cut, carat) %>%
  mutate(vol = if_else(carat > 3.5, "A", "B"))
```

```
## # A tibble: 13 x 3
##    cut       carat vol
##    <ord>     <dbl> <chr>
##  1 Premium    3.01 B
##  2 Fair       3.02 B
##  3 Ideal      3.22 B
##  4 Premium    4.01 A
##  5 Very Good  3.04 B
##  6 Very Good  4    A
##  7 Premium    3.67 A
##  8 Premium    3.01 B
##  9 Fair       3.01 B
## 10 Fair       3.01 B
## 11 Good       3.01 B
## 12 Good       3.01 B
```

©黄湘云


```
## 13 Premium    3.04 B
```

9.1.7 去重

数据去重在 dplyr 中的实现[1]。

```
set.seed(123)
df <- data.frame(
  x = sample(0:1, 10, replace = T),
  y = sample(0:1, 10, replace = T),
  z = 1:10
)
df
```

```
##    x y  z
## 1  0 1  1
## 2  0 1  2
## 3  0 1  3
## 4  1 0  4
## 5  0 1  5
## 6  1 0  6
## 7  1 1  7
## 8  1 0  8
## 9  0 0  9
## 10 0 0 10
```

去掉列重复的数据点 (x, y)

```
df %>%
  group_by(x, y) %>%
  filter(row_number(z) == 1)
```

```
## # A tibble: 4 x 3
## # Groups:   x, y [4]
##       x     y     z
##   <int> <int> <int>
## 1     0     1     1
## 2     1     0     4
## 3     1     1     7
## 4     0     0     9
```

```
# 此处不对，没有了 z
df %>%
  distinct(x, y)
```

```
##   x y
## 1 0 1
```

[1] https://stackoverflow.com/questions/22959635/

© 王建平


```
## 2 1 0
## 3 1 1
## 4 0 0
```

```
# 应该为
df %>%
  distinct(x, y, .keep_all = TRUE)
```

```
##   x y z
## 1 0 1 1
## 2 1 0 4
## 3 1 1 7
## 4 0 0 9
```

9.2 高频问题

常用的数据操作包含

1. 创建空的数据框或者说初始化一个数据框，
2. 按指定的列对数据框排序，
3. 选择特定的一些列，复杂情况是可能需要正则表达式从列名或者值中筛选
4. 合并两个数据框，分为 (inner outer left right) 四种情况
5. 宽格式和长格式互相转换，即重塑操作 reshape，单独的 tidyr 包操作，是 reshape2 包的进化版，提供 `spread` 和 `gather` 两个主要函数

9.2.1 初始化数据框

创建空的数据框，就是不包含任何行、记录

```
empty_df <- data.frame(
  Doubles = double(),
  Ints = integer(),
  Factors = factor(),
  Logicals = logical(),
  Characters = character(),
  stringsAsFactors = FALSE
)
str(empty_df)
```

```
## 'data.frame':    0 obs. of  5 variables:
##  $ Doubles   : num
##  $ Ints      : int
##  $ Factors   : Factor w/ 0 levels:
##  $ Logicals  : logi
##  $ Characters: chr
```

如果数据框 df 包含数据，现在要依据它创建一个空的数据框

©黄湘云


```
empty_df = df[FALSE,]
```

还可以使用 structure 构造一个数据框，并且我们发现它的效率更高

```
s <- function() structure(list(
    Date = as.Date(character()),
    File = character(),
    User = character()
  ),
  class = "data.frame"
  )
d <- function() data.frame(
    Date = as.Date(character()),
    File = character(),
    User = character(),
    stringsAsFactors = FALSE
  )
microbenchmark::microbenchmark(s(), d())
```

```
## Unit: microseconds
##   expr   min     lq    mean median     uq    max neval
##    s()  27.4  32.20  84.926  43.35  47.50 3776.8   100
##    d() 294.3 323.45 391.996 343.10 383.65 3556.9   100
```

9.2.2 移除缺失记录

只要行中包含缺失值，我们就把这样的行移除出去

```
airquality[complete.cases(airquality), ]
```

```
##     Ozone Solar.R Wind Temp Month Day
## 1      41     190  7.4   67     5   1
## 2      36     118  8.0   72     5   2
## 3      12     149 12.6   74     5   3
## 4      18     313 11.5   62     5   4
## 7      23     299  8.6   65     5   7
## 8      19      99 13.8   59     5   8
## 9       8      19 20.1   61     5   9
## 12     16     256  9.7   69     5  12
## 13     11     290  9.2   66     5  13
## 14     14     274 10.9   68     5  14
## 15     18      65 13.2   58     5  15
## 16     14     334 11.5   64     5  16
## 17     34     307 12.0   66     5  17
## 18      6      78 18.4   57     5  18
## 19     30     322 11.5   68     5  19
## 20     11      44  9.7   62     5  20
```

©黄海洋

```
## 21      1       8  9.7   59     5  21
## 22     11     320 16.6   73     5  22
## 23      4      25  9.7   61     5  23
## 24     32      92 12.0   61     5  24
## 28     23      13 12.0   67     5  28
## 29     45     252 14.9   81     5  29
## 30    115     223  5.7   79     5  30
## 31     37     279  7.4   76     5  31
## 38     29     127  9.7   82     6   7
## 40     71     291 13.8   90     6   9
## 41     39     323 11.5   87     6  10
## 44     23     148  8.0   82     6  13
## 47     21     191 14.9   77     6  16
## 48     37     284 20.7   72     6  17
## 49     20      37  9.2   65     6  18
## 50     12     120 11.5   73     6  19
## 51     13     137 10.3   76     6  20
## 62    135     269  4.1   84     7   1
## 63     49     248  9.2   85     7   2
## 64     32     236  9.2   81     7   3
## 66     64     175  4.6   83     7   5
## 67     40     314 10.9   83     7   6
## 68     77     276  5.1   88     7   7
## 69     97     267  6.3   92     7   8
## 70     97     272  5.7   92     7   9
## 71     85     175  7.4   89     7  10
## 73     10     264 14.3   73     7  12
## 74     27     175 14.9   81     7  13
## 76      7      48 14.3   80     7  15
## 77     48     260  6.9   81     7  16
## 78     35     274 10.3   82     7  17
## 79     61     285  6.3   84     7  18
## 80     79     187  5.1   87     7  19
## 81     63     220 11.5   85     7  20
## 82     16       7  6.9   74     7  21
## 85     80     294  8.6   86     7  24
## 86    108     223  8.0   85     7  25
## 87     20      81  8.6   82     7  26
## 88     52      82 12.0   86     7  27
## 89     82     213  7.4   88     7  28
## 90     50     275  7.4   86     7  29
## 91     64     253  7.4   83     7  30
## 92     59     254  9.2   81     7  31
## 93     39      83  6.9   81     8   1
## 94      9      24 13.8   81     8   2
```

©黄海广

```
## 95     16      77  7.4   82     8   3
## 99    122     255  4.0   89     8   7
## 100    89     229 10.3   90     8   8
## 101   110     207  8.0   90     8   9
## 104    44     192 11.5   86     8  12
## 105    28     273 11.5   82     8  13
## 106    65     157  9.7   80     8  14
## 108    22      71 10.3   77     8  16
## 109    59      51  6.3   79     8  17
## 110    23     115  7.4   76     8  18
## 111    31     244 10.9   78     8  19
## 112    44     190 10.3   78     8  20
## 113    21     259 15.5   77     8  21
## 114     9      36 14.3   72     8  22
## 116    45     212  9.7   79     8  24
## 117   168     238  3.4   81     8  25
## 118    73     215  8.0   86     8  26
## 120    76     203  9.7   97     8  28
## 121   118     225  2.3   94     8  29
## 122    84     237  6.3   96     8  30
## 123    85     188  6.3   94     8  31
## 124    96     167  6.9   91     9   1
## 125    78     197  5.1   92     9   2
## 126    73     183  2.8   93     9   3
## 127    91     189  4.6   93     9   4
## 128    47      95  7.4   87     9   5
## 129    32      92 15.5   84     9   6
## 130    20     252 10.9   80     9   7
## 131    23     220 10.3   78     9   8
## 132    21     230 10.9   75     9   9
## 133    24     259  9.7   73     9  10
## 134    44     236 14.9   81     9  11
## 135    21     259 15.5   76     9  12
## 136    28     238  6.3   77     9  13
## 137     9      24 10.9   71     9  14
## 138    13     112 11.5   71     9  15
## 139    46     237  6.9   78     9  16
## 140    18     224 13.8   67     9  17
## 141    13      27 10.3   76     9  18
## 142    24     238 10.3   68     9  19
## 143    16     201  8.0   82     9  20
## 144    13     238 12.6   64     9  21
## 145    23      14  9.2   71     9  22
## 146    36     139 10.3   81     9  23
## 147     7      49 10.3   69     9  24
```

©黄海洋

```
## 148    14      20 16.6   63     9  25
## 149    30     193  6.9   70     9  26
## 151    14     191 14.3   75     9  28
## 152    18     131  8.0   76     9  29
## 153    20     223 11.5   68     9  30
```

9.2.3 数据类型转化

```
str(PlantGrowth)
```

```
## 'data.frame':    30 obs. of  2 variables:
##  $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
##  $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
bob <- PlantGrowth
i <- sapply(bob, is.factor)
bob[i] <- lapply(bob[i], as.character)
str(bob)
```

```
## 'data.frame':    30 obs. of  2 variables:
##  $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
##  $ group : chr  "ctrl" "ctrl" "ctrl" "ctrl" ...
```

9.2.4 跨列分组求和

输入是一个数据框 data.frame，按照其中某一变量分组，然后计算任意数量的变量的行和和列和。

空气质量数据集 airquality 按月份 Month 分组，然后求取满足条件的列的和

```
Reduce(rbind, lapply(unique(airquality$Month), function(gv) {
  subdta <- subset(airquality, subset = Month == gv)
  data.frame(
    Colsum = as.numeric(
      colSums(subdta[, grepl("[mM]", names(airquality))], na.rm = TRUE)
    ),
    Month = gv
  )
}))
```

```
##    Colsum Month
## 1    2032     5
## 2     155     5
## 3    2373     6
## 4     180     6
## 5    2601     7
## 6     217     7
## 7    2603     8
```

© 黄湘云


```
## 8     248     8
## 9    2307     9
## 10    270     9
```

什么是函数式编程，R 语言环境下的函数式编程是如何操作的

9.3 管道操作

Stefan Milton Bache 开发了 magrittr 包实现管道操作，增加代码的可读性和维护性，但是这个 R 包的名字取的太奇葩，因为 记不住，它其实是一个复杂的法语发音，中式英语就叫它马格里特吧！这下应该好记多了吧！

我要查看是否需要新添加一个 R 包依赖，假设该 R 包是 reticulate 没有出现在 DESCRIPTION 文件中，但是可能已经被其中某（个）些 R 包依赖了

```
"reticulate" %in% sort(unique(unlist(tools::package_dependencies(desc::desc_get_deps()$package, recu
```

```
## [1] TRUE
```

安装 pkg 的依赖

```
pkg <- c(
  "bookdown",
  "e1071",
  "formatR",
  "lme4",
  "mvtnorm",
  "prettydoc", "psych",
  "reticulate", "rstan", "rstanarm", "rticles",
  "svglite",
  "TMB", "glmmTMB"
)
# 获取 pkg 的所有依赖
dep_pkg <- tools::package_dependencies(pkg, recursive = TRUE)
# 将列表 list 合并为向量 vector
merge_pkg <- Reduce("c", dep_pkg, accumulate = FALSE)
# 所有未安装的 R 包
miss_pkg <- setdiff(unique(merge_pkg), unique(.packages(TRUE)))
# 除了 pkg 外，未安装的 R 包，安装 pkg 的依赖
sort(setdiff(miss_pkg, pkg))
```

```
## [1] "mnormt"  "tmvnsim"
```

转化为管道操作，增加可读性

再举一个关于数据模拟的例子

模拟 0-1 序列，

©黄文海

```
set.seed(2019)
binom_sample <- function(n) {
  sum(sample(x = c(0,1), size = n, prob = c(0.8, 0.2), replace = TRUE))/n
}
# 频率估计概率
one_prob <- sapply(10^(seq(8)), binom_sample)
# 估计的误差
one_abs <- abs(one_prob - 0.2)
one_abs
```

```
## [1] 1.000e-01 1.000e-02 1.100e-02 4.400e-03 1.460e-03 3.980e-04 4.700e-06
## [8] 9.552e-05
```

似然估计

9.4 运行环境

```
xfun::session_info()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Locale:
##   LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##   LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##   LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##   LC_PAPER=en_US.UTF-8       LC_NAME=C
##   LC_ADDRESS=C               LC_TELEPHONE=C
##   LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## Package version:
##   askpass_1.1          assertthat_0.2.1     backports_1.4.1
##   base64enc_0.1.3      bit_4.0.4            bit64_4.0.5
##   blob_1.2.2           bookdown_0.25        broom_0.7.12
##   bslib_0.3.1          callr_3.7.0          cellranger_1.1.0
##   cli_3.2.0            clipr_0.8.0          colorspace_2.0-3
##   compiler_4.1.3       cpp11_0.4.2          crayon_1.5.1
##   curl_4.3.2           data.table_1.14.2    DBI_1.1.2
##   dbplyr_2.1.1         desc_1.4.1           digest_0.6.29
##   dplyr_1.0.8          dtplyr_1.2.1         ellipsis_0.3.2
##   evaluate_0.15        fansi_1.0.3          farver_2.1.0
##   fastmap_1.1.0        forcats_0.5.1        fs_1.5.2
##   gargle_1.2.0         generics_0.1.2       ggplot2_3.3.5
##   glue_1.6.2           googledrive_2.0.0    googlesheets4_1.0.0
```

© 黄湘云


```
##   graphics_4.1.3       grDevices_4.1.3      grid_4.1.3
##   gtable_0.3.0         haven_2.4.3          highr_0.9
##   hms_1.1.1            htmltools_0.5.2      httr_1.4.2
##   ids_1.0.1            isoband_0.2.5        jquerylib_0.1.4
##   jsonlite_1.8.0       knitr_1.38           labeling_0.4.2
##   lattice_0.20.45      lifecycle_1.0.1      lubridate_1.8.0
##   magrittr_2.0.3       MASS_7.3.56          Matrix_1.4.1
##   methods_4.1.3        mgcv_1.8.40          microbenchmark_1.4.9
##   mime_0.12            modelr_0.1.8         munsell_0.5.0
##   nlme_3.1.157         openssl_2.0.0        pillar_1.7.0
##   pkgconfig_2.0.3      prettyunits_1.1.1    processx_3.5.3
##   progress_1.2.2       ps_1.6.0             purrr_0.3.4
##   R6_2.5.1             rappdirs_0.3.3       RColorBrewer_1.1.2
##   readr_2.1.2          readxl_1.4.0         rematch_1.0.1
##   rematch2_2.1.2       reprex_2.0.1         rlang_1.0.2
##   rmarkdown_2.13       rprojroot_2.0.2      rstudioapi_0.13
##   rvest_1.0.2          sass_0.4.1           scales_1.1.1
##   selectr_0.4.2        splines_4.1.3        stats_4.1.3
##   stringi_1.7.6        stringr_1.4.0        sys_3.4
##   sysfonts_0.8.8       tibble_3.1.6         tidyr_1.2.0
##   tidyselect_1.1.2     tidyverse_1.3.1      tinytex_0.38
##   tools_4.1.3          tzdb_0.3.0           utf8_1.2.2
##   utils_4.1.3          uuid_1.0.4           vctrs_0.4.0
##   viridisLite_0.4.0    vroom_1.5.7          withr_2.5.0
##   xfun_0.30            xml2_1.3.3           yaml_2.3.5
```

© 黄湘云

第二部分

统计图形

© 黄湘云

介绍

统计图形

© 王湘君

第十章　图形基础

```
library(survival)
library(lattice)
library(nlme)
library(MASS)
library(RColorBrewer)
library(latticeExtra)
library(shape)
library(splines)
library(mgcv)
library(maps)
library(mapproj)
```

数据可视化是一种重要的数据分析手段，R 提供了两套图形系统，分别是 graphics 包提供的基础绘图系统和 grid 包提供的栅格绘图系统，后者主要以两个 R 包为大家所熟知，一个是 lattice 包，另一个是 ggplot2 包。

Base 图形系统的扩展包 basetheme 可以设置主题，prettyB 和 gridGraphics

为了方便记忆函数 `par` 的各个参数，Paul Murrell 整理了一份 助记符，此外，LaTeX 宏包 geometry 对版面设置有很多专业的说明

10.1　绘图基本要素

10.1.1　点线

点和线是最常见的画图元素，在 `plot` 函数中，分别用参数 `pch` 和 `lty` 来设定类型，点的大小、线的宽度分别用参数 `cex` 和 `lwd` 来指定，颜色由参数 `col` 设置。参数 `type` 不同的值设置如下，`p` 显示点，`l` 绘制线，`b` 同时绘制空心点，并用线连接，`c` 只有线，`o` 在线上绘制点，`s` 和 `S` 点线连接绘制阶梯图，`h` 绘制类似直方图一样的垂线，最后 `n` 表示什么也不画。

点 points 、线 grid 背景线 abline lines rug 刻度线（线段 segments、箭头 arrows）、

```
## -------- Showing all the extra & some char graphics symbols ---------
pchShow <-
  function(extras = c("*", ".", "o", "O", "0", "+", "-", "|", "%", "#"),
           cex = 2, ## good for both .Device=="postscript" and "x11"
           col = "red3", bg = "gold", coltext = "brown", cextext = 1.2,
```

© 高等教育出版社


```
            main = paste(
              "plot symbols :  points (...  pch = *, cex =",
              cex, ")"
            )) {
    nex <- length(extras)
    np <- 26 + nex
    ipch <- 0:(np - 1)
    k <- floor(sqrt(np))
    dd <- c(-1, 1) / 2
    rx <- dd + range(ix <- ipch %/% k)
    ry <- dd + range(iy <- 3 + (k - 1) - ipch %% k)
    pch <- as.list(ipch) # list with integers & strings
    if (nex > 0) pch[26 + 1:nex] <- as.list(extras)
    plot(rx, ry, type = "n", axes = FALSE, xlab = "", ylab = "", main = main)
    abline(v = ix, h = iy, col = "lightgray", lty = "dotted")
    for (i in 1:np) {
      pc <- pch[[i]]
      ## 'col' symbols with a 'bg'-colored interior (where available) :
      points(ix[i], iy[i], pch = pc, col = col, bg = bg, cex = cex)
      if (cextext > 0) {
        text(ix[i] - 0.3, iy[i], pc, col = coltext, cex = cextext)
      }
    }
  }

pchShow()
```

```
## ------------ test code for various pch specifications -------------
# Try this in various font families (including Hershey)
# and locales.  Use sign = -1 asserts we want Latin-1.
# Standard cases in a MBCS locale will not plot the top half.
TestChars <- function(sign = 1, font = 1, ...) {
  MB <- l10n_info()$MBCS
  r <- if (font == 5) {
    sign <- 1
    c(32:126, 160:254)
  } else if (MB) 32:126 else 32:255
  if (sign == -1) r <- c(32:126, 160:255)
  par(pty = "s")
  plot(c(-1, 16), c(-1, 16),
    type = "n", xlab = "", ylab = "",
    xaxs = "i", yaxs = "i",
    main = sprintf("sign = %d, font = %d", sign, font)
  )
  grid(17, 17, lty = 1)
```

© 王建湘

图 10.1: 不同的 pch 参数值

```
  mtext(paste("MBCS:", MB))
  for (i in r) try(points(i %% 16, i %/% 16, pch = sign * i, font = font, ...))
}
TestChars()
```

```
try(TestChars(sign = -1))
```

```
TestChars(font = 5) # Euro might be at 160 (0+10*16).
```

```
# macOS has apple at 240 (0+15*16).
try(TestChars(-1, font = 2)) # bold
```

```
x <- 0:12
y <- sin(pi / 5 * x)
par(mfrow = c(3, 3), mar = .1 + c(2, 2, 3, 1))
for (tp in c("p", "l", "b", "c", "o", "h", "s", "S", "n")) {
  plot(y ~ x, type = tp, main = paste0("plot(*, type = \"", tp, "\")"))
  if (tp == "S") {
    lines(x, y, type = "s", col = "red", lty = 2)
    mtext("lines(*, type = \"s\", ...)", col = "red", cex = 0.8)
  }
}
```

颜色 col 连续型和离散型

线帽/端和字体的样式

© 黄湘云

图 10.2: pch 支持的字符

图 10.3: pch 支持的字符

© 黄湘云

图 10.4: pch 支持的字符

图 10.5: pch 支持的字符

© 黄湘云

图 10.6: 不同的 type 参数值

© 基础

```r
# 合并为一个图 三条粗横线 横线上三种字形
plot(c(1, 20), c(1, 20), type = "n", ann = FALSE)
lines(x = c(5, 15), y = c(5, 5), lwd = 15, lend = "round")
text(10, 5, "Hello, Helvetica", cex = 1.5, family = "sans", pos = 1, offset = 1.5)
text(5, 5, "sans", cex = 1.5, family = "sans", pos = 2, offset = .5)
text(15, 5, "lend = round", pos = 4, offset = .5)

lines(x = c(5, 15), y = c(10, 10), lwd = 15, lend = "butt")
text(10, 10, "Hello, Helvetica", cex = 1.5, family = "mono", pos = 1, offset = 1.5)
text(5, 10, "mono", cex = 1.5, family = "mono", pos = 2, offset = .5)
text(15, 10, "lend = butt", pos = 4, offset = .5)

lines(x = c(5, 15), y = c(15, 15), lwd = 15, lend = "square")
text(10, 15, "Hello, Helvetica", cex = 1.5, family = "serif", pos = 1, offset = 1.5)
text(5, 15, "serif", cex = 1.5, family = "serif", pos = 2, offset = .5)
text(15, 15, "lend = square", pos = 4, offset = .5)
```

图 10.7: 不同的线端样式

lend：线端的样式，可用一个整数或字符串指定：

- 0 或 “round” 圆形（默认）
- 1 或 “butt” 对接形
- 2 或 “square” 方形

10.1.2 区域

矩形，多边形，曲线交汇出来的区域面（矩形 rect，多边形 polygon）、路径 polypath 面/多边形 rect 颜色填充

© 王建平


```
# From the manual
ch.col <- c(
  "rainbow(n, start=.7, end=.1)",
  "heat.colors(n)",
  "terrain.colors(n)",
  "topo.colors(n)",
  "cm.colors(n)"
) # 选择颜色
n <- 16
nt <- length(ch.col)
i <- 1:n
j <- n / nt
d <- j / 6
dy <- 2 * d
plot(i, i + d,
  type = "n",
  yaxt = "n",
  ylab = "",
  xlab = "",
  main = paste("color palettes; n=", n)
)
for (k in 1:nt) {
  rect(i - .5, (k - 1) * j + dy, i + .4, k * j,
    col = eval(parse(text = ch.col[k]))
  ) # 咬人的函数/字符串解析为/转函数
  text(2 * j, k * j + dy / 4, ch.col[k])
}
```

`clip(x1, x2, y1, y2)` 在用户坐标中设置剪切区域

```
x <- rnorm(1000)
hist(x, xlim = c(-4, 4))
usr <- par("usr")
clip(usr[1], -2, usr[3], usr[4])
hist(x, col = "red", add = TRUE)
clip(2, usr[2], usr[3], usr[4])
hist(x, col = "blue", add = TRUE)
```

© 黄湘云

图 10.8: rect 函数画长方形

© 黄湘云

```
do.call("clip", as.list(usr)) # reset to plot region
```

```
my.col <- function(f, g, xmin, xmax, col, N = 200,
                   xlab = "", ylab = "", main = "") {
  x <- seq(xmin, xmax, length = N)
  fx <- f(x)
  gx <- g(x)
  plot(0, 0,
    type = "n",
    xlim = c(xmin, xmax),
    ylim = c(min(fx, gx), max(fx, gx)),
    xlab = xlab, ylab = ylab, main = main
  )
  polygon(c(x, rev(x)), c(fx, rev(gx)),
    col = "#EA4335", border = 0
  )
  lines(x, fx, lwd = 3, col = "#34A853")
  lines(x, gx, lwd = 3, col = "#4285f4")
}
my.col(function(x) x^2, function(x) x^2 + 10 * sin(x),
  -6, 6,
  main = "The \"polygon\" function"
)
```

各种符号 10.10

© 黄湘云

图 10.9: 区域重叠 polygon 函数

```
plot(0, 0,
  xlim = c(1, 5), ylim = c(-.5, 4),
  axes = F,
  xlab = "", ylab = ""
)
for (i in 0:4) {
  for (j in 1:5) {
    n <- 5 * i + j
    points(j, i,
      pch = n,
      cex = 3
    )
    text(j, i - .3, as.character(n))
  }
}
```

点、线、多边形和圆聚集在图 10.11 中

```
# https://jeroen.github.io/uros2018/#23
plot.new()
plot.window(xlim = c(0, 100), ylim = c(0, 100))
polygon(c(10, 40, 80), c(10, 80, 40), col = "hotpink")
text(40, 90, labels = "My drawing", col = "navyblue", cex = 3)
symbols(c(70, 80, 90), c(20, 50, 80),
  circles = c(10, 20, 10),
  bg = c("#4285f4", "#EA4335", "red"), add = TRUE, lty = "dashed"
)
```

© 黄湘云

图 10.10: cex 支持的符号

图 10.11: 多边形和符号元素

© 基础统计

在介绍各种统计图形之前，先介绍几个绘图函数 plot 和 text 还有 par 参数设置，作为最简单的开始，尽量依次介绍其中的每个参数的含义并附上图形对比。

```
y <- x <- 1:4
plot(x, y, ann = F, col = "blue", pch = 16)
text(x, y,
  labels = c("1st", "2nd", "3rd", "4th"),
  col = "red", pos = c(3, 4, 4, 1), offset = 0.6
)
ahat <- "sigma"
# title(substitute(hat(a) == ahat, list(ahat = ahat)))
title(bquote(hat(a) == .(ahat)))
```

图 10.12: pos 位置参数

其中 labels，pos 都是向量化的参数

10.1.3 参考线

矩形网格线是用做背景参考线的，常常是淡灰色的细密虚线，plot 函数的 panel.first 参数和 grid 函数常用来画这种参考线

```
# modified from https://yihui.name/cn/2018/02/cohen-s-d/
n <- 30 # 样本量（只是一个例子）
x <- seq(0, 12, 0.01)
par(mar = c(4, 4, 0.2, 0.1))
plot(x / sqrt(n), 2 * (1 - pt(x, n - 1)),
  xlab = expression(d = x / sqrt(n)),
  type = "l", panel.first = grid()
```

© 黄湘云


```
)
abline(v = c(0.01, 0.2, 0.5, 0.8, 1.2, 2), lty = 2)
```

图 10.13: 添加背景参考线

10.1.4 坐标轴

图形控制参数默认设置下 par 通常的一幅图形，改变坐标轴标签是很简单的

```
x <- 1:100
y <- runif(100, -2, 2)
plot(x, y)
```

© 黄湘云

```
plot(x, y, xlab = "Index", ylab = "Uniform draws")
```

改变坐标轴标签和标题

© 黄湘云


```
op <- par(no.readonly = TRUE) # 保存默认的 par 设置
par(cex.lab = 1.5, cex.axis = 1.3)
plot(x, y, xlab = "Index", ylab = "Uniform draws")

# 设置更大的坐标轴标签内容
par(mar = c(6, 6, 3, 3), cex.axis = 1.5, cex.lab = 2)
plot(x, y, xlab = "Index", ylab = "Uniform draws")
```

使用 axis 函数可以更加精细地控制坐标轴

```
par(op) # 恢复默认的 par 设置
plot(x, y, xaxt = "n") # 去掉 x 轴
axis(side = 1, at = c(5, 50, 100)) # 添加指定的刻度标签
```

© 黄湘云

指定刻度标签的内容

```
plot(x, y, yaxt = "n")
axis(side = 2, at = c(-2, 0, 2), labels = c("Small", "Medium", "Big"))
```

© 黄湘云

控制刻度线和轴线和刻度标签

```
plot(x, y)
axis(side = 3, at = c(5, 25, 75), lwd = 4, lwd.ticks = 2, col.ticks = "red")
```

© 黄湘云

还可以把 box 移除，绘图区域的边框去掉，只保留坐标轴

```
plot(x, y, bty = "n", xaxt = "n", yaxt = "n")
axis(side = 1, at = seq(0, 100, 20), lwd = 3)
axis(side = 2, at = seq(-2, 2, 2), lwd = 3)
```

© 黄湘云

```
# 双Y轴
N <- 200
x <- seq(-4, 4, length = N)
y1 <- sin(x)
y2 <- cos(x)
op <- par(mar = c(5, 4, 4, 4)) # Add some space in the right margin
# The default is c(5,4,4,2) + .1
xlim <- range(x)
ylim <- c(-1.1, 1.1)
plot(x, y1,
  col = "blue", type = "l",
  xlim = xlim, ylim = ylim,
  axes = F, xlab = "", ylab = "", main = "Title"
)
axis(1)
axis(2, col = "blue")
par(new = TRUE)
plot(x, y2,
  col = "red", type = "l",
  xlim = xlim, ylim = ylim,
  axes = F, xlab = "", ylab = "", main = ""
)
axis(4, col = "red")
mtext("First Y axis", 2, line = 2, col = "blue", cex = 1.2)
```

© 黄湘云


```
mtext("Second Y axis", 4, line = 2, col = "red", cex = 1.2)
```

图 10.14: 两个 Y 轴

```
# 1,2,3,4 分别代表下左上右四个位置
```

调整坐标轴标签的距离

```
## Changing default gap between labels:
plot(c(0, 100), c(0, 50), type = "n", axes = FALSE, ann = FALSE)
title(quote("axis(1, .., gap.axis = f)," ~ ~ f >= 0))
axis(2, at = 5 * (0:10), las = 1, gap.axis = 1 / 4)
gaps <- c(4, 2, 1, 1 / 2, 1 / 4, 0.1, 0)
chG <- paste0(
  ifelse(gaps == 1, "default:  ", ""),
  "gap.axis=", formatC(gaps)
)
jj <- seq_along(gaps)
linG <- -2.5 * (jj - 1)
for (j in jj) {
  isD <- gaps[j] == 1 # is default
  axis(1,
    at = 5 * (0:20), gap.axis = gaps[j], padj = -1, line = linG[j],
    col.axis = if (isD) "forest green" else 1, font.axis = 1 + isD
  )
}
```

© 统计学习与R语言


```
mtext(chG,
  side = 1, padj = -1, line = linG - 1 / 2, cex = 3 / 4,
  col = ifelse(gaps == 1, "forest green", "blue3")
)
```

图 10.15: gap.axis 用法

```
## now shrink the window (in x- and y-direction) and observe the axis labels drawn
```

旋转坐标轴标签

```
# Rotated axis labels in R plots
# https://menugget.blogspot.com/2014/08/rotated-axis-labels-in-r-plots.html
# Example data
tmin <- as.Date("2000-01-01")
tmax <- as.Date("2001-01-01")
tlab <- seq(tmin, tmax, by = "month")
lab <- format(tlab, format = "%Y-%b")
set.seed(111)
x <- seq(tmin, tmax, length.out = 100)
y <- cumsum(rnorm(100))

# Plot
# png("plot_w_rotated_axis_labels.png", height = 3,
#     width = 6, units = "in", res = 300)
op <- par(mar = c(6, 4, 1, 1))
plot(x, y, t = "l", xaxt = "n", xlab = "")
axis(1, at = tlab, labels = FALSE)
text(
```

© 基础图形


```
  x = tlab, y = par()$usr[3] - 0.1 * (par()$usr[4] - par()$usr[3]),
  labels = lab, srt = 45, adj = 1, xpd = TRUE
)
```

```
par(op)
# dev.off()
```

旋转坐标抽标签的例子来自手册《R FAQ》的第 7 章第 27 个问题 [Hornik, 2020]，在基础图形中，旋转坐标轴标签需要 `text()` 而不是 `mtext()`，因为后者不支持 `par("srt")`

```
## Increase bottom margin to make room for rotated labels
par(mar = c(5, 4, .5, 2) + 0.1)
## Create plot with no x axis and no x axis label
plot(1:8, xaxt = "n", xlab = "")
## Set up x axis with tick marks alone
axis(1, labels = FALSE)
## Create some text labels
labels <- paste("Label", 1:8, sep = " ")
## Plot x axis labels at default tick marks
text(1:8, par("usr")[3] - 0.5,
  srt = 45, adj = 1,
  labels = labels, xpd = TRUE
)
## Plot x axis label at line 6 (of 7)
```

© 黄湘云


```
mtext(side = 1, text = "X Axis Label", line = 4)
```

图 10.16: 旋转坐标轴标签

`srt = 45` 表示文本旋转角度，`xpd = TRUE` 允许文本越出绘图区域，`adj = 1` to place the right end of text at the tick marks；You can adjust the value of the 0.5 offset as required to move the axis labels up or down relative to the x axis. 详细地参考 [Murrell, 2003]

10.1.5 刻度线

通过 `par` 或 `axis` 函数实现刻度线的精细操作，tcl 控制刻度线的长度，正值让刻度画在绘图区域内，负值正好相反，画在外面，mgp 参数有三个值，第一个值控制绘图区域和坐标轴标题之间的行数，第二个是绘图区域与坐标轴标签的行数，第三个绘图区域与轴线的行数，行数表示间距

```
par(tcl = 0.4, mgp = c(1.5, 0, 0))
plot(x, y)
```

© 黄湘云

```
# 又一个例子
par(op)
plot(x, y, xaxt = "n", yaxt = "n", xlab = "", ylab = "")
axis(side = 1, at = seq(5, 95, 30), tcl = 0.4, lwd.ticks = 3, mgp = c(0, 0.5, 0))
mtext(side = 1, text = "X axis", line = 1.5)
# mtext 设置坐标轴标签
axis(side = 2, at = seq(-2, 2, 2), tcl = 0.3, lwd.ticks = 3, col.ticks = "orange", mgp = c(0, 0, 2))
mtext(side = 2, text = "Numbers taken randomly", line = 2.2)
```

© 黄湘云

10.1.6 标题

添加多个标题

```
N <- 200
x <- runif(N, -4, 4)
y <- sin(x) + .5 * rnorm(N)
plot(x, y, xlab = "", ylab = "", main = "")
mtext("Subtitle", 3, line = .8)
mtext("Title", 3, line = 2, cex = 1.5)
mtext("X axis", 1, line = 2.5, cex = 1.5)
mtext("X axis subtitle", 1, line = 3.7)
```

10.1.7 注释

数学符号注释，图10.18 自定义坐标轴 [Murrell and Ihaka, 2000]。

```
# 自定义坐标轴
plot(c(1, 1e6), c(-pi, pi),
  type = "n",
  axes = FALSE, ann = FALSE, log = "x"
)
axis(1,
  at = c(1, 1e2, 1e4, 1e6),
```

© 黄湘云

图 10.17: 图标题/子标题 x 轴标题/子标题

```
  labels = expression(1, 10^2, 10^4, 10^6)
)
axis(2,
  at = c(-pi, -pi / 2, 0, pi / 2, pi),
  labels = expression(-pi, -pi / 2, 0, pi / 2, pi)
)
text(1e3, 0, expression(italic("Customized Axes")))
box()
```

在标题中添加数学公式

```
x <- seq(-5, 5, length = 200)
y <- sqrt(1 + x^2)
plot(y ~ x,
  type = "l",
  ylab = expression(sqrt(1 + x^2))
)
title(main = expression(
  "graph of the function f"(x) == sqrt(1 + x^2)
))
```

修改参数使用 substitute 函数批量生成

© 黄湘云

图 10.18: 创建自定义的坐标轴和刻度标签

图 10.19: 标题含有数学公式

© 基础统计


```
x <- seq(-5, 5, length = 200)
for (i in 1:4) { # 画四个图
  y <- sqrt(i + x^2)
  plot(y ~ x,
    type = "l",
    ylim = c(0, 6),
    ylab = substitute(
      expression(sqrt(i + x^2)),
      list(i = i)
    )
  )
  title(main = substitute(
    "graph of the function f"(x) == sqrt(i + x^2),
    list(i = i)
  ))
}
```

图 10.20: 批量生成函数图形

基础绘图函数，如 plot 标签 `xlab` 支持 Unicode 代码表示的希腊字母，常用字母表备查，公式环境下，也可以用在绘图中

表 10.1: 希腊字母表

希腊字母	LaTeX 代码	Unicode 代码	希腊字母	LaTeX 代码	Unicode 代码
α	`\alpha`	`\u03B1`	μ	`\mu`	`\u03BC`
β	`\beta`	`\u03B2`	ν	`\nu`	`\u03BD`
γ	`\gamma`	`\u03B3`	ξ	`\xi`	`\u03BE`
δ	`\delta`	`\u03B4`	φ	`\varphi`	`\u03C6`
ϵ	`\epsilon`	`\u03B5`	π	`\pi`	`\u03C0`
ζ	`\zeta`	`\u03B6`	ρ	`\rho`	`\u03C1`

© 黄湘云

希腊字母	LaTeX 代码	Unicode 代码	希腊字母	LaTeX 代码	Unicode 代码
η	\eta	\u03B7	υ	\upsilon	\u03C5
θ	\theta	\u03B8	ϕ	\phi	\u03C6
ι	\iota	\u03B9	χ	\chi	\u03C7
κ	\kappa	\u03BA	ψ	\psi	\u03C8
λ	\lambda	\u03BB	ω	\omega	\u03C9
σ	\sigma	\u03C3	τ	\tau	\u03C4

表 10.2: 数字上下标

上标数字	LaTeX 代码	Unicode 代码	下标数字	LaTeX 代码	Unicode 代码
^{0}	{}^0	\u2070	$_{0}$	{}_0	\u2080
^{1}	{}^1	\u00B9	$_{1}$	{}_1	\u2081
^{2}	{}^2	\u00B2	$_{2}$	{}_2	\u2082
^{3}	{}^3	\u00B2	$_{3}$	{}_3	\u2083
^{4}	{}^4	\u2074	$_{4}$	{}_4	\u2084
^{5}	{}^5	\u2075	$_{5}$	{}_5	\u2085
^{6}	{}^6	\u2076	$_{6}$	{}_6	\u2086
^{7}	{}^7	\u2077	$_{7}$	{}_7	\u2087
^{8}	{}^8	\u2078	$_{8}$	{}_8	\u2088
^{9}	{}^9	\u2079	$_{9}$	{}_9	\u2089
^{n}	{}^n	\u207F	$_{n}$	{}_n	-

其它字母，请查看 Unicode 字母表

10.1.8 图例

```
x <- seq(-6, 6, length = 200)
y <- sin(x)
z <- cos(x)
plot(y ~ x,
  type = "l", lwd = 3,
  ylab = "", xlab = "angle", main = "Trigonometric functions"
)
abline(h = 0, lty = 3)
abline(v = 0, lty = 3)
lines(z ~ x, type = "l", lwd = 3, col = "red")
legend(-6, -1,
  yjust = 0,
  c("Sine", "Cosine"),
  lwd = 3, lty = 1, col = c(par("fg"), "red")
)
```

© 黄湘云

图 10.21: 批量生成函数图形

图 10.22: 批量生成函数图形

© 黄湘云

图 10.23: 批量生成函数图形

图 10.24: 三角函数添加图例

© 基础统计


```
xmin <- par("usr")[1]
xmax <- par("usr")[2]
ymin <- par("usr")[3]
ymax <- par("usr")[4]

plot(y ~ x,
  type = "l", lwd = 3,
  ylab = "", xlab = "angle", main = "Trigonometric functions"
)
abline(h = 0, lty = 3)
abline(v = 0, lty = 3)
lines(z ~ x, type = "l", lwd = 3, col = "red")
legend("bottomleft",
  c("Sine", "Cosine"),
  lwd = 3, lty = 1, col = c(par("fg"), "red")
)
```

图 10.25: 设置图例的位置

```
plot(y ~ x,
  type = "l", lwd = 3,
```

© 黄湘云


```
  ylab = "", xlab = "angle", main = "Trigonometric functions"
)
abline(h = 0, lty = 3)
abline(v = 0, lty = 3)
lines(z ~ x, type = "l", lwd = 3, col = "red")
legend("bottomleft",
  c("Sine", "Cosine"),
  inset = c(.03, .03),
  lwd = 3, lty = 1, col = c(par("fg"), "red")
)
```

图 10.26: insert 函数微调图例位置

```
op <- par(no.readonly = TRUE)
plot(y ~ x,
  type = "l", lwd = 3,
  ylab = "", xlab = "angle", main = "Trigonometric functions"
)
abline(h = 0, lty = 3)
abline(v = 0, lty = 3)
lines(z ~ x, type = "l", lwd = 3, col = "red")
```

©南京邮电大学

```
par(xpd = TRUE) # Do not clip to the drawing area 关键一行/允许出界
lambda <- .025
legend(par("usr")[1],
  (1 + lambda) * par("usr")[4] - lambda * par("usr")[3],
  c("Sine", "Cosine"),
  xjust = 0, yjust = 0,
  lwd = 3, lty = 1, col = c(par("fg"), "red")
)
```

图 10.27: 将图例放在绘图区域外面

```
par(op)
```

Hmisc 包的 labcurve 函数可以在曲线上放置名称，而不是遥远的图例上

10.1.9 边空

边空分为内边空和外边空

```
## Warning in knitr::include_graphics(path = paste(system.file("help/figures", : It
## is highly recommended to use relative paths for images. You had absolute paths:
## "/opt/R/4.1.3/lib/R/library/graphics/help/figures/mai.png"; "/opt/R/4.1.3/lib/R/
```

© 王建硕


```
## library/graphics/help/figures/oma.png"
```

(a) 内边空 (b) 外边空

图 10.28: 边空

line 第一行

```r
N <- 200
x <- runif(N, -4, 4)
y <- sin(x) + .5 * rnorm(N)
plot(x, y,
  xlab = "", ylab = "",
  main = paste(
    "The \"mtext\" function",
    paste(rep(" ", 60), collapse = "")
  )
)
for (i in seq(from = 0, to = 1, by = 1)) {
  mtext(paste("Line", i), 3, line = i)
}
```

par

```r
# 多图排列/分屏 page 47
# 最常用的是 par mfrow mfcol分别按行/列放置图形
op <- par(
  mfrow = c(2, 2),
  oma = c(0, 0, 4, 0) # Outer margins
)
for (i in 1:4) {
  plot(runif(20), runif(20),
    main = paste("random plot (", i, ")", sep = "")
  )
```

© 黄湘云

图 10.29: 外边空在图的边缘添加文字

```
}
par(op)
mtext("Four plots, without enough room for this title",
  side = 3, font = 2, cex = 1.5, col = "red"
) # 总/大标题放不下
```

par 的 oma 用来设置外边空的大小，默认情形下没有外边空的

```
par()$oma
```

```
## [1] 0 0 0 0
```

我们可以自己设置外边空

```
op <- par(
  mfrow = c(2, 2),
  oma = c(0, 0, 3, 0) # Outer margins
)
for (i in 1:4) {
  plot(runif(20), runif(20),
    main = paste("random plot (", i, ")", sep = "")
  )
}
par(op)
```

© 黄湘云

图 10.30: 多图排列共享一个大标题

© 黄湘云

```
mtext("Four plots, with some room for this title",
  side = 3, line = 1.5, font = 1, cex = 1.5, col = "red"
)
```

图 10.31: 设置外边空放置大标题

除了内边空还有外边空，内外边空用来放注释说明

```
op <- par(no.readonly = TRUE)
par(oma = c(2, 2, 2, 2))
plot(1, 1, type = "n", xlab = "", ylab = "", xaxt = "n", yaxt = "n")
for (side in 1:4) {
  inner <- round(par()$mar[side], 0) - 1
  for (line in 0:inner) {
    mtext(text = paste0("Inner line ", line), side = side, line = line)
  }
```

© 基础图形


```
  outer <- round(par()$oma[side], 0) - 1
  for (line in 0:inner) {
    mtext(text = paste0("Outer line ", line), side = side, line = line, outer = TRUE)
  }
}
```

外边空可以用来放图例

```
set.seed(1234)
x <- runif(10)
y <- runif(10)
cols <- rep(hcl.colors(5), each = 2)
op <- par(oma = c(2, 2, 0, 4), mar = c(3, 3, 2, 0), mfrow = c(2, 2), pch = 16)
for (i in 1:4) {
  plot(x, y, col = cols, ylab = "", xlab = "")
}
mtext(text = "A common x-axis label", side = 1, line = 0, outer = TRUE)
mtext(text = "A common y-axis label", side = 2, line = 0, outer = TRUE)
legend(
  x = 1, y = 1.2, legend = LETTERS[1:5],
  col = unique(cols), pch = 16, bty = "n", xpd = NA
)
```

© 黄湘云

```
par(op)
```

坐标轴标签 `xlab` 和 `ylab` 的内容很长的时候需要内边空

```
par(cex.lab = 1.7)
plot(1, 1,
  ylab = "A very very long axis title\nthat need special care",
  xlab = "", type = "n"
)

# 增加内边空的大小
par(mar = c(5, 7, 4, 2))
plot(1, 1,
  ylab = "A very very long axis title\nthat need special care",
  xlab = "", type = "n"
)
```

© 黄湘云

有时候，仅仅增加内边空还不够，坐标轴标签内容甚至可以出现在绘图区域外面，设置 `outer = TRUE`

```
par(oma = c(0, 4, 0, 0))
plot(1, 1, ylab = "", xlab = "", type = "n")
mtext(
  text = "A very very long axis title\nthat need special care",
  side = 2, line = 0, outer = TRUE, cex = 1.7
)
```

© 黄湘云

```
op <- par(
  mfrow = c(2, 2),
  oma = c(0, 0, 3, 0),
  mar = c(3, 3, 4, 1) + .1 # Margins
)
for (i in 1:4) {
  plot(runif(20), runif(20),
    xlab = "", ylab = "",
    main = paste("random plot (", i, ")", sep = "")
  )
}
par(op)
mtext("Title",
  side = 3, line = 1.5, font = 2, cex = 2, col = "red"
)
```

10.1.10 图层

覆盖图形 `add = T` or `par(new=TRUE)`

```
plot(runif(5), runif(5),
  xlim = c(0, 1), ylim = c(0, 1)
)
points(runif(5), runif(5),
```

© 黄湘云

图 10.32: 设置每个子图的边空 mar

© 基湘石


```
  col = "#EA4335", pch = 16, cex = 3
)
lines(runif(5), runif(5), col = "red")
segments(runif(5), runif(5), runif(5), runif(5),
  col = "blue"
)
title(main = "Overlaying points, segments, lines...")
```

图 10.33: 添加图层

10.1.11 布局

layout 函数布局，绘制复杂组合图形

```
op <- par(oma = c(0, 0, 3, 0))
layout(matrix(c(
  1, 1, 1,
  2, 3, 4,
  2, 3, 4
), nr = 3, byrow = TRUE))
hist(rnorm(n), col = "light blue")
```

© 黄湘云


```
hist(rnorm(n), col = "light blue")
hist(rnorm(n), col = "light blue")
hist(rnorm(n), col = "light blue")
mtext("The \"layout\" function",
  side = 3, outer = TRUE,
  font = 2, cex = 1.2
)
```

图 10.34: 更加复杂的组合图形

10.1.12 组合

par 之 fig 参数很神奇，使得多个图可以叠加在一起，它接受一个数值向量 c(x1, x2, y1, y2)，是图形设备显示区域中的绘图区域的 (NDC, normalized device coordinates) 坐标。

© 基础统计


```
plot(1:12,
  type = "b", main = "'fg' : axes, ticks and box in gray",
  fg = gray(0.7), bty = "7", sub = R.version.string
)
par(fig = c(1, 6, 5, 10) / 10, new = T)
plot(6:10,
  type = "b", main = "",
  fg = gray(0.7), bty = "7", xlab = R.version.string
)
```

图 10.35: 多图叠加

fig 参数控制图形的位置，用来绘制组合图形

© 王兆军


```
n <- 1000
x <- rt(n, df = 10)
hist(x,
  col = "light blue",
  probability = "TRUE", main = "",
  ylim = c(0, 1.2 * max(density(x)$y))
)
lines(density(x),
  col = "red",
  lwd = 3
)
op <- par(
  fig = c(.02, .4, .5, .98),
  new = TRUE
)
qqnorm(x,
  xlab = "", ylab = "", main = "",
  axes = FALSE
)
qqline(x, col = "red", lwd = 2)
box(lwd = 2)
```

图 10.36: 组合图形

内部资料 ©

```
par(op)
```

10.1.13 分屏

split.screen 分屏组合

```
random.plot <- function() {
  N <- 200
  f <- sample(
    list(
      rnorm,
      function(x) {
        rt(x, df = 2)
      },
      rlnorm,
      runif
    ),
    1
  ) [[1]]
  x <- f(N)
  hist(x, col = "lightblue", main = "", xlab = "", ylab = "", axes = F)
  axis(1)
}
op <- par(bg = "white", mar = c(2.5, 2, 1, 2))
split.screen(c(2, 1))
```

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

```
split.screen(c(1, 3), screen = 2)
```

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

```
screen(1)
random.plot()
# screen(2); random.plot() # Screen 2 was split into three screens: 3, 4, 5
screen(3)
random.plot()
screen(4)
random.plot()
screen(5)
random.plot()
```

```
close.screen(all = TRUE)
par(op)
```

© 黄湘云

图 10.37: 分屏

10.1.14 交互

辅助绘图 identify locator

10.2 基础统计图形

按图的类型划分，最后在小结部分给出各图适用的数据类型

根据数据类型划分：对于一元数据，可用什么图来描述；多元数据呢，连续数据和离散数据（分类数据）

先找一个不重不漏的划分，指导原则是根据数据类型选择图，根据探索到的数据中的规律，选择图

其它 assocplot fourfoldplot sunflowerplot

10.2.1 条形图

条形图

简单条形图

```
data(diamonds, package = "ggplot2") # 加载数据
par(mar = c(2, 5, 1, 1))
barCenters <- barplot(table(diamonds$cut),
  col = "lightblue", axes = FALSE,
```

© 基础统计

```
  axisnames = FALSE, horiz = TRUE, border = "white"
)
text(
  y = barCenters, x = par("usr")[3],
  adj = 1, labels = names(table(diamonds$cut)), xpd = TRUE
)
axis(1,
  labels = seq(0, 25000, by = 5000), at = seq(0, 25000, by = 5000),
  las = 1, col = "gray"
)
grid()
```

图 10.38: 条形图

简单柱形图

```
set.seed(123456)
barPois <- table(stats::rpois(1000, lambda = 5))
plot(barPois, col = "lightblue", type = "h", lwd = 10, main = "")
box(col = "gray")
```

复合条形图

```
par(mar = c(4.1, 2.1, 0.5, 4.5))
barplot(VADeaths,
  border = "white", horiz = FALSE, col = hcl.colors(5),
```

© 黄湘云

图 10.39: 柱形图

```
  legend.text = rownames(VADeaths), xpd = TRUE, beside = TRUE,
  cex.names = 0.9,
  args.legend = list(
    x = "right", border = "white", title = "Age",
    box.col = NA, horiz = FALSE, inset = c(-.2, 0),
    xpd = TRUE
  ),
  panel.first = grid(nx = 0, ny = 7)
)
```

堆积条形图

```
par(mar = c(4.1, 2.1, 0.5, 4.5))
barplot(VADeaths,
  border = "white", horiz = FALSE, col = hcl.colors(5),
  legend.text = rownames(VADeaths), xpd = TRUE, beside = FALSE,
  cex.names = 0.9,
  args.legend = list(
    x = "right", border = "white", title = "Age",
    box.col = NA, horiz = FALSE, inset = c(-.2, 0),
    xpd = TRUE
  ),
  panel.first = grid(nx = 0, ny = 4)
```

© 黄湘云

图 10.40: 复合条形图

```
)
```

- 堆积条形图 spineplot

简单条形图

```
barplot(
  data = BOD, demand ~ Time, ylim = c(0, 20),
  border = "white", horiz = FALSE, col = hcl.colors(1)
)
```

© 黄湘云

图 10.41: 堆积条形图

© 清华大学出版社

```
pg_mean <- aggregate(weight ~ group, data = PlantGrowth, mean)
barplot(
  data = pg_mean, weight ~ group,
  border = "white", horiz = FALSE, col = hcl.colors(3)
)
```

Titanic 数据集是 table 数据类型

简单条形图

复合条形图

```
barplot(Freq ~ Class + Survived,
  data = Titanic,
  subset = Age == "Adult" & Sex == "Male",
  beside = TRUE,
  border = "white", horiz = FALSE, col = hcl.colors(4),
  args.legend = list(
    border = "white", title = "Class",
    box.col = NA, horiz = FALSE,
    xpd = TRUE
  ),
  ylab = "# {passengers}", legend = TRUE
)
```

© 黄湘云

堆积条形图

```
barplot(Freq ~ Class + Survived,
  data = Titanic,
  subset = Age == "Adult" & Sex == "Male",
  border = "white", horiz = FALSE, col = hcl.colors(4),
  args.legend = list(
    border = "white", title = "Class",
    box.col = NA, horiz = FALSE,
    xpd = TRUE
  ),
  ylab = "# {passengers}", legend = TRUE
)
```

© 黄湘云

10.2.2 直方图

```
set.seed(1234)
n <- 2^24
x <- runif(n, 0, 1)
delta <- 0.01
len <- diff(c(0, which(x < delta), n + 1)) - 1
ylim <- seq(0, 1800, by = 300)
xlim <- seq(0, 100, by = 20)
p <- hist(len[len < 101], breaks = -1:100 + 0.5, plot = FALSE)
plot(p, ann = FALSE, axes = FALSE, col = "lightblue", border = "white", main = "")
axis(1, labels = xlim, at = xlim, las = 1) # x 轴
axis(2, labels = ylim, at = ylim, las = 0) # y 轴
box(col = "gray")
```

```
with(faithful, plot(eruptions ~ waiting, pch = 16))
```

© 黄湘云

图 10.42: 直方图

```
with(faithful, hist(waiting,
  main = "Time between Old Faithful eruptions",
  xlab = "Minutes", ylab = "",
  cex.main = 1.5, cex.lab = 1.5, cex.axis = 1.4
))
```

© 黄湘云

```
with(data = faithful, {
  hist(eruptions, seq(1.6, 5.2, 0.2),
    prob = TRUE,
    main = "", col = "lightblue", border = "white"
  )
  lines(density(eruptions, bw = 0.1), col = "#EA4335")
  rug(eruptions, col = "#EA4335") # 添加数据点
})
```

```
hist(longley$Unemployed,
  probability = TRUE,
  col = "light blue", main = ""
)
# 添加密度估计
lines(density(longley$Unemployed),
  col = "red",
  lwd = 3
)
```

直方图有很多花样的，添加阴影线，angle 控制倾斜的角度

```
# hist(longley$Unemployed, density = 1, angle = 45)
# hist(longley$Unemployed, density = 3, angle = 15)
# hist(longley$Unemployed, density = 1, angle = 15)
hist(longley$Unemployed, density = 3, angle = 45, main = "")
```

© 黄湘云

图 10.43: 老忠实泉间歇性喷水的时间间隔分布

图 10.44: 概率密度分布

© 黄湘云

图 10.45: density 数值越大阴影线越密

10.2.3 密度图

```
data(galaxies, package = "MASS")
galaxies <- galaxies / 1000
# Bandwidth Selection by Pilot Estimation of Derivatives
c(MASS::width.SJ(galaxies, method = "dpi"), MASS::width.SJ(galaxies))
```

```
## [1] 3.256151 2.566423
```

```
plot(
  x = c(5, 40), y = c(0, 0.2), type = "n", bty = "l",
  xlab = "velocity of galaxy (km/s)", ylab = "density"
)
rug(galaxies)
lines(density(galaxies, width = 3.25, n = 200), col = "blue", lty = 1)
lines(density(galaxies, width = 2.56, n = 200), col = "red", lty = 3)
```

© 黄湘云

```
x <- seq(from = 100, to = 174, by = 0.5)
y1 <- dnorm(x, mean = 145, sd = 9)
y2 <- dnorm(x, mean = 128, sd = 8)
plot(x, y1,
  type = "l", lwd = 2, col = "firebrick3",
 main = "Systolic Blood Pressure Before and After Treatment",
  xlab = "Systolic Blood Pressure (mmHg)",
  ylab = "Frequency", yaxt = "n",
  xlim = c(100, 175), ylim = c(0, 0.05)
)

lines(x, y2, col = "dodgerblue4")
polygon(c(117, x, 175), c(0, y2, 0),
  col = "dodgerblue4",
  border = "white"
)

polygon(c(100, x, 175), c(0, y1, 0),
  col = "firebrick3",
  border = "white"
)

axis(2,
  at = seq(from = 0, to = 0.05, length.out = 8),
```

© 基础统计

```
  labels = seq(from = 0, to = 175, by = 25), las = 1
)

text(x = 100, y = 0.0445, "Pre-Treatment BP", col = "dodgerblue4", cex = 0.9, pos = 4)
text(x = 100, y = 0.0395, "Post-Treatment BP", col = "firebrick3", cex = 0.9, pos = 4)
points(100, 0.0445, pch = 15, col = "dodgerblue4")
points(100, 0.0395, pch = 15, col = "firebrick3")
abline(v = c(145, 128), lwd = 2, lty = 2, col = 'gray')
```

```
days <- abs(rnorm(1000, 80, 125))
plot(density(days, from = 0),
  main = "Density plot",
  xlab = "Number of days since trial started"
)
```

© 基湘工

```
plot(density(days, from = 0, to = 180, adjust = 0.2),
  main = "Density plot - Up to 180 days (86% of data)",
  xlab = "Number of days since trial started"
)
```

```
library(survival)
surv.days <- Surv(days)
surv.fit <- survfit(surv.days ~ 1)
plot(surv.fit,
  main = "Kaplan-Meier estimate with 95% confidence bounds (86% of data)",
  xlab = "Days since trial started",
  xlim = c(0, 180),
```

© 基础图

```
  ylab = "Survival function"
)
grid(20, 10, lwd = 2)
```

10.2.4 经验图

```
with(data = faithful, {
  long <- eruptions[eruptions > 3]
  plot(ecdf(long), do.points = FALSE, verticals = TRUE, main = "")
  x <- seq(3, 5.4, 0.01)
  lines(x, pnorm(x, mean = mean(long), sd = sqrt(var(long))), lty = 3)
})
```

10.2.5 QQ 图

```
with(data = faithful, {
  long <- eruptions[eruptions > 3]
  par(pty = "s") # arrange for a square figure region
  qqnorm(long, main = "")
  qqline(long)
```

© 黄湘云

图 10.46: 累积经验分布图

```
})
```

© 黄湘云

10.2.6 时序图

时序图最适合用来描述股价走势

```
matplot(time(EuStockMarkets), EuStockMarkets,
  main = "",
  xlab = "Date", ylab = "closing prices",
  pch = 17, type = "l", col = 1:4
)
legend("topleft", colnames(EuStockMarkets), pch = 17, lty = 1, col = 1:4)
```

10.2.7 饼图

clockwise 参数

```
pie.sales <- c(0.12, 0.3, 0.26, 0.16, 0.04, 0.12)
names(pie.sales) <- c(
  "Blueberry", "Cherry",
  "Apple", "Boston Cream", "Other", "Vanilla Cream"
)
pie(pie.sales, clockwise = TRUE, main = "")
segments(0, 0, 0, 1, col = "red", lwd = 2)
text(0, 1, "init.angle = 90", col = "red")
```

© 基湘石

图 10.47: 1991–1998 年间主要欧洲股票市场日闭市价格指数图德国 DAX (Ibis), Switzerland SMI, 法国 CAC 和英国 FTSE

© 黄湘云

10.2.8 茎叶图

```
stem(longley$Unemployed)
```

```
##
##   The decimal point is 2 digit(s) to the right of the |
##
##   1 | 99
##   2 | 134899
##   3 | 46789
##   4 | 078
```

10.2.9 散点图

在一维空间上，绘制散点图，其实是在看散点的疏密程度随坐标轴的变化

```
stripchart(longley$Unemployed,
  method = "jitter",
  jitter = 0.1, pch = 16, col = "lightblue"
)
stripchart(longley$Unemployed,
  method = "overplot",
```

© 基湘石


```
  pch = 16, col = "lightblue"
)
```

(a) 抖动图

图 10.48: 一维散点图

气泡图是二维散点图的一种变体，气泡的大小可以用来描述第三个变量，下面以数据集 topo 为例展示气泡图

```
# 加载数据集
data(topo, package = "MASS")
# 查看数据集
str(topo)
```

```
## 'data.frame':    52 obs. of  3 variables:
##  $ x: num  0.3 1.4 2.4 3.6 5.7 1.6 2.9 3.4 3.4 4.8 ...
##  $ y: num  6.1 6.2 6.1 6.2 6.2 5.2 5.1 5.3 5.7 5.6 ...
##  $ z: int  870 793 755 690 800 800 730 728 710 780 ...
```

topo 是空间地形数据集，包含有 52 行 3 列，数据点是 310 平方英尺范围内的海拔高度数据，x 坐标每单位 50 英尺，y 坐标单位同 x 坐标，海拔高度 z 单位是英尺

```
plot(y ~ x,
  cex = (960 - z) / (960 - 690) * 3, data = topo,
  xlab = "X Coordinates", ylab = "Y coordinates"
)
```

散点图也适合分类数据的展示，在图中用不同颜色或符号标记数据点所属类别，即在普通散点图的基础上添加一分类变量的描述

```
plot(mpg ~ hp,
  data = subset(mtcars, am == 1), pch = 16, col = "blue",
  xlim = c(50, 350), ylim = c(10, 35)
)
points(mpg ~ hp,
  col = "red", pch = 16,
  data = subset(mtcars, am == 0)
)
```

© 黄湘云

图 10.49: 地形图之海拔高度

```
legend(300, 35,
  c("1", "0"),
  title = "am",
  col = c("blue", "red"),
  pch = c(16, 16)
)
```

iris 数据

```
plot(Sepal.Length ~ Sepal.Width, data = iris, col = Species, pch = 16)
legend("topright",
  legend = unique(iris$Species), box.col = "gray",
  pch = 16, col = unique(iris$Species)
)
box(col = "gray")
```

分组散点图和平滑

```
library(carData)
library(car)
scatterplot(Sepal.Length ~ Sepal.Width,
  col = c("black", "red", "blue"), pch = c(16, 16, 16),
  smooth = TRUE, boxplots = "xy", groups = iris$Species,
  xlab = "Sepal.Width", ylab = "Sepal.Length", data = iris
)
```

© 黄湘云

图 10.50: 分类散点图

图 10.51: 分类散点图

© 基础统计

有时为了实现特定的目的，需要高亮其中某些点，按类别或者因子变量分组绘制散点图，这里继续采用 stripchart 函数绘制二维散点图10.52，由左图可知，函数 stripchart 提供的参数 pch 不接受向量，实际只是取了前三个值 16 16 17 对应于 Species 的三类，关键是高亮的分界点是有区分意义的

```
data("iris")
pch <- rep(16, length(iris$Petal.Length))
pch[which(iris$Petal.Length < 1.4)] <- 17
stripchart(Petal.Length ~ Species,
  data = iris,
  vertical = TRUE, method = "jitter",
  pch = pch
)
# 对比一下
stripchart(Petal.Length ~ Species,
  data = iris, subset = Petal.Length > 1.4,
  vertical = TRUE, method = "jitter", ylim = c(1, 7),
  pch = 16
)
stripchart(Petal.Length ~ Species,
  data = iris, subset = Petal.Length < 1.4,
  vertical = TRUE, method = "jitter", add = TRUE,
  pch = 17, col = "red"
)
```

(a) 原图 (b) 高亮

图 10.52: 高亮图中部分散点

如果存在大量散点

```
densCols(x,
  y = NULL, nbin = 128, bandwidth,
  colramp = colorRampPalette(blues9[-(1:3)])
```

© 黄湘云


```
)
```

densCols 函数根据点的局部密度生成颜色，密度估计采用核平滑法，由 **KernSmooth** 包的 bkde2D 函数实现。参数 colramp 传递一个函数，colorRampPalette 根据给定的几种颜色生成函数，参数 bandwidth 实际上是传给 bkde2D 函数

```
plot(faithful,
  col = densCols(faithful),
  pch = 20, panel.first = grid()
)
```

图 10.53: 根据点的密度生成颜色

气泡图也是散点图的一种

```
plot(Volume ~ Height,
  data = trees, pch = 16, cex = Girth / 8,
  col = rev(terrain.colors(nrow(trees), alpha = .5))
)
box(col = "gray")
```

气泡图

```
# 空白画布
plot(c(1, 5, 10), c(1, 5, 10), panel.first = grid(10, 10),
     type = "n", axes = FALSE, ann = FALSE)
```

© 黄湘云

图 10.54: 气泡图

```
# 添加坐标轴
axis(1, at = seq(10), labels = TRUE)
axis(2, at = seq(10), labels = TRUE)
par(new = TRUE) # 在当前图形上添加图形
# axes 坐标轴上的刻度 "xaxt" or "yaxt"  ann 坐标轴和标题的标签
set.seed(1234)
plot(rnorm(100, 5, 1), rnorm(100, 5, 1),
  cex = runif(100, 0, 2),
  col = hcl.colors(4)[rep(seq(4), 100)],
  bg = paste0("gray", replicate(100, sample(seq(100), 1, replace = TRUE))),
  axes = FALSE, ann = FALSE, pch = 21, lwd = 2
)
legend("top",
  legend = paste0("class", seq(4)), col = hcl.colors(4),
  pt.lwd = 2, pch = 21, box.col = "gray", horiz = TRUE
)
```

© 黄湘云

除了 par(new=TRUE) 设置外，有些函数本身就具有 add 选项

```
set.seed(1234)
plot(dist ~ speed, data = cars, pch = 17, col = "red", cex = 1)
with(cars, symbols(dist ~ speed,
  circles = runif(length(speed), 0, 1),
  pch = 16, inches = .5, add = TRUE
))
z <- lm(dist ~ speed, data = cars)
abline(z, col = "blue")
curve(tan, from = 0, to = 8 * pi, n = 100, add = TRUE)
lines(stats::lowess(cars))
points(10, 100, pch = 16, cex = 3, col = "green")
text(10, 80, "text here", cex = 3)
```

© 黄湘云

10.2.10 抖动图

抖动散点图

```
mat <- matrix(1:length(colors()), ncol = 9, byrow = TRUE)
df <- data.frame(
  col = colors(),
  x = as.integer(cut(1:length(colors()), 9)),
  y = rep(1:73, 9), stringsAsFactors = FALSE
)
par(mar = c(4, 4, 1, 0.1))
plot(y ~ jitter(x),
  data = df, col = df$col,
  pch = 16, main = "Visualizing colors() split in 9 groups",
  xlab = "Group",
  ylab = "Element of the group (min = 1, max = 73)",
  sub = "x = 3, y = 1 means that it's the 2 * 73 + 1 = 147th color"
)
```

© 黄湘云

图 10.55: 抖动散点图

内部资料 ©

10.2.11 箱线图

boxplotdbl: Double Box Plot for Two-Axes Correlation. Correlation chart of two set (x and y) of data. Using Quartiles with boxplot style. Visualize the effect of factor.

复合箱线图

```
with(data = iris, {
  op <- par(mfrow = c(2, 2), mar = c(4, 4, 2, .5))
  plot(Sepal.Length ~ Species)
  plot(Sepal.Width ~ Species)
  plot(Petal.Length ~ Species)
  plot(Petal.Width ~ Species)
  par(op)
  mtext("Edgar Anderson's Iris Data", side = 3, line = 4)
})
```

箱线图的花样也很多

```
data(InsectSprays)
par(mar = c(4, 4, .5, .5))
boxplot(
  data = InsectSprays, count ~ spray,
  col = "gray", xlab = "Spray", ylab = "Count"
)

boxplot(
  data = InsectSprays, count ~ spray,
  col = "gray", horizontal = TRUE,
  las = 1, xlab = "Count", ylab = "Spray"
)
```

Notched Boxplots

```
set.seed(1234)
n <- 8
g <- gl(n, 100, n * 100) # n水平个数 100是重复次数
x <- rnorm(n * 100) + sqrt(as.numeric(g))
boxplot(split(x, g), col = gray.colors(n), notch = TRUE)
title(
  main = "Notched Boxplots", xlab = "Group",
  font.main = 4, font.lab = 1
)
```

© 黄湘云

图 10.56: 安德森的鸢尾花数据

© 黄湘云

(a) 垂直放置

(b) 水平放置

图 10.57: 箱线图

© 基湘石

真实的情况是这样的

```
cumcm2011A <- readRDS(file = "cumcm2011A.RDS")
par(mfrow = c(2, 4), mar = c(4, 3, 1, 1))
with(cumcm2011A, boxplot(As, xlab = "As"))
abline(h = c(1.8, 3.6, 5.4), col = c("green", "blue", "red"), lty = 2)

with(cumcm2011A, boxplot(Cd, xlab = "Cd"))
abline(h = c(70, 130, 190), col = c("green", "blue", "red"), lty = 2)

with(cumcm2011A, boxplot(Cr, xlab = "Cr"))
abline(h = c(13, 31, 49), col = c("green", "blue", "red"), lty = 2)

with(cumcm2011A, boxplot(Cu, xlab = "Cu"))
abline(h = c(6.0, 13.2, 20.4), col = c("green", "blue", "red"), lty = 2)

with(cumcm2011A, boxplot(Hg, xlab = "Hg"))
abline(h = c(19, 35, 51), col = c("green", "blue", "red"), lty = 2)

with(cumcm2011A, boxplot(Ni, xlab = "Ni"))
abline(h = c(4.7, 12.3, 19.9), col = c("green", "blue", "red"), lty = 2)

with(cumcm2011A, boxplot(Pb, xlab = "Pb"))
abline(h = c(19, 31, 43), col = c("green", "blue", "red"), lty = 2)

with(cumcm2011A, boxplot(Zn, xlab = "Zn"))
```

© 草稿

```
abline(h = c(41, 69, 97), col = c("green", "blue", "red"), lty = 2)
```

```
boxplot(As ~ area,
  data = cumcm2011A,
  col = hcl.colors(5)
)
abline(
  h = c(1.8, 3.6, 5.4), col = c("green", "blue", "red"),
  lty = 2, lwd = 2
)
```

10.2.12 残差图

iris 四个测量指标

```
vec_mean <- colMeans(iris[, -5])
vec_sd <- apply(iris[, -5], 2, sd)
plot(seq(4), vec_mean,
  ylim = range(c(vec_mean - vec_sd, vec_mean + vec_sd)),
  xlab = "Species", ylab = "Mean +/- SD", lwd = 1, pch = 19,
  axes = FALSE
)
axis(1, at = seq(4), labels = colnames(iris)[-5])
axis(2, at = seq(7), labels = seq(7))
arrows(seq(4), vec_mean - vec_sd, seq(4), vec_mean + vec_sd,
  length = 0.05, angle = 90, code = 3
)
box()
```

10.2.13 提琴图

Tom Kelly 维护的 vioplot 包 https://github.com/TomKellyGenetics/vioplot

10.2.14 轮廓图

topo 是地形数据

等高线图

10.2.15 折线图

函数曲线，样条曲线，核密度曲线，平行坐标图

- 折线图
- 点线图 `plot(type="b")` 函数曲线图 `curve` `matplot` X 样条曲线 `xspline`

© 黄湘云

图 10.58: 带标准差的均值散点图

- 时序图

太阳黑子活动数据

sunspot.month Monthly Sunspot Data, from 1749 to "Present" sunspot.year Yearly Sunspot Data, 1700-1988 sunspots Monthly Sunspot Numbers, 1749-1983

```
plot(AirPassengers)
box(col = "gray")
```

10.2.16 函数图

```
x0 <- 2^(-20:10)
nus <- c(0:5, 10, 20)
x <- seq(0, 4, length.out = 501)

plot(x0, x0^-8,
  frame.plot = TRUE, # 添加绘图框
  log = "xy", # x 和 y 轴都取对数尺度
  axes = FALSE, # 去掉坐标轴
  xlab = "$u$", ylab = "$\\mathcal{K}_{\\kappa}(u)$", # 设置坐标轴标签
  type = "n", # 清除绘图区域的内容
  ann = TRUE, # 添加标题 x和y轴标签
  panel.first = grid() # 添加背景参考线
```

© 黄湘云

图 10.59: 折线图

```
)

axis(1,
  at = 10^seq(from = -8, to = 2, by = 2),
  labels = paste0("$\\mathsf{10^{", seq(from = -8, to = 2, by = 2), "}}$")
)
axis(2,
  at = 10^seq(from = -8, to = 56, by = 16),
  labels = paste0("$\\mathsf{10^{", seq(from = -8, to = 56, by = 16), "}}$"), las = 1
)

for (i in seq(length(nus))) {
  lines(x0, besselK(x0, nu = nus[i]), col = hcl.colors(9)[i], lwd = 2)
}
legend("topright",
  legend = paste0("$\\kappa=", rev(nus), "$"),
  col = hcl.colors(9, rev = T), lwd = 2, cex = 1
)
```

还有 eta 函数和 gammaz 函数

© 基湘石

图 10.60: 贝塞尔函数

10.2.17 马赛克图

马赛克图 mosaicplot

```
plot(HairEyeColor, col = "lightblue", border = "white", main = "")
```

10.2.18 点图

dotchart 克利夫兰点图

条件图 coplot

10.2.19 矩阵图

在对角线上添加平滑曲线、密度曲线

```
pairs(longley,
  gap = 0,
  diag.panel = function(x, ...) {
    par(new = TRUE)
    hist(x,
      col = "light blue",
      probability = TRUE,
      axes = FALSE,
      main = ""
```

© 黄湘云

图 10.61: 马赛克图

© 黄湘云


```
    )
    lines(density(x),
      col = "red",
      lwd = 3
    )
    rug(x)
  }
)
```

图 10.62: 变量关系

```
# 自带 layout
plot(iris[, -5], col = iris$Species)
```

10.2.20 雷达图

星图 stars 多元数据

10.2.21 玫瑰图

注意与 image 函数区别

```
x <- 10 * (1:nrow(volcano))
y <- 10 * (1:ncol(volcano))
image(x, y, volcano, col = terrain.colors(100), axes = FALSE)
```

© 黄湘云

图 10.63: 矩阵图

© 基础统计


```
contour(x, y, volcano,
  levels = seq(90, 200, by = 5),
  add = TRUE, col = "peru"
)
axis(1, at = seq(100, 800, by = 100))
axis(2, at = seq(100, 600, by = 100))
box()
title(main = "Maunga Whau Volcano", font.main = 4)
```

图 10.64: image 图形

10.2.22 透视图

```
par(mar = c(.5, 2.1, .5, .5))
x1 <- seq(-10, 10, length = 51)
x2 <- x1
f <- function(x1, x2, mu1 = 0, mu2 = 0, s11 = 10, s12 = 15, s22 = 10, rho = 0.5) {
  term1 <- 1 / (2 * pi * sqrt(s11 * s22 * (1 - rho^2)))
  term2 <- -1 / (2 * (1 - rho^2))
  term3 <- (x1 - mu1)^2 / s11
  term4 <- (x2 - mu2)^2 / s22
  term5 <- -2 * rho * ((x1 - mu1) * (x2 - mu2)) / (sqrt(s11) * sqrt(s22))
  term1 * exp(term2 * (term3 + term4 - term5))
}
z <- outer(x1, x2, f)
```

© 统计学

```
library(shape)
persp(x1, x2, z,
  xlab = "", ylab = "", zlab = "",
  col = drapecol(z, col = terrain.colors(20)),
  border = NA, shade = 0.1, r = 50, d = 0.1, expand = 0.5,
  theta = 120, phi = 15, ltheta = 90, lphi = 180,
  ticktype = "detailed", nticks = 5
)
```

图 10.65: 统计学的世界

10.3 栅格统计图形

> If you imagine that this pen is Trellis, then Lattice is not this pen.
>
> — Paul Murrell [1]

把网站搬出来，汉化 http://latticeextra.r-forge.r-project.org/

10.3.1 箱线图

```
library(lattice)
# plot(data = InsectSprays, count ~ spray)
bwplot(count ~ spray, data = InsectSprays)
```

[1]Paul 在 DSC 2001 大会上的幻灯片见https://www.stat.auckland.ac.nz/~paul/Talks/dsc2001.pdf

© 黄湘云

10.3.2 折线图

latticeExtra 包提供了强大的图层函数 `layer()`

多元时间序列

```
library(RColorBrewer)
library(latticeExtra)
xyplot(EuStockMarkets) +
  layer(panel.scaleArrow(
    x = 0.99, append = " units", col = "grey", srt = 90, cex = 0.8
  ))
```

© 黄湘云

如何解释

时序图

Plot many time series in parallel

```
horizonplot(EuStockMarkets,
  colorkey = TRUE,
  origin = 4000, horizonscale = 1000
) +
  layer(panel.scaleArrow(
    x = 0.99, digits = 1, col = "grey",
    srt = 90, cex = 0.7
  )) +
  layer(
    lim <- current.panel.limits(),
    panel.text(lim$x[1], lim$y[1], round(lim$y[1], 1),
      font = 2,
      cex = 0.7, adj = c(-0.5, -0.5), col = "#9FC8DC"
    )
  )
```

© 黄湘云

```
# # https://stackoverflow.com/questions/25109196/r-lattice-package-add-legend-to-a-figure
library(lattice)
library(nlme)

plot(Orange,
  outer = ~1,
  key = list(
    space = "right", title = "Tree", cex.title = 1,
    lines = list(lty = 1, col = gray.colors(5)),
    # points = list(pch = 1, col = gray.colors(5)),
    text = list(c("3", "1", "5", "2", "4"))
  ),
  par.settings = list(
    # plot.line = list(col = gray.colors(5), border = "transparent"),
    # plot.symbol = list(col = gray.colors(5), border = "transparent"),
    strip.background = list(col = "white"),
    strip.border = list(col = "black")
  )
)
```

© 黄湘云

```
library(MASS)
library(lattice)
## Plot the claims frequency against age group by engine size and district

barchart(Claims / Holders ~ Age | Group,
  groups = District,
  data = Insurance, origin = 0, auto.key = TRUE
)
```

© 黄湘云

```
barchart(Claims / Holders ~ Age | Group,
  groups = District, data = Insurance,
  main = "Claims frequency",
  auto.key = list(
    space = "top", columns = 4,
    title = "District", cex.title = 1
  )
)
```

© 黄湘云

lattice 图形的参数设置

```
show.settings()
```

© 黄湘云

```
myColours <- brewer.pal(6, "Blues")
my.settings <- list(
  superpose.polygon = list(col = myColours[2:5], border = "transparent"),
  strip.background = list(col = myColours[6]),
  strip.border = list(col = "black")
)

# 获取参数设置
trellis.par.get()

# 全局参数设置
trellis.par.set(my.settings)
```

```
library(MASS)
library(lattice)
```

© 黄湘云

```
barchart(Claims / Holders * 100 ~ Age | Group,
  groups = District, data = Insurance,
  origin = 0, main = "Motor insurance claims frequency",
  xlab = "Age", ylab = "Claims frequency %",
  scales = list(alternating = 1),
  auto.key = list(
    space = "top", columns = 4,
    points = FALSE, rectangles = TRUE,
    title = "District", cex.title = 1
  ),
  par.settings = list(
    superpose.polygon = list(col = gray.colors(4), border = "transparent"),
    strip.background = list(col = "gray80"),
    strip.border = list(col = "black")
  ),
  par.strip.text = list(col = "gray40", font = 2),
  panel = function(x, y, ...) {
    panel.grid(h = -1, v = 0)
    panel.barchart(x, y, ...)
  }
)
```

© 王建平

10.3.3 平滑图

```
set.seed(1)
xy <- data.frame(
  x = runif(100),
  y = rt(100, df = 5)
)

xyplot(y ~ x, xy, panel = function(...) {
  panel.xyplot(...)
  panel.smoother(..., span = 0.9)
})
```

```
library(splines)
xyplot(y ~ x, xy) +
  layer(panel.smoother(y ~ ns(x, 5), method = "lm"))
```

© 黄湘云

```
library(nlme)
library(mgcv)
xyplot(y ~ x, xy) +
  layer(panel.smoother(y ~ s(x), method = "gam"))
```

© 黄湘云

Trellis Displays of Tukey's Hanging Rootograms

```
x <- rpois(1000, lambda = 50)
rootogram(~x, dfun = function(x) dpois(x, lambda = 50))
```

© 黄湘云

10.3.4 点图

```
# 添加背景网格线作为参考线
segplot(reorder(factor(county), rate.male) ~ LCL95.male + UCL95.male,
  data = subset(USCancerRates, state == "Washington"),
  draw.bands = FALSE, centers = rate.male
)
```

© 黄湘云

10.3.5 阶梯图

经验累积分布图

```
ecdfplot(~height | voice.part, data = singer)
```

© 黄湘云

10.3.6 分面图

```
## a variant of Figure 5.6 from Sarkar (2008)
## http://lmdvr.r-forge.r-project.org/figures/figures.html?chapter=05;figure=05_06

depth.ord <- rev(order(quakes$depth))
quakes$Magnitude <- equal.count(quakes$mag, 4)
quakes.ordered <- quakes[depth.ord, ]

levelplot(depth ~ long + lat | Magnitude,
  data = quakes.ordered,
  panel = panel.levelplot.points, type = c("p", "g"),
  aspect = "iso", prepanel = prepanel.default.xyplot
)
```

© 黄湘云

10.3.7 等高线图

```
set.seed(1)
xyz <- data.frame(x = rnorm(100), y = rnorm(100))
xyz$z <- with(xyz, x * y + rnorm(100, sd = 1))

## GAM smoother with smoothness by cross validation
library(mgcv)
levelplot(z ~ x * y, xyz,
  panel = panel.2dsmoother,
  form = z ~ s(x, y), method = "gam"
)
```

© 黄湘云

10.3.8 透视图

```
library(shape)
persp(volcano,
  theta = 30, phi = 20,
  r = 50, d = 0.1, expand = 0.5, ltheta = 90, lphi = 180,
  shade = 0.1, ticktype = "detailed", nticks = 5, box = TRUE,
  col = drapecol(volcano, col = terrain.colors(100)),
  xlab = "X", ylab = "Y", zlab = "Z", border = "transparent",
  main = "Topographic Information \n on Auckland's Maunga Whau Volcano"
)
```

10.3.9 聚类图

```
xyplot(Sepal.Length ~ Petal.Length,
  groups = Species,
  data = iris, scales = "free",
  par.settings = list(
    superpose.symbol = list(pch = c(15:17)),
    superpose.line = list(lwd = 2, lty = 1:3)
  ),
  panel = function(x, y, ...) {
    panel.xyplot(x, y, ...)
```

© 黄湘云

图 10.66: (ref:volcano-topo)

© 基础图

```
    panel.ellipse(x, y, ...)
  },
  auto.key = list(x = .1, y = .8, corner = c(0, 0))
)
```

```
# lattice 书 6.3.1 节 参数曲面

kx <- function(u, v) cos(u) * (r + cos(u / 2))
ky <- function(u, v) {
  sin(u) * (r + cos(u / 2) * sin(t * v) -
    sin(u / 2) * sin(2 * t * v)) * sin(t * v) -
    sin(u / 2) * sin(2 * t * v)
}


kz <- function(u, v) sin(u / 2) * sin(t * v) + cos(u / 2) * sin(t * v)
n <- 50
u <- seq(0.3, 1.25, length = n) * 2 * pi
v <- seq(0, 1, length = n) * 2 * pi
um <- matrix(u, length(u), length(u))
vm <- matrix(v, length(v), length(v), byrow = TRUE)
r <- 2
t <- 1

wireframe(kz(um, vm) ~ kx(um, vm) + ky(um, vm),
  shade = TRUE, xlab = expression(x[1]),
```

©谢益辉


```
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  screen = list(z = 170, x = -60),
  alpha = 0.75, panel.aspect = 0.6, aspect = c(1, 0.4),
  scales = list(arrows = FALSE, col = "black"),
  lattice.options = list(
    layout.widths = list(
      left.padding = list(x = -.6, units = "inches"),
      right.padding = list(x = -1.0, units = "inches")
    ),
    layout.heights = list(
      bottom.padding = list(x = -.8, units = "inches"),
      top.padding = list(x = -1.0, units = "inches")
    )
  ),
  par.settings = list(
    axis.line = list(col = "transparent")
  )
)
```

10.4 运行环境

```
xfun::session_info()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Locale:
##   LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##   LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##   LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##   LC_PAPER=en_US.UTF-8       LC_NAME=C
##   LC_ADDRESS=C               LC_TELEPHONE=C
##   LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## Package version:
##   askpass_1.1         base64enc_0.1.3     bookdown_0.25
##   bslib_0.3.1         cli_3.2.0           codetools_0.2-18
##   compiler_4.1.3      curl_4.3.2          digest_0.6.29
##   evaluate_0.15       fastmap_1.1.0       filehash_2.4-3
##   fs_1.5.2            glue_1.6.2          graphics_4.1.3
##   grDevices_4.1.3     grid_4.1.3          highr_0.9
##   htmltools_0.5.2     jpeg_0.1-9          jquerylib_0.1.4
```

© 基础统计

```
##   jsonlite_1.8.0      KernSmooth_2.23-20  knitr_1.38
##   lattice_0.20-45     latticeExtra_0.6-29 magick_2.7.3
##   magrittr_2.0.3      mapproj_1.2.8       maps_3.4.0
##   MASS_7.3-56         Matrix_1.4-1        methods_4.1.3
##   mgcv_1.8-40         nlme_3.1-157        pdftools_3.1.1
##   png_0.1-7           qpdf_1.1            R6_2.5.1
##   rappdirs_0.3.3      RColorBrewer_1.1-2  Rcpp_1.0.8.3
##   rlang_1.0.2         rmarkdown_2.13      sass_0.4.1
##   shape_1.4.6         splines_4.1.3       stats_4.1.3
##   stringi_1.7.6       stringr_1.4.0       survival_3.3-1
##   sys_3.4             sysfonts_0.8.8      tikzDevice_0.12.3.1
##   tinytex_0.38        tools_4.1.3         utils_4.1.3
##   xfun_0.30           yaml_2.3.5
```

© 黄湘云

第十一章　数据可视化

```
library(ggplot2)          # ggplot2 图形
library(patchwork)        # 图形布局
library(magrittr)         # 管道操作
library(ggrepel)          # 文本注释
library(extrafont)        # 加载外部字体 TTF
library(hrbrthemes)       # 主题
library(maps)             # 地图数据
library(mapdata)          # 地图数据
library(xkcd)             # 漫画字体
library(RgoogleMaps)      # 静态地图
library(data.table)       # 数据操作
library(KernSmooth)       # 核平滑
library(ggnormalviolin)   # 提琴图
library(ggbeeswarm)       # 蜂群图
library(gert)             # Git 数据操作
library(ggridges)         # 岭线图
library(ggpubr)           # 组合图
library(treemap)          # 树状图
library(treemapify)       # 树状图
library(ggalluvial)       # 桑基图
library(ggquiver)         # 向量场图
library(ggmosaic)         # 马赛克图
library(ggbump)           # 凹凸图
library(ggstream)         # 水流图
library(timelineS)        # 时间线
library(ggdendro)         # 聚类图
library(ggfortify)        # 统计分析结果可视化：主成分图
library(gganimate)        # 动态图
```

David Robinson 给出为何使用 ggplot2 [1] 当然也有 Jeff Leek 指出在某些重要场合不适合 ggplot2 [2] 并且给出强有力的 证据，其实不管怎么样，适合自己的才是好的。也不枉费 Garrick Aden-Buie 花费 160 页幻灯片逐步分解介绍 优雅的 ggplot2，Malcolm Barrett 也介绍了 ggplot2 基础用法，还有 Selva Prabhakaran 精心总结给出了 50 个 ggplot2 数据可视化的 例子 以及 Victor Perrier 为小白用 ggplot2 操碎了心地开发

[1] http://varianceexplained.org/r/why-I-use-ggplot2/

[2] https://simplystatistics.org/2016/02/11/why-i-dont-use-ggplot2/

© 草稿

RStudio 插件 esquisse 包，Claus O. Wilke 教你一步步创建出版级的图形 https://github.com/clauswilke/practical_ggplot2。

ggplot2 是十分方便的统计作图工具，相比 Base R，为了一张出版级的图形，不需要去调整每个参数，实现快速出图。集成了很多其它统计计算的 R 包，支持丰富的统计分析和计算功能，如回归、平滑等，实现了作图和模型的无缝连接。比如图11.1，使用 loess 局部多项式平滑得到数据的趋势，不仅仅是散点图，代码量也非常少。

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = TRUE, method = "loess") +
  labs(
    title = "Fuel efficiency generally decreases with engine size",
    subtitle = "Two seaters (sports cars) are an exception because of their light weight",
    caption = "Data from fueleconomy.gov"
  )
```

图 11.1: 简洁美观

故事源于一幅图片，我不记得第一次见到这幅图是什么时候了，只因多次在多个场合中见过，所以留下了深刻的印象，后来才知道它出自于一篇博文 — Using R packages and education to scale Data Science at Airbnb，作者 Ricardo Bion 还在其 Github 上传了相关代码[3]。除此之外还有几篇重要的参考资料：

1. Pablo Barberá 的 Data Visualization with R and ggplot2

[3] https://github.com/ricardo-bion/medium_visualization

©黄湘云

2. Matt Leonawicz 的新作 mapmate, 可以去其主页欣赏系列作品[4]
3. tidytuesday 可视化挑战官方项目 还有 tidytuesday
4. ggstatsplot 可视化统计检验、模型的结果
5. ggpubr 制作出版级统计图形
6. Thomas Lin Pedersen Drawing Anything with ggplot2
7. Designing ggplots: making clear figures that communicate
8. ggh4x 提供 ggplot2 的额外定制功能
9. ggdist Visualizations of distributions and uncertainty
10. gghighlight
11. ggnetwork
12. ggPMX 'ggplot2' Based Tool to Facilitate Diagnostic Plots for NLME Models
13. ggpp ggpp: Grammar Extensions to 'ggplot2'

如 Berton Gunter 所说，数据可视化只是一种手段，根据数据实际情况作展示才是重要的，并不是要追求酷炫。

> 3-D bar plots are an abomination. Just because Excel can do them doesn't mean you should. (Dismount pulpit).
>
> — Berton Gunter [5]

grid 是 **lattice** 和 **ggplot2** 的基础，**gganimate** 是 ggplot2 一个扩展，它将静态图形视为帧，调用第三方工具合成 GIF 动图或 MP4 视频等，要想深入了解 ggplot2，可以去看 Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen 合著的《ggplot2: elegant graphics for data analysis》第三版 https://ggplot2-book.org/。

11.1 元素

以数据集 airquality 为例介绍 GGplot2 图层、主题、配色、坐标、尺度、注释和组合等

11.1.1 图层

```
ls("package:ggplot2", pattern = "^geom_")
```

```
##  [1] "geom_abline"            "geom_area"              "geom_bar"
##  [4] "geom_bin_2d"            "geom_bin2d"             "geom_blank"
##  [7] "geom_boxplot"           "geom_col"               "geom_contour"
## [10] "geom_contour_filled"    "geom_count"             "geom_crossbar"
## [13] "geom_curve"             "geom_density"           "geom_density_2d"
## [16] "geom_density_2d_filled" "geom_density2d"         "geom_density2d_filled"
## [19] "geom_dotplot"           "geom_errorbar"          "geom_errorbarh"
## [22] "geom_freqpoly"          "geom_function"          "geom_hex"
## [25] "geom_histogram"         "geom_hline"             "geom_jitter"
## [28] "geom_label"             "geom_line"              "geom_linerange"
```

[4] https://leonawicz.github.io/

[5] https://stat.ethz.ch/pipermail/r-help/2007-October/142420.html

© 王建湘

```
## [31] "geom_map"          "geom_path"          "geom_point"
## [34] "geom_pointrange"   "geom_polygon"       "geom_qq"
## [37] "geom_qq_line"      "geom_quantile"      "geom_raster"
## [40] "geom_rect"         "geom_ribbon"        "geom_rug"
## [43] "geom_segment"      "geom_sf"            "geom_sf_label"
## [46] "geom_sf_text"      "geom_smooth"        "geom_spoke"
## [49] "geom_step"         "geom_text"          "geom_tile"
## [52] "geom_violin"       "geom_vline"
```

生成一个散点图

```
ggplot(airquality, aes(x = Temp, y = Ozone)) + geom_point()
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```

11.1.2 标签

图形的标签分为横纵轴标签、刻度标签、主标题、副标题等

```
data.frame(
  dates = seq.Date(
    from = as.Date("1945-01-01"),
    to = as.Date("1974-12-31"),
    by = "quarter"
  ),
```

© 黄湘云


```
  presidents = as.vector(presidents)
) |>
  ggplot(aes(x = dates, y = presidents)) +
  geom_line(color = "slategray", na.rm = TRUE) +
  geom_point(size = 1.5, color = "darkslategray", na.rm = TRUE) +
  scale_x_date(date_breaks = "4 year", date_labels = "%Y") +
  labs(
    title = "1945年至1974年美国总统每季度支持率",
    x = "年份", y = "支持率 (%)",
    caption = "数据源: R 包 datasets"
  ) +
  theme_minimal(base_size = 10.54, base_family = "Noto Serif CJK SC")
```

图 11.2: 自 1945 年第一季度至 1974 年第四季度美国总统的支持

11.1.3 注释

图中注释的作用在于高亮指出关键点，提请读者注意。文本注释可由 **ggrepel** 包提供的标签图层 `geom_label_repel()` 添加，标签数据可独立于之前的数据层，标签所在的位置可以通过参数 `direction` 和 `nudge_y` 精调，图 11.3 模拟了一组数据。

```
set.seed(2020)
library(ggrepel)
dat <- data.frame(
  x = seq(100),
  y = cumsum(rnorm(100))
)
anno_data <- dat |>
  subset(x %% 25 == 10)  |>
```

© 基础统计

```
  transform(text = "text")

ggplot(data = dat, aes(x, y)) +
  geom_line() +
  geom_label_repel(aes(label = text),
    data = anno_data,
    direction = "y",
    nudge_y = c(-5, 5, 5, 5)
  ) +
  theme_minimal()
```

图 11.3: 文本注释

ggrepel 包的图层 `geom_text_repel()` 支持所有数据点的注释，并且自动调整文本的位置，防止重叠，增加辨识度，如图 11.4。当然，数据点如果过于密集也不适合全部注释，高亮其中的关键点即可。

```
mtcars |>
  transform(cyl = as.factor(cyl)) |>
  ggplot(aes(wt, mpg, label = rownames(mtcars), color = cyl)) +
  geom_point() +
  geom_text_repel(max.overlaps = 12) +
  theme_minimal()
```

Claus Wilke 开发的 ggtext 包支持更加丰富的注释样式，详见网站 https://wilkelab.org/ggtext/

```
ls("package:ggplot2", pattern = "^annotation_")
```

```
## [1] "annotation_custom"   "annotation_logticks" "annotation_map"
## [4] "annotation_raster"
```

```
ggplot(airquality, aes(x = Temp, y = Ozone)) +
  geom_point(na.rm = TRUE)
```

© 黄湘云

图 11.4: 少量点的情况下可以全部注释，且可以解决注释重叠的问题

```
ggplot(airquality, aes(x = Temp, y = Ozone)) +
  geom_point(na.rm = TRUE) +
  labs(title = substitute(paste(d *
    bolditalic(x)[italic(t)] == alpha * (theta - bolditalic(x)[italic(t)]) *
    d * italic(t) + lambda * d * italic(B)[italic(t)]), list(lambda = 4)))
```

© 基础

geomtextpath 曲线上的文字随曲线弯曲变化

ggsvg 曲线上散点以图片、彩色图标表示

11.1.4 刻度

```
ls("package:ggplot2", pattern = "^scale_(x|y)_")
```

```
##  [1] "scale_x_binned"     "scale_x_continuous" "scale_x_date"
##  [4] "scale_x_datetime"   "scale_x_discrete"   "scale_x_log10"
##  [7] "scale_x_reverse"    "scale_x_sqrt"       "scale_x_time"
## [10] "scale_y_binned"     "scale_y_continuous" "scale_y_date"
## [13] "scale_y_datetime"   "scale_y_discrete"   "scale_y_log10"
## [16] "scale_y_reverse"    "scale_y_sqrt"       "scale_y_time"
```

```
range(airquality$Temp, na.rm = TRUE)
```

```
## [1] 56 97
```

```
range(airquality$Ozone, na.rm = TRUE)
```

```
## [1]   1 168
```

```
ggplot(airquality, aes(x = Temp, y = Ozone)) +
  geom_point(na.rm = TRUE) +
  scale_x_continuous(breaks = seq(50, 100, 5)) +
```

© 黄湘云


```
  scale_y_continuous(breaks = seq(0, 200, 20))
```

11.1.5 图例

二维的图例 biscale 和 multiscales 和 ggnewscale

11.1.6 坐标系

极坐标，直角坐标

```
ls("package:ggplot2", pattern = "^coord_")
```

```
##  [1] "coord_cartesian" "coord_equal"     "coord_fixed"     "coord_flip"
##  [5] "coord_map"       "coord_munch"     "coord_polar"     "coord_quickmap"
##  [9] "coord_sf"        "coord_trans"
```

11.1.7 坐标轴

坐标轴标签位置、大小、字体

© 王道图书

11.1.8 配色

```
ls("package:ggplot2", pattern = "^scale_(color|fill)_")
```

```
##  [1] "scale_color_binned"     "scale_color_brewer"     "scale_color_continuous"
##  [4] "scale_color_date"       "scale_color_datetime"   "scale_color_discrete"
##  [7] "scale_color_distiller"  "scale_color_fermenter"  "scale_color_gradient"
## [10] "scale_color_gradient2"  "scale_color_gradientn"  "scale_color_grey"
## [13] "scale_color_hue"        "scale_color_identity"   "scale_color_manual"
## [16] "scale_color_ordinal"    "scale_color_steps"      "scale_color_steps2"
## [19] "scale_color_stepsn"     "scale_color_viridis_b"  "scale_color_viridis_c"
## [22] "scale_color_viridis_d"  "scale_fill_binned"      "scale_fill_brewer"
## [25] "scale_fill_continuous"  "scale_fill_date"        "scale_fill_datetime"
## [28] "scale_fill_discrete"    "scale_fill_distiller"   "scale_fill_fermenter"
## [31] "scale_fill_gradient"    "scale_fill_gradient2"   "scale_fill_gradientn"
## [34] "scale_fill_grey"        "scale_fill_hue"         "scale_fill_identity"
## [37] "scale_fill_manual"      "scale_fill_ordinal"     "scale_fill_steps"
## [40] "scale_fill_steps2"      "scale_fill_stepsn"      "scale_fill_viridis_b"
## [43] "scale_fill_viridis_c"   "scale_fill_viridis_d"
```

```
ggplot(airquality, aes(x = Temp, y = Ozone, color = as.factor(Month))) +
  geom_point(na.rm = TRUE)
```

©黄海广


```
ggplot(airquality, aes(x = Temp, y = Ozone, color = as.ordered(Month))) +
  geom_point(na.rm = TRUE)
```

11.1.9 主题

ggcharts 和 bbplot prettyB 美化 Base R 图形 ggprism

```
ls("package:ggplot2", pattern = "^theme_")
```

```
##  [1] "theme_bw"       "theme_classic"  "theme_dark"     "theme_get"
##  [5] "theme_gray"     "theme_grey"     "theme_light"    "theme_linedraw"
##  [9] "theme_minimal"  "theme_replace"  "theme_set"      "theme_test"
## [13] "theme_update"   "theme_void"
```

这里只展示 `theme_bw()` `theme_void()` `theme_minimal()` 和 `theme_void()` 等四个常见主题，更多主题参考 ggsci、ggthemes 、ggtech、hrbrthemes 和 ggthemr 包

```
ggplot(airquality, aes(x = Temp, y = Ozone)) + geom_point() + theme_bw()
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```

```
ggplot(airquality, aes(x = Temp, y = Ozone)) + geom_point() + theme_void()
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```

```
ggplot(airquality, aes(x = Temp, y = Ozone)) + geom_point() + theme_minimal()
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```

```
ggplot(airquality, aes(x = Temp, y = Ozone)) + geom_point() + theme_classic()
```

```
## Warning: Removed 37 rows containing missing values (geom_point).
```

(a) 黑白主题

(b) 无主题

(c) 极少配置的主题

(d) 经典主题

图 11.5: ggplot2 内置的主题

除主题之外，还有一类提供一整套统一的风格样式来绘制各种统计图形，如 ggpubr 和 bbplot

11.1.10 布局

```
ggplot(airquality) +
  geom_point(aes(x = Temp, y = Ozone), na.rm = TRUE) +
  facet_wrap(~ as.ordered(Month))
```

© 黄湘云

```
ggplot(airquality) +
  geom_point(aes(x = Temp, y = Ozone), na.rm = TRUE) +
  facet_wrap(~ as.ordered(Month), nrow = 1)
```

cowplot 是以作者 Claus O. Wilke 命名的，用来组合 ggplot 对象画图，类似的组合图形的功能包还有 baptiste auguié 开发的 gridExtra 和 egg，Thomas Lin Pedersen 开发的 patchwork

Dean Attali 开发的 ggExtra 可以在图的边界添加密度估计曲线，直方图等

11.2 字体

firatheme 包提供基于 fira sans 字体的 ggplot2 主题，类似的字体主题包还有 trekfont 、fontHind，fontquiver 包与 fontBitstreamVera（Bitstream Vera 字体）、fontLiberation（Liberation 字体）包和 fontDejaVu （DejaVu 字体）包一道提供了一些可允许使用的字体文件，这样，我们可以不依赖系统制作可重复的图形。Thomas Lin Pedersen 开发的 systemfonts 可直接使用系统自带的字体。

© 草稿

11.2.1 系统字体

以 CentOS 系统为例，软件仓库中包含 Noto ，DejaVu 、liberation 等字体。可以安装自己喜欢的字体类型，比如：

```
sudo dnf install -y \
  google-noto-mono-fonts \
  google-noto-sans-fonts \
  google-noto-serif-fonts \
  dejavu-sans-mono-fonts \
  dejavu-sans-fonts \
  dejavu-serif-fonts
# 或者
sudo dnf install -y dejavu-fonts liberation-fonts
```

liberation 系列的四款字体可以用来替换 Windows 系统上对应的四款字体，对应关系见表 11.1

表 11.1: Windows 系统上四款字体的替代品

	CentOS 系统	Windows 系统
衬线体/宋体	liberation-serif-fonts	Times New Roman
无衬线体/黑体	liberation-sans-fonts	Arial
Arial 的细瘦版	liberation-narrow-fonts	Arial Narrow
等宽体/微软雅黑	liberation-mono-fonts	Courier New

Lionel Henry 将 Liberation 系列字体打包到 R 包 fontLiberation，非常便携，不需要操心跨平台的字体安装了。那如何使用呢？

```
# install.packages("fontLiberation")
system.file(package = "fontLiberation", "fonts", "liberation-fonts")
```

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

此外，我们还可以从网上获取各种个样的字体，特别地，Boryslav Larin 收录的 awesome-fonts 列表是一个不错的开始，比如图标字体 Font-Awesome，

```
sudo dnf install -y fontawesome-fonts
```

再安装宏包 fontawesome 后，即可在 LaTeX 文档中使用，下面这个示例推荐用 XeLaTeX 引擎编译。

```
\documentclass[border=10pt]{standalone}
\usepackage{fontawesome}
\begin{document}
Hello, \faGithub
\end{document}
```

而在 R 绘制的图形中，通过指定 `par()`、`plot()`、`title()` 等函数的 `family` 参数值，比如 `family = "Liberation Sans"` 来调用系统无衬线 Liberation 字体，效果见图 11.6。

© 黄湘云


```
library(extrafont)
plot(data = pressure, pressure ~ temperature,
     xlab = "Temperature (deg C)", ylab = "Pressure (mm of Hg)",
     col.lab = "red", col.axis = "blue",
     font.lab = 3, font.axis = 2, family = "Liberation Sans")
title(main = "Vapor Pressure of Mercury as a Function of Temperature",
      family = "Liberation Serif", font.main = 3)
title(sub = "Data Source: Weast, R. C",
      family = "Liberation Mono", font.sub = 1)
```

图 11.6: 调用系统字体绘图

为了符合出版的要求，需要在 11.6 中嵌入字体，

```
# embed fonts to pdf
embed_fonts <- function(fig_path) {
  if(knitr::is_latex_output()){
    embedFonts(
      file = fig_path, outfile = fig_path,
      fontpaths = "~/Library/Fonts"
    )
  }
  return(fig_path)
}
```

设置代码块选项 `fig.process=embed_fonts`，这样生成 PDF 格式图形的时候，会调用此函数处理 PDF 图形。在 ggplot2 绘图中的调用方式是类似的，便不再赘述了。值得注意的是，extrafont 和 showtext 有些不一样，前者只能处理系统字体，后者还能获取网络字体和使用 OTF 字体，下面从 Google 开源的字体库获取 Noto 系列的四款字体，如图 11.7。

© 基础统计

```
sysfonts::font_add_google(name = "Noto Sans", family = "Noto Sans")
sysfonts::font_add_google(name = "Noto Serif", family = "Noto Serif")
sysfonts::font_add_google(name = "Noto Serif SC", family = "Noto Serif SC")
sysfonts::font_add_google(name = "Noto Sans SC", family = "Noto Sans SC")
```

警告

在本书中，不要全局加载 showtext 包或调用 `showtext::showtext_auto()`，会和 extrafont 冲突，使得绘图时默认就只能使用 showtext 提供的字体。extrafont 包提供的函数 `font_import()` 仅支持系统安装的 TrueType/Type1 字体

```
p1 <- ggplot(pressure, aes(x = temperature, y = pressure)) +
  geom_point() +
  ggtitle(label = "默认字体设置")

p2 <- p1 + theme(
  axis.title = element_text(family = "Noto Sans"),
  axis.text = element_text(family = "Noto Serif")
) +
  theme(
    title = element_text(family = "Noto Serif SC")
  ) +
  ggtitle(label = "英文字体设置")

p3 <- p1 + labs(x = "温度", y = "压力") +
  theme(
    axis.title = element_text(family = "Noto Serif SC"),
    axis.text = element_text(family = "Noto Serif")
  ) +
  ggtitle(label = "中文字体设置")

p4 <- p1 + labs(
  x = "温度", y = "压力", title = "散点图",
  subtitle = "Vapor Pressure of Mercury as a Function of Temperature",
  caption = paste("Data on the relation
                  between temperature in degrees Celsius and",
    "vapor pressure of mercury in millimeters (of mercury).",
    sep = "\n"
  )
) +
  theme(
    axis.title = element_text(family = "Noto Serif SC"),
    axis.text.x = element_text(family = "Noto Serif"),
    axis.text.y = element_text(family = "Noto Sans"),
    title = element_text(family = "Noto Serif SC"),
    plot.subtitle = element_text(family = "Noto Sans", size = rel(0.7)),
```

© 黄湘云


```
    plot.caption = element_text(family = "Noto Sans", size = rel(0.6))
  ) +
  ggtitle(label = "任意字体设置")

(p1 + p2) / (p3 + p4)
```

图 11.7: 在 ggplot2 绘图系统中设置中英文字体

另外值得一提的是 hrbrthemes 包，除了定制了很多 ggplot2 主题，它还打包了很多的字体主题。比如默认主题 `theme_ipsum()` 使用 Arial Narrow 字体，如果没有该字体就自动寻找系统中的替代品，如图 11.8 实际使用的是 Nimbus Sans Narrow 字体，因为在 GitHub Action 中，我实际使用的测试环境是 Ubuntu 20.04，该系统自带 Nimbus Sans Narrow 字体，Arial Narrow 毕竟是 Windows 上的闭源字体。

```
# brew install font-roboto
# 导入字体
# hrbrthemes::import_roboto_condensed()
sysfonts::font_add_google(name = "Roboto Condensed", family = "Roboto Condensed")
```

© 基础统计

```
library(hrbrthemes)
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(
    x = "Fuel efficiency (mpg)", y = "Weight (tons)",
    title = "Seminal ggplot2 scatterplot example",
    subtitle = "A plot that is only useful for demonstration purposes",
    caption = "Brought to you by the letter 'g'"
  ) +
  theme_ipsum(base_family = "Roboto Condensed")
```

图 11.8: 调用 hrbrthemes 包设置字体主题

如果系统没有安装 Arial Narrow 字体，可以导入 hrbrthemes 包自带的一些字体，比如 `hrbrthemes::import_roboto_condense` 然后调用字体主题 `theme_ipsum_rc()`。如果不想使用这个包自带的字体，可以用系统中安装的字体去修改主题 `theme_ipsum()` 和 `theme_ipsum_rc()` 中的字体设置。如图 11.9 使用了 `theme_ipsum()` 中的 Arial Narrow 字体。

```
ggplot(mtcars, aes(mpg, wt)) +
  geom_point() +
  labs(
    x = "Fuel efficiency (mpg)", y = "Weight (tons)",
    title = "Seminal ggplot2 scatterplot example",
    subtitle = "A plot that is only useful for demonstration purposes",
    caption = "Brought to you by the letter 'g'"
  ) +
  theme_ipsum(base_family = "Noto Sans")
```

© 黄湘云

图 11.9: 默认字体 Arial Narrow

提示

hrbrthemes 包提供了一个全局字体加载选项 `hrbrthemes.loadfonts`，如果设置为 TRUE，即 `options(hrbrthemes.loadfonts = TRUE)` 会先调用函数 `extrafont::loadfonts()` 预加载系统字体，就不用一次次手动加载字体了。后续在第 11.2.3 节还会提及 extrafont 包的其它功能。

11.2.2 思源字体

邱怡轩开发的 showtext 包支持丰富的外部字体，支持 Base R 和 ggplot2 图形，图 11.10 嵌入了 5 号思源宋体，图例和坐标轴文本使用 serif 字体，更多详细的使用文档见 [Qiu, 2015]。

```
# 安装 showtext 包
install.packages('showtext')
# 思源宋体
showtextdb::font_install(showtextdb::source_han_serif())
# 思源黑体
showtextdb::font_install(showtextdb::source_han_sans())
```

```
# ggplot(iris, aes(Sepal.Length, Sepal.Width)) +
#   geom_point(aes(colour = Species)) +
#   scale_colour_brewer(palette = "Set1") +
#   labs(
#     title = "鸢尾花数据的散点图",
#     x = "萼片长度", y = "萼片宽度", colour = "鸢尾花类别",
#     caption = "鸢尾花数据集最早见于 Edgar Anderson (1935) "
```

© 黄湘云


```
#   ) +
#   theme(
#     title = element_text(family = "source-han-sans-cn"),
#     axis.title = element_text(family = "source-han-serif-cn"),
#     legend.title = element_text(family = "source-han-serif-cn")
#   )
ggplot(iris, aes(Sepal.Length, Sepal.Width)) +
  geom_point(aes(colour = Species)) +
  scale_colour_brewer(palette = "Set1") +
  labs(
    title = "鸢尾花数据的散点图",
    x = "萼片长度", y = "萼片宽度", colour = "鸢尾花类别",
    caption = "鸢尾花数据集最早见于 Edgar Anderson (1935) "
  ) +
  theme(
    title = element_text(family = "Noto Sans SC"),
    axis.title = element_text(family = "Noto Serif SC"),
    legend.title = element_text(family = "Noto Serif SC")
  )
```

图 11.10: showtext 包处理图里的中文

斐济是太平洋上的一个岛国，受地壳板块运动的影响，地震活动频繁，图 11.11 清晰展示了它的地震带。

```
library(maps)
library(mapdata)
FijiMap <- map_data("worldHires", region = "Fiji")
ggplot(FijiMap, aes(x = long, y = lat)) +
```

©黄湘云


```
  geom_map(map = FijiMap, aes(map_id = region), size = .2) +
  geom_point(data = quakes, aes(x = long, y = lat, colour = mag)) +
  xlim(160, 195) +
  scale_colour_distiller(palette = "Spectral") +
  scale_y_continuous(breaks = (-18:18) * 5) +
  coord_map("ortho", orientation = c(-10, 180, 0)) +
  labs(colour = "震级", x = "经度", y = "纬度", title = "斐济地震带") +
  theme_minimal(base_family = "Noto Serif SC")
```

图 11.11: 斐济地震带

11.2.3 数学字体

Winston Chang 将 Paul Murrell 的 Computer Modern 字体文件打包成 fontcm 包 [Chang et al., 2014]，**fontcm** 包可以在 Base R 图形中嵌入数学字体 [6]，图形中嵌入重音字符 [7]。下面先下载、安装、加载字体，

```
library(extrafont)
font_addpackage(pkg = "fontcm")
```

查看可被 `pdf()` 图形设备使用的字体列表

```
# 可用的字体
fonts()
```

```
##  [1] "CM Roman"              "CM Roman Asian"        "CM Roman CE"
##  [4] "CM Roman Cyrillic"     "CM Roman Greek"        "CM Sans"
```

[6] https://www.stat.auckland.ac.nz/~paul/R/CM/CMR.html
[7] https://www.stat.auckland.ac.nz/~paul/Reports/maori/maori.html

```
##  [7] "CM Sans Asian"          "CM Sans CE"             "CM Sans Cyrillic"
## [10] "CM Sans Greek"          "CM Symbol"              "CM Typewriter"
## [13] "CM Typewriter Asian"    "CM Typewriter CE"       "CM Typewriter Cyrillic"
## [16] "CM Typewriter Greek"
```

fontcm 包提供数学字体，`grDevices::embedFonts()` 函数调用 Ghostscript 软件将数学字体嵌入 ggplot2 图形中，达到正确显示数学公式的目的，此方法适用于 pdf 设备保存的图形，对 `cairo_pdf()` 保存的 PDF 格式图形无效。

```r
library(fontcm)
library(ggplot2)
library(extrafont)
library(patchwork)
p <- ggplot(
  data = data.frame(x = c(1, 5), y = c(1, 5)),
  aes(x = x, y = y)
) +
  geom_point() +
  labs(
    x = "Made with CM fonts", y = "Made with CM fonts",
    title = "Made with CM fonts"
  )
# 公式
eq <- "italic(sum(frac(1, n*'!'), n==0, infinity) ==
       lim(bgroup('(', 1 + frac(1, n), ')')^n, n %->% infinity))"
# 默认字体
p1 <- p + annotate("text",
  x = 3, y = 3,
  parse = TRUE, label = eq # , family = "CM Roman"
)
# 使用 CM Roman 字体
p2 <- p + annotate("text",
  x = 3, y = 3,
  parse = TRUE, label = eq, family = "CM Roman"
) +
  theme(
    text = element_text(size = 10, family = "CM Roman"),
    axis.title.x = element_text(face = "italic"),
    axis.title.y = element_text(face = "bold")
  )
p1 + p2
```

为实现图 11.12 的最终效果，需要启用一个有超级牛力的 fig.process 选项，主要是传递一个函数给它，对用 R 语言生成的图形再操作。

```r
# embed math fonts to pdf
embed_math_fonts <- function(fig_path) {
```

© 黄湘云

图 11.12: fontcm 处理数学公式

```
  if(knitr::is_latex_output()){
    embedFonts(
      file = fig_path, outfile = fig_path,
      fontpaths = system.file("fonts", package = "fontcm")
    )
  }
  return(fig_path)
}
```

代码块选项中设置 `fig.process=embed_math_fonts` 可在绘图后，立即插入字体，此操作仅限于以 pdf 格式保存的图形设备，也适用于 Base R 绘制的图形，见图 11.13。

```
par(mar = c(4.1, 4.1, 1.5, 0.5), family = "CM Roman")
x <- seq(-4, 4, len = 101)
y <- cbind(sin(x), cos(x))
matplot(x, y,
  type = "l", xaxt = "n",
  main = expression(paste(
    plain(sin) * phi, "  and  ",
    plain(cos) * phi
  )),
  ylab = expression("sin" * phi, "cos" * phi),
  xlab = expression(paste("Phase Angle ", phi)),
  col.main = "blue"
)
axis(1,
  at = c(-pi, -pi / 2, 0, pi / 2, pi),
  labels = expression(-pi, -pi / 2, 0, pi / 2, pi)
)
```

© 黄湘云

图 11.13: 嵌入数学字体

11.2.4 TikZ 设备

与 11.2.3 小节不同，Ralf Stubner 维护的 **tikzDevice** 包提供了另一种嵌入数学字体的方式，其提供的 `tikzDevice::tikz()` 绘图设备将图形对象转化为 TikZ 代码，调用 LaTeX 引擎编译成 PDF 文档。安装后，先测试一下 LaTeX 编译环境是否正常。

```
tikzDevice::tikzTest()
```

```
##
## Active compiler:
##  /home/runner/.TinyTeX/bin/x86_64-linux/xelatex
##  XeTeX 3.141592653-2.6-0.999994 (TeX Live 2022)
##  kpathsea version 6.3.4
```

```
## [1] 7.90259
```

确认没有问题后，下面图 11.14 的坐标轴标签，标题，图例等位置都支持数学公式，使用 **tikzDevice** 打造出版级的效果图。更多功能的介绍见 https://www.daqana.org/tikzDevice/。

```
x <- rnorm(10)
y <- x + rnorm(5, sd = 0.25)
model <- lm(y ~ x)
rsq <- summary(model)$r.squared
rsq <- signif(rsq, 4)
plot(x, y,
  main = "Hello \\LaTeX!", xlab = "$x$", ylab = "$y$",
  sub = "$\\mathcal{N}(x;\\mu,\\Sigma)$"
)
abline(model, col = "red")
```

© 黄湘云


```
mtext(paste0("Linear model: $R^{2}=", rsq, "$"), line = 0.5)
legend("bottomright",
  legend = paste0(
    "$y = ",
    round(coef(model)[2], 3),
    "x +",
    round(coef(model)[1], 3),
    "$"
  ),
  bty = "n"
)
```

图 11.14: 线性回归模型

推荐的全局 LaTeX 环境配置如下：

```
options(
  tinytex.engine = "xelatex",
  tikzDefaultEngine = "xetex",
  tikzDocumentDeclaration = "\\documentclass[tikz]{standalone}\n",
  tikzXelatexPackages = c(
    "\\usepackage[fontset=adobe]{ctex}",
    "\\usepackage[default,semibold]{sourcesanspro}",
    "\\usepackage{amsfonts,mathrsfs,amssymb}\n"
  )
)
```

设置默认的 LaTeX 编译引擎为 XeLaTeX，相比于 PDFLaTeX，它对中文的兼容性更好，支持多平台下的中文环境，中文字体这里采用了 Adobe 的字体，默认加载了 mathrsfs 宏包支持 `\mathcal`、`\mathscr` 等命令，此外，LaTeX 发行版采用谢益辉自定义的 TinyTeX。绘制独立的 PDF 图形的过程如下：

```
library(tikzDevice)
tf <- file.path(getwd(), "tikz-regression.tex")
tikz(tf, width = 6, height = 5.5, pointsize = 30, standAlone = TRUE)
# 绘图代码
dev.off()
# 编译成 PDF 图形
tinytex::latexmk(file = "tikz-regression.tex")
```

11.2.5 漫画字体

下载 XKCD 字体，并刷新系统字体缓存

```
mkdir -p ~/.fonts
curl -fLo ~/.fonts/xkcd.ttf http://simonsoftware.se/other/xkcd.ttf
fc-cache -fsv
```

将 XKCD 字体导入到 R 环境，以便后续被 ggplot2 图形设备调用。

```
R -e 'library(extrafont);font_import(pattern="[X/x]kcd.ttf", prompt = FALSE)'
```

下图是一个使用 xkcd 字体的简单例子，更多高级特性请看 **xkcd** 包文档 [Torres-Manzanera, 2018]

```
library(xkcd)
ggplot(aes(mpg, wt), data = mtcars) +
  geom_point() +
  theme_xkcd()
```

11.2.6 表情字体

余光创开发的 emojifont 包和 Hadley 开发的 emo 包，下面使用 Noto Emoji 字体，支持的表情图见 https://www.google.com/get/noto/help/emoji/food-drink/，下面给出一个示例。先从 GitHub 安装 **emo** 包，目前它还未正式发布到 CRAN 上。

```
remotes::install_github("hadley/emo")
```

除了安装 emo 包，系统需要先安装好 emoji 字体，图形才会正确地渲染出来，想调用更多 emoji 图标请参考 Emoji 速查手册，给出 emoji 对应的名字。

```
# CentOS
sudo dnf install -y google-noto-emoji-color-fonts \
  google-noto-emoji-fonts
# MacOS
brew cask install font-noto-color-emoji font-noto-emoji
```

© 黄湘云


```
data.frame(
  category = c("pineapple", "apple", "watermelon", "mango", "pear"),
  value = c(5, 4, 3, 6, 2)
) |>
  transform(category = sapply(category, emo::ji)) |>
  ggplot(aes(x = category, y = value)) +
  scale_y_continuous(limits = c(2, 7)) +
  geom_text(aes(label = category), size = 12, vjust = -0.5) +
  theme_minimal()
```

图 11.15: 表情字体

Noto Color Emoji 字体在 MacOS 上有问题，为了跨平台的便携性，提供 emojifont 包的例子，要引入更多的依赖。

```
library(ggplot2)
library(emojifont)

names <- c("smile", "school", "office", "blush", "smirk", "heart_eyes")
n <- length(names):1
e <- sapply(names, emojifont::emoji)
dat <- data.frame(emoji_name = names, n = n, emoji = e, stringsAsFactors = F)

ggplot(data = dat, aes(emoji_name, n)) +
  geom_bar(stat = "identity") +
  scale_x_discrete(breaks = dat$emoji_name, labels = dat$emoji) +
  theme(axis.text.y = element_text(size = 20, family = "EmojiOne")) +
```

© 黄湘云

```
coord_flip()
```

11.3 配色

配色真的是一门学问，有的人功力非常深厚，仅用黑白灰就可以创造出一个世界，如中国的水墨画，科波拉执导的《教父》，沃卓斯基姐妹执导的《黑客帝国》等。黑西装、白衬衫和黑领带是《黑客帝国》的经典元素，《教父》开场的黑西装、黑领结和白衬衫，尤其胸前的红玫瑰更是点睛之笔。导演将黑白灰和光影混合形成了层次丰富立体的画面，打造了一场视觉盛宴，无论是呈现在纸上还是银幕上都可以给人留下深刻的印象。正所谓食色性也，花花世界，岂能都是法印眼中的白骨！再说《红楼梦》里，芍药丛中，桃花树下，滴翠亭边，栊翠庵里，处处都是湘云、黛玉、宝钗、妙玉留下的四季诗歌。

为什么需要这么多颜色模式呢？主要取决于颜色输出的通道，比如印刷机，照相机，自然界，网页，人眼等，显示器因屏幕和分辨率的不同呈现的色彩数量是不一样的。读者大概都听说过 RGB、CMYK、AdobeRGB、sRGB、P3 广色域等名词，我想这主要归功于各大电子设备厂商的宣传。普清、高清、超高清、全高清、2K、4K、5K、视网膜屏，而 HSV、HCL 估计听说的人就少很多了。本节的目的是简单阐述背后的色彩原理，颜色模式及其之间的转化，在应对天花乱坠的销售时少交一些智商税，同时，告诉读者如何在 R 环境中使用色彩。早些时候我在统计之都论坛上发帖 – R 语言绘图用调色板大全 https://d.cosx.org/d/419378，如果读者希望拿来即用，不妨去看看。

```
filled.contour(volcano, nlevels = 10, color.palette = terrain.colors)
filled.contour(volcano, nlevels = 10, color.palette = heat.colors)
filled.contour(volcano, nlevels = 10, color.palette = topo.colors)
filled.contour(volcano, nlevels = 10, color.palette = cm.colors)
```

```
filled.contour(volcano,
  nlevels = 10,
  color.palette = function(n, ...) hcl.colors(n, "Grays", rev = TRUE, ...)
)
filled.contour(volcano,
  nlevels = 10,
  color.palette = function(n, ...) hcl.colors(n, "YlOrRd", rev = TRUE, ...)
)
filled.contour(volcano,
  nlevels = 10,
  color.palette = function(n, ...) hcl.colors(n, "purples", rev = TRUE, ...)
)
filled.contour(volcano,
  nlevels = 10,
  color.palette = function(n, ...) hcl.colors(n, "viridis", rev = FALSE, ...)
)
```

注意

`hcl.colors()` 函数是在 R 3.6.0 引入的，之前的 R 软件版本中没有，同时内置了 110 个调色板，详见 `hcl.pals()`。

© 黄湘云

(a) terrain.colors 调色板

(b) heat.colors 调色板

(c) topo.colors 调色板

(d) cm.colors 调色板

图 11.16: R 3.6.0 以前的调色板

© 王建国

(a) Grays 调色板

(b) YlOrRd 调色板

(c) Purples 3 调色板

(d) Viridis 调色板

图 11.17: R 3.6.0 以后的调色板

© 王建湘

11.3.1 调色板

R 预置的灰色有 224 种，挑出其中的调色板

```
grep("^gr(a|e)y", grep("gr(a|e)y", colors(), value = TRUE),
     value = TRUE, invert = TRUE)
```

```
##  [1] "darkgray"       "darkgrey"       "darkslategray"  "darkslategray1"
##  [5] "darkslategray2" "darkslategray3" "darkslategray4" "darkslategrey"
##  [9] "dimgray"        "dimgrey"        "lightgray"      "lightgrey"
## [13] "lightslategray" "lightslategrey" "slategray"      "slategray1"
## [17] "slategray2"     "slategray3"     "slategray4"     "slategrey"
```

```
gray_colors <- paste0(rep(c("slategray", "darkslategray"), each = 4), seq(4))
barplot(1:8, col = gray_colors, border = NA)
```

图 11.18: 灰度调色板

gray 与 grey 是一样的，类似 color 和 colour 的关系，可能是美式和英式英语的差别，且看

```
all.equal(
  col2rgb(paste0("gray", seq(100))),
  col2rgb(paste0("grey", seq(100)))
)
```

```
## [1] TRUE
```

`gray100` 代表白色，`gray0` 代表黑色，提取灰色调色板，去掉首尾部分是必要的

© 黄湘云

```
barplot(1:8,
  col = gray.colors(8, start = .3, end = .9),
  main = "gray.colors function", border = NA
)
```

图 11.19: 提取 10 种灰色做调色板

首先选择一组合适的颜色，比如从桃色到梨色，选择 6 种颜色，以此为基础，可以借助 `grDevices::colorRampPalette()` 函数扩充至想要的数目，用 `graphics::rect()` 函数预览这组颜色配制的调色板

```
# Colors from https://github.com/johannesbjork/LaCroixColoR
colors_vec <- c("#FF3200", "#E9A17C", "#E9E4A6",
                "#1BB6AF", "#0076BB", "#172869")
# 代码来自 ?colorspace::rainbow_hcl
pal <- function(n = 20, colors = colors, border = "light gray", ...) {
  colorname <- (grDevices::colorRampPalette(colors))(n)
  plot(0, 0,
    type = "n", xlim = c(0, 1), ylim = c(0, 1),
    axes = FALSE, ...
  )
  rect(0:(n - 1) / n, 0, 1:n / n, 1, col = colorname, border = border)
}
par(mar = rep(0, 4))
pal(n = 20, colors = colors_vec, xlab = "Colors from Peach to Pear", ylab = "")
```

© 黄湘云

图 11.20: 桃色至梨色的渐变

colorRampPalette() 自制调色板

```
create_palette <- function(n = 1000, colors = c("blue", "orangeRed")) {
  color_palette <- colorRampPalette(colors)(n)
  barplot(rep(1, times = n), col = color_palette,
          border = color_palette, axes = FALSE)
}
par(mfrow = c(3, 1), mar = c(0.1, 0.1, 0.5, 0.1), xaxs = "i", yaxs = "i")
create_palette(n = 1000, colors = c("blue", "orangeRed"))
create_palette(n = 1000, colors = c("darkgreen", "yellow", "orangered"))
create_palette(n = 1000, colors = c("blue", "white", "orangered"))
```

图 11.21: colorRampPalette 自制调色板

```
par(mar = c(0, 4, 0, 0))
RColorBrewer::display.brewer.all()
```

```
# 代码来自 ?palettes
demo.pal <- function(n, border = if (n < 32) "light gray" else NA,
           main = paste("color palettes: alpha = 1,  n=", n),
           ch.col = c(
             "rainbow(n, start=.7, end=.1)", "heat.colors(n)",
             "terrain.colors(n)", "topo.colors(n)",
             "cm.colors(n)", "gray.colors(n, start = 0.3, end = 0.9)"
           )) {
    nt <- length(ch.col)
    i <- 1:n
    j <- n / nt
```

© 黄湘云

图 11.22: RColorBrewer 调色板

© 王菲菲


```
    d <- j / 6
    dy <- 2 * d
    plot(i, i + d, type = "n", axes = FALSE, ylab = "", xlab = "", main = main)
    for (k in 1:nt) {
      rect(i - .5, (k - 1) * j + dy, i + .4, k * j,
        col = eval(parse(text = ch.col[k])), border = border
      )
      text(2 * j, k * j + dy / 4, ch.col[k])
    }
  }
n <- if (.Device == "postscript") 64 else 16
# Since for screen, larger n may give color allocation problem
par(mar = c(0, 0, 2, 0))
demo.pal(n)
```

图 11.23: grDevices 调色板

```
par(mfrow = c(33, 1), mar = c(0, 0, .8, 0))
for (i in seq(32)) {
```

© 基础统计

```
  pal(
    n = length((1 + 20 * (i - 1)):(20 * i)),
    colors()[(1 + 20 * (i - 1)):(20 * i)],
    main = paste(1 + 20 * (i - 1), "to", 20 * i)
  )
}
pal(n = 17, colors()[641:657], main = "641 to 657")
```

```
library(colorspace)
## a few useful diverging HCL palettes
par(mar = c(0,0,2,0), mfrow = c(16, 2))

pal(n = 16, diverge_hcl(16), main = "diverging HCL palettes")
pal(n = 16, diverge_hcl(16, h = c(246, 40), c = 96, l = c(65, 90)))
pal(n = 16, diverge_hcl(16, h = c(130, 43), c = 100, l = c(70, 90)))
pal(n = 16, diverge_hcl(16, h = c(180, 70), c = 70, l = c(90, 95)))

pal(n = 16, diverge_hcl(16, h = c(180, 330), c = 59, l = c(75, 95)))
pal(n = 16, diverge_hcl(16, h = c(128, 330), c = 98, l = c(65, 90)))
pal(n = 16, diverge_hcl(16, h = c(255, 330), l = c(40, 90)))
pal(n = 16, diverge_hcl(16, c = 100, l = c(50, 90), power = 1))

## sequential palettes
pal(n = 16, sequential_hcl(16), main= "sequential palettes")
pal(n = 16, heat_hcl(16, h = c(0, -100),
                     l = c(75, 40), c = c(40, 80), power = 1))
pal(n = 16, terrain_hcl(16, c = c(65, 0), l = c(45, 95), power = c(1/3, 1.5)))
pal(n = 16, heat_hcl(16, c = c(80, 30), l = c(30, 90), power = c(1/5, 1.5)))

## compare base and colorspace palettes
## (in color and desaturated)
## diverging red-blue colors
pal(n = 16, diverge_hsv(16), main = "diverging red-blue colors")
pal(n = 16, diverge_hcl(16, c = 100, l = c(50, 90)))
pal(n = 16, desaturate(diverge_hsv(16)))
pal(n = 16, desaturate(diverge_hcl(16, c = 100, l = c(50, 90))))

## diverging cyan-magenta colors
pal(n = 16, cm.colors(16), main = "diverging cyan-magenta colors")
pal(n = 16, diverge_hcl(16, h = c(180, 330), c = 59, l = c(75, 95)))
pal(n = 16, desaturate(cm.colors(16)))
pal(n = 16, desaturate(diverge_hcl(16, h = c(180, 330), c = 59, l = c(75, 95))))

## heat colors
pal(n = 16, heat.colors(16), main = "heat colors")
```

© 黄湘云

图 11.24: grDevices 调色板

© 清华大学出版社

```
pal(n = 16, heat_hcl(16))
pal(n = 16, desaturate(heat.colors(16)))
pal(n = 16, desaturate(heat_hcl(16)))

## terrain colors
pal(n = 16, terrain.colors(16), main = "terrain colors")
pal(n = 16, terrain_hcl(16))
pal(n = 16, desaturate(terrain.colors(16)))
pal(n = 16, desaturate(terrain_hcl(16)))

pal(n = 16, rainbow_hcl(16, start = 30, end = 300), main = "dynamic")
pal(n = 16, rainbow_hcl(16, start = 60, end = 240), main = "harmonic")
pal(n = 16, rainbow_hcl(16, start = 270, end = 150), main = "cold")
pal(n = 16, rainbow_hcl(16, start = 90, end = -30), main = "warm")
```

除之前提到的 **grDevices** 包，**colorspace** (https://hclwizard.org/) 包 [Stauffer et al., 2009, Zeileis et al., 2009, 2019]，RColorBrewer 包 [Neuwirth, 2014] https://colorbrewer2.org/，viridis 包、colourvalues、wesanderson、dichromat 包、pals 包，palr 包，colorRamps 包、ColorPalette 包、colortools 包就不一一详细介绍了。

colormap 包基于 node.js 的 colormap 模块提供 44 个预定义的调色板 paletteer 包收集了很多 R 包提供的调色板，同时也引入了很多依赖。根据电影 Harry Potter 制作的调色板 harrypotter，根据网站 CARTO 设计的 rcartocolor 包，colorblindr 模拟色盲环境下的配色方案。

yarrr 包主要是为书籍《YaRrr! The Pirate's Guide to R》https://github.com/ndphillips/ThePiratesGuideToR 提供配套资源，兼顾收集了一组调色板。

注意

RColorBrewer 调色板数量必须至少 3 个，这是上游 colorbrewer 的 问题，具体体现在调用 `RColorBrewer::brewer.pal(n = 2, name = "Set2")` 时会有警告。plotly 调用

```
[1] "#66C2A5" "#FC8D62" "#8DA0CB"
Warning message:
In RColorBrewer::brewer.pal(n = 2, name = "Set2") :
  minimal value for n is 3, returning requested palette with 3 different levels
```

```
par(mar = c(1, 2, 1, 0), mfrow = c(3, 2))
set.seed(1234)
x <- sample(seq(8), 8, replace = FALSE)
barplot(x, col = palette(), border = "white")
barplot(x, col = heat.colors(8), border = "white")
barplot(x, col = gray.colors(8), border = "white")
barplot(x, col = "lightblue", border = "white")
barplot(x, col = colorspace::sequential_hcl(8), border = "white")
barplot(x, col = colorspace::diverge_hcl(8,
  h = c(130, 43),
  c = 100, l = c(70, 90)
), border = "white")
```

© 基础图书

图 11.25: colorspace 调色板

© 黄湘云

图 11.26: 源起

© 黄湘云

与图 11.90 对比，图11.27 的层次更加丰富，识别性更高

```
expand.grid(months = month.abb, years = 1949:1960) |>
  transform(num = as.vector(AirPassengers)) |>
  ggplot(aes(x = years, y = months, fill = num)) +
  scale_fill_distiller(palette = "Spectral") +
  geom_tile(color = "white", size = 0.4) +
  scale_x_continuous(
    expand = c(0.01, 0.01),
    breaks = seq(1949, 1960, by = 1),
    labels = 1949:1960
  ) +
  theme_minimal(
    base_size = 10.54,
    base_family = "Noto Serif SC"
  ) +
  labs(x = "年", y = "月", fill = "人数")
```

图 11.27: Spectral 调色板

再举例子，图 11.28 是正负例对比，其中好在哪里呢？这张图要表达美国黄石国家公园的老忠实泉间歇喷发的时间规律，那么好的标准就是层次分明，以突出不同颜色之间的时间差异。这个差异，还要看起来不

©黄湘云

那么费眼睛，一目了然最好。

```
erupt <- ggplot(faithfuld, aes(waiting, eruptions, fill = density)) +
  geom_raster() +
  scale_x_continuous(NULL, expand = c(0, 0)) +
  scale_y_continuous(NULL, expand = c(0, 0)) +
  theme(legend.position = "none")
p1 <- erupt + scale_fill_gradientn(colours = gray.colors(7))
p2 <- erupt + scale_fill_distiller(palette = "Spectral")
p3 <- erupt + scale_fill_gradientn(colours = terrain.colors(7))
p4 <- erupt + scale_fill_continuous(type = 'viridis')
(p1 + p2) / (p3 + p4)
```

图 11.28: 美国黄石国家公园的老忠实泉

RColorBrewer 包提供了有序 (Sequential) 、定性 (Qualitative) 和发散 (Diverging) 三类调色板，一般来讲，分别适用于连续或有序分类变量、无序分类变量、两类分层对比变量的绘图。再加上强大的 ggplot2 包内置的对颜色处理的函数，如 `scale_alpha_*` 、`scale_colour_*` 和 `scale_fill_*` 等，详见：

```
ls("package:ggplot2", pattern = "scale_col(ou|o)r_")
```

```
##  [1] "scale_color_binned"      "scale_color_brewer"
##  [3] "scale_color_continuous"  "scale_color_date"
##  [5] "scale_color_datetime"    "scale_color_discrete"
##  [7] "scale_color_distiller"   "scale_color_fermenter"
##  [9] "scale_color_gradient"    "scale_color_gradient2"
```

© 黄湘云


```
## [11] "scale_color_gradientn"   "scale_color_grey"
## [13] "scale_color_hue"         "scale_color_identity"
## [15] "scale_color_manual"      "scale_color_ordinal"
## [17] "scale_color_steps"       "scale_color_steps2"
## [19] "scale_color_stepsn"      "scale_color_viridis_b"
## [21] "scale_color_viridis_c"   "scale_color_viridis_d"
## [23] "scale_colour_binned"     "scale_colour_brewer"
## [25] "scale_colour_continuous" "scale_colour_date"
## [27] "scale_colour_datetime"   "scale_colour_discrete"
## [29] "scale_colour_distiller"  "scale_colour_fermenter"
## [31] "scale_colour_gradient"   "scale_colour_gradient2"
## [33] "scale_colour_gradientn"  "scale_colour_grey"
## [35] "scale_colour_hue"        "scale_colour_identity"
## [37] "scale_colour_manual"     "scale_colour_ordinal"
## [39] "scale_colour_steps"      "scale_colour_steps2"
## [41] "scale_colour_stepsn"     "scale_colour_viridis_b"
## [43] "scale_colour_viridis_c"  "scale_colour_viridis_d"
```

```
ls("package:ggplot2", pattern = "scale_fill_")
```

```
##  [1] "scale_fill_binned"     "scale_fill_brewer"     "scale_fill_continuous"
##  [4] "scale_fill_date"       "scale_fill_datetime"   "scale_fill_discrete"
##  [7] "scale_fill_distiller"  "scale_fill_fermenter"  "scale_fill_gradient"
## [10] "scale_fill_gradient2"  "scale_fill_gradientn"  "scale_fill_grey"
## [13] "scale_fill_hue"        "scale_fill_identity"   "scale_fill_manual"
## [16] "scale_fill_ordinal"    "scale_fill_steps"      "scale_fill_steps2"
## [19] "scale_fill_stepsn"     "scale_fill_viridis_b"  "scale_fill_viridis_c"
## [22] "scale_fill_viridis_d"
```

colourlovers 包借助 XML, jsonlite 和 httr 包可以在线获取网站 COLOURlovers 的调色板

```
library(colourlovers)
palette1 <- clpalette('113451')
palette2 <- clpalette('92095')
palette3 <- clpalette('629637')
palette4 <- clpalette('694737')
```

使用调色板

```
layout(matrix(1:4, nrow = 2))
par(mar = c(2, 2, 2, 2))

barplot(VADeaths, col = swatch(palette1)[[1]], border = NA)
barplot(VADeaths, col = swatch(palette2)[[1]], border = NA)
barplot(VADeaths, col = swatch(palette3)[[1]], border = NA)
barplot(VADeaths, col = swatch(palette4)[[1]], border = NA)
```

调色板的描述信息

```
palette1
```

获取调色板中的颜色向量

```
swatch(palette1)[[1]]
```

11.3.2 颜色模式

不同的颜色模式，从 RGB 到 HCL 的基本操作 https://stat545.com/block018_colors.html

```
# https://github.com/hadley/ggplot2-book
hcl <- expand.grid(x = seq(-1, 1, length = 100), y = seq(-1, 1, length = 100)) |>
  subset(subset = x^2 + y^2 < 1) |>
  transform(
    r = sqrt(x^2 + y^2)
  ) |>
  transform(
    h = 180 / pi * atan2(y, x),
    c = 100 * r,
    l = 65
  ) |>
  transform(
    colour = hcl(h, c, l)
  )

# sin(h) = y / (c / 100)
# y = sin(h) * c / 100

cols <- scales::hue_pal()(5)
selected <- colorspace::RGB(t(col2rgb(cols)) / 255) %>%
  as("polarLUV") %>%
  colorspace::coords() %>%
  as.data.frame() %>%
  transform(
    x = cos(H / 180 * pi) * C / 100,
    y = sin(H / 180 * pi) * C / 100,
    colour = cols
  )

ggplot(hcl, aes(x, y)) +
  geom_raster(aes(fill = colour)) +
  scale_fill_identity() +
  scale_colour_identity() +
  coord_equal() +
  scale_x_continuous("", breaks = NULL) +
  scale_y_continuous("", breaks = NULL) +
```

© 王建硕


```
  geom_point(data = selected, size = 10, color = "white") +
  geom_point(data = selected, size = 5, aes(colour = colour))
```

图 11.29: HCL 调色

R 内置了 502 种不同颜色的名称，下面随机地选取 20 种颜色

```
sample(colors(TRUE), 20)
```

```
##  [1] "royalblue4"      "plum1"           "papayawhip"      "darkslategray"
##  [5] "darkturquoise"   "gray79"          "darkred"         "maroon4"
##  [9] "darkolivegreen4" "springgreen2"    "orchid4"         "lemonchiffon2"
## [13] "paleturquoise4"  "gray49"          "cyan"            "antiquewhite1"
## [17] "yellow2"         "gray13"          "cadetblue2"      "gray77"
```

R 包 grDevices 提供 hcl 调色板[8] 调制两个色板

```
# Colors from https://github.com/johannesbjork/LaCroixColoR
color_pal <- c("#FF3200", "#E9A17C", "#E9E4A6", "#1BB6AF", "#0076BB", "#172869")
n <- 16
more_colors <- (grDevices::colorRampPalette(color_pal))(n)
scales::show_col(colours = more_colors)
```

```
# colors in colortools from http://www.gastonsanchez.com/
fish_pal <- c(
  "#69D2E7", "#6993E7", "#7E69E7", "#BD69E7",
```

[8] https://developer.r-project.org/Blog/public/2019/04/01/hcl-based-color-palettes-in-grdevices/index.html

© 黄湘云

#FF3200	#F75729	#F07B52	#E9A17C
#E9B78A	#E9CD98	#E9E4A6	#A4D4A9
#5FC5AC	#1BB6AF	#12A0B3	#098BB6
#0076BB	#075C9F	#0F4284	#172869

图 11.30: 桃色至梨色的渐变

```
  "#E769D2", "#E76993", "#E77E69", "#E7BD69",
  "#D2E769", "#93E769", "#69E77E", "#69E7BD"
)
more_colors <- (grDevices::colorRampPalette(fish_pal))(n)
scales::show_col(colours = more_colors)
```

```
rgb(red = 86, green = 180, blue = 233, maxColorValue = 255) # "#56B4E9"
```

```
## [1] "#56B4E9"
```

```
rgb(red = 0, green = 158, blue = 115, maxColorValue = 255) # "#009E73"
```

```
## [1] "#009E73"
```

```
rgb(red = 240, green = 228, blue = 66, maxColorValue = 255) # "#F0E442"
```

```
## [1] "#F0E442"
```

```
rgb(red = 0, green = 114, blue = 178, maxColorValue = 255) # "#0072B2"
```

```
## [1] "#0072B2"
```

举例子，直方图配色与不配色

```
# library(pander)
# evalsOptions('graph.unify', TRUE)
# panderOptions('graph.colors') 获取调色板
# https://www.fontke.com/tool/rgbschemes/ 在线配色
cols <- c(
```

© 黄湘云

#69D2E7	#69A3E7	#727FE7	#8A69E7
#B869E7	#D969D9	#E769B8	#E76B8D
#E77B6E	#E7A369	#E0CB69	#CDE769
#9FE769	#7CE774	#69E78E	#69E7BD

图 11.31: Hue-Saturation-Value (HSV) 颜色模型

```
  "#56B4E9", "#009E73", "#F0E442", "#0072B2",
  "#D55E00", "#CC79A7", "#999999", "#E69F00"
)
hist(mtcars$hp, col = "#56B4E9", border = "white", grid = grid())
```

```
ggplot(mtcars) +
  geom_histogram(aes(x = hp, fill = as.factor(..count..)),
    color = "white", bins = 6
  ) +
  scale_fill_manual(values = rep("#56B4E9", 10)) +
  ggtitle("Histogram with ggplot2") +
  theme_minimal() +
  theme(legend.position = "none")
```

11.3.2.1 RGB

红 (red)、绿 (green)、蓝 (blue) 是三原色

```
rgb(red, green, blue, alpha, names = NULL, maxColorValue = 1)
```

函数参数说明:

- `red`, `blue`, `green`, `alpha` 取值范围 $[0, M]$，M 是 *maxColorValue*
- `names` 字符向量，给这组颜色值取名
- `maxColorValue` 红，绿，蓝三色范围的最大值

© 黄湘云

图 11.32: 直方图

图 11.33: 直方图

© 王建硕

The colour specification refers to the standard sRGB colorspace (IEC standard 61966).

rgb 产生一种颜色，如 `rgb(255, 0, 0, maxColorValue = 255)` 的颜色是 `"#FF0000"`，这是一串 16 进制数，每两个一组，那么一组有 $16^2 = 256$ 种组合，整个一串有 $256^3 = 16777216$ 种组合，这就是 RGB 表达的所有颜色。

11.3.2.2 HSL

色相饱和度亮度 hue–saturation–luminance (HSL)

11.3.2.3 HSV

Create a vector of colors from vectors specifying hue, saturation and value. 色相饱和度值

```
hsv(h = 1, s = 1, v = 1, alpha)
```

This function creates a vector of colors corresponding to the given values in HSV space. rgb and rgb2hsv for RGB to HSV conversion;

hsv 函数通过设置色调、饱和度和亮度获得颜色，三个值都是 0-1 的相对量

RGB HSV HSL 都是不连续的颜色空间，缺点

11.3.2.4 HCL

基于感知的颜色空间替代 RGB 颜色空间

通过指定色相 (hue)，色度 (chroma) 和亮度 (luminance/lightness)，创建一组（种）颜色

```
hcl(h = 0, c = 35, l = 85, alpha, fixup = TRUE)
```

函数参数说明：

- **h** 颜色的色调，取值范围为 [0,360]，0、120、240 分别对应红色、绿色、蓝色
- **c** 颜色的色度，其上界取决于色调和亮度
- **l** 颜色的亮度，取值范围 [0,100]，给定色调和色度，只有一部分子集可用
- **alpha** 透明度，取值范围 [0,1]，0 和 1 分别表示透明和不透明

This function corresponds to polar coordinates in the CIE-LUV color space

选色为什么这么难

色相与阴影相比是无关紧要的，色相对于标记和分类很有用，但表示（精细的）空间数据或形状的效果较差。颜色是改善图形的好工具，但糟糕的配色方案 (color schemes) 可能会导致比灰度调色板更差的效果。[Stauffer et al., 2009]

黑、白、灰，看似有三种颜色，其实只有一种颜色，黑和白只是灰色的两极，那么如何设置灰色梯度，使得人眼比较好区分它们呢？这样获得的调色板适用于什么样的绘图环境呢？

© 草稿

11.3.2.5 CMYK

印刷三原色：青 (cyan)、品红 (magenta)、黄 (yellow)

- 颜色模式转化

`col2rgb()` 、`rgb2hsv()` 和 `rgb()` 函数 `hex2RGB()` 函数 colorspace `col2hcl()` 函数 scales `col2HSV()` colortools `col2hex()`

```
col2rgb("lightblue") # color to  RGB
```

```
##       [,1]
## red    173
## green  216
## blue   230
```

```
scales::col2hcl("lightblue") # color to HCL
```

```
## [1] "#ADD8E6"
```

```
# palr::col2hex("lightblue") # color to HEX
# colortools::col2HSV("lightblue") # color to HSV

rgb(173, 216, 230, maxColorValue = 255) # RGB to HEX
```

```
## [1] "#ADD8E6"
```

```
colorspace::hex2RGB("#ADD8E6") # HEX to RGB
```

```
##               R         G         B
## [1,] 0.6784314 0.8470588 0.9019608
```

```
rgb(.678, .847, .902, maxColorValue = 1) # RGB to HEX
```

```
## [1] "#ADD8E6"
```

```
rgb2hsv(173, 216, 230, maxColorValue = 255) # RGB to HSV
```

```
##        [,1]
## h 0.5409357
## s 0.2478261
## v 0.9019608
```

11.3.3 LaTeX 配色

LaTeX 宏包 xcolor 中定义颜色的常用方式有两种，其一，`\textcolor{green!40!yellow}` 表示 40% 的绿色和 60% 的黄色混合色彩，其二，`\textcolor[HTML]{34A853}` HEX 表示的色彩直接在 LaTeX 文档中使用的方式，类似地 `\textcolor[RGB]{52,168,83}` 也表示 Google 图标中的绿色。

```
\documentclass[tikz,border=10pt]{standalone}
\begin{document}
\begin{tikzpicture}
\draw (0,0) rectangle (2,1) node [midway] {\textcolor[RGB]{52,168,83}{Hello} \textcolor[HTML]{34A853}{\TeX
```

© 王建硕


```
\end{tikzpicture}
\end{document}
```

对应于 R 中的调用方式为：

```
rgb(52, 168, 83, maxColorValue = 255)
```

```
## [1] "#34A853"
```

11.3.4 ggplot2 配色

```
boxplot(weight ~ group,
  data = PlantGrowth, col = "lightgray",
  notch = FALSE, varwidth = TRUE
)
# 类似 boxplot
ggplot(data = PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot(notch = FALSE, varwidth = TRUE, fill = "lightgray")

# 默认调色板
ggplot(data = PlantGrowth, aes(x = group, y = weight, fill = group)) +
  geom_boxplot(notch = FALSE, varwidth = TRUE)

# Google 调色板
ggplot(data = PlantGrowth, aes(x = group, y = weight, fill = group)) +
  geom_boxplot(notch = FALSE, varwidth = TRUE) +
  scale_fill_manual(values = c("#4285f4", "#34A853", "#FBBC05", "#EA4335"))
```

11.4 图库

11.4.1 饼图

我对饼图是又爱又恨，爱的是它表示百分比的时候，往往让读者联想到蛋糕，份额这类根深蒂固的情景，从而让数字通俗易懂、深入人心，是一种很好的表达方式，恨的也是这一点，我用柱状图表达不香吗？人眼对角度的区分度远不如柱状图呢，特别是当两个类所占的份额比较接近的时候，所以很多时候，除了用饼图表达份额，还会在旁边标上百分比，从数据可视化的角度来说，如图 11.35 所示，这是信息冗余！

```
BOD %>% transform(., ratio = demand / sum(demand)) %>%
  ggplot(., aes(x = "", y = demand, fill = reorder(Time, demand))) +
  geom_bar(stat = "identity", show.legend = FALSE, color = "white") +
  coord_polar(theta = "y") +
  geom_text(aes(x = 1.6, label = paste0(round(ratio, digits = 4) * 100, "%")),
    position = position_stack(vjust = 0.5), color = "black"
  ) +
  geom_text(aes(x = 1.2, label = Time),
```

© 黄湘云

(a) 简单箱线图

(b) ggplot2 绘制的箱线图

(c) ggplot2 调用默认调色板

(d) ggplot2 调用 Google 调色板

图 11.34: 几种不同的箱线图

© 黄湘云


```
    position = position_stack(vjust = 0.5), color = "black"
  ) +
  theme_void(base_size = 14)
```

图 11.35: 饼图

plot_ly(type = "pie", ...) 和添加图层 add_pie() 的效果是一样的

```
dat = aggregate(formula = carat ~ cut, data = diamonds, FUN = length)
plotly::plot_ly() %>%
  plotly::add_pie(
    data = dat, labels = ~cut, values = ~carat,
    name = "简单饼图1", domain = list(row = 0, column = 0)
  ) %>%
  plotly::add_pie(
    data = dat, labels = ~cut, values = ~carat, hole = 0.6,
    textposition = "inside", textinfo = "label+percent",
    name = "简单饼图2", domain = list(row = 0, column = 1)
  ) %>%
  plotly::layout(
    title = "多图布局", showlegend = F,
    grid = list(rows = 1, columns = 2),
```

© 黄湘云


```
    xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
    yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE)
  ) %>%
  plotly::config(displayModeBar = FALSE)
```

设置参数 hole 可以绘制环形饼图，比如 hole = 0.6

11.4.2 地图

USArrests 数据集描述了 1973 年美国 50 个州每 10 万居民中因袭击、抢劫和强奸而逮捕的人，以及城市人口占比。这里的地图是指按照行政区划为边界的示意图，比如图 11.36

```
library(maps)
crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)
# 等价于 crimes %>% tidyr::pivot_longer(Murder:Rape)
vars <- lapply(names(crimes)[-1], function(j) {
  data.frame(state = crimes$state, variable = j, value = crimes[[j]])
})
crimes_long <- do.call("rbind", vars)
states_map <- map_data("state")
ggplot(crimes, aes(map_id = state)) +
  geom_map(aes(fill = Murder), map = states_map) +
  expand_limits(x = states_map$long, y = states_map$lat) +
  scale_fill_binned(type = "viridis") +
  coord_map() +
  theme_minimal()
```

图 11.36: 1975 年美国各州犯罪事件

先来看看中国及其周边，见图11.37，这个地图的缺陷就是中国南海及九段线没有标记，台湾和中国大陆不是一种颜色标记，这里的地图数据来自 R 包 **maps** 和 **mapdata**，像这样的地图就不宜在国内正式刊物

© 黄湘云

上出现。

```
library(maps)
library(mapdata)
east_asia <- map_data("worldHires",
  region = c(
    "Japan", "Taiwan", "China",
    "North Korea", "South Korea"
  )
)
ggplot(east_asia, aes(x = long, y = lat, group = group, fill = region)) +
  geom_polygon(colour = "black") +
  scale_fill_brewer(palette = "Set2") +
  theme_minimal()
```

图 11.37: 中国及其周边

绘制真正的地图需要考虑投影坐标系，观察角度、分辨率、政策法规等一系列因素，它是一种复杂的图形，如图 11.38 所示。

```
worldmap <- map_data("world")

# 默认 mercator 投影下的默认视角 c(90, 0, mean(range(x)))
ggplot(worldmap, aes(long, lat, group = group)) +
  geom_polygon(aes(fill = region), show.legend = FALSE) +
  coord_map(
    xlim = c(-120, 40), ylim = c(30, 90)
  )

# 换观察角度
ggplot(worldmap, aes(long, lat, group = group)) +
```

© 基湘 1 3


```
  geom_polygon(aes(fill = region), show.legend = FALSE) +
  coord_map(
    xlim = c(-120, 40), ylim = c(30, 90),
    orientation = c(90, 0, 0)
  )

# 换投影坐标系
ggplot(worldmap, aes(long, lat, group = group)) +
  geom_polygon(aes(fill = region), show.legend = FALSE) +
  coord_map("ortho",
    xlim = c(-120, 40), ylim = c(30, 90)
  )

# 二者皆换
ggplot(worldmap, aes(long, lat, group = group)) +
  geom_polygon(aes(fill = region), show.legend = FALSE) +
  coord_map("ortho",
    xlim = c(-120, 40), ylim = c(30, 90),
    orientation = c(90, 0, 0)
  )
```

Google 地图

```
library(RgoogleMaps)
# 一组坐标的中心位置
lat <- c(40.702147, 40.718217, 40.711614)
lon <- c(-74.012318, -74.015794, -73.998284)
center <- c(mean(lat), mean(lon))
zoom <- min(MaxZoom(range(lat), range(lon)))
# 矩形对角线的两个顶点
bb <- qbbox(lat, lon)
# 获取地图数据
myMap <- GetMap(center, size = c(640, 640), zoom = zoom, type = "osm")
# 在地图上添加红、蓝、绿三个点
PlotOnStaticMap(myMap,
  lat = lat, lon = lon, pch = 20, cex = 10,
  col = c("red", "blue", "green")
)
```

11.4.3 热图

Zuguang Gu 开发的 ComplexHeatmap 包实现复杂数据的可视化，用以发现关联数据集之间的模式。特别地，比如基因数据、生存数据等，更多应用见开发者的书籍 ComplexHeatmap 完全手册 。R 包发布在 Bioconductor 上 https://www.bioconductor.org/packages/ComplexHeatmap。使用之前我要确保已经安装 **BiocManager** 包，这个包负责管理 Bioconductor 上所有的包，需要先安装它，然后安装 **ComplexHeatmap**

© 黄湘云

(a) 墨卡托投影

(b) 北极观察

(c) 正交投影

(d) 正交投影北极观察

图 11.38: 画地图的正确姿势

© 黄湘云

图 11.39: Google 地图示例

© 基础统计

包 [Gu et al., 2016]。

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("ComplexHeatmap")
```

11.4.4 散点图

下面以 diamonds 数据集为例展示 ggplot2 的绘图过程，首先加载 diamonds 数据集，查看数据集的内容

```
data(diamonds)
str(diamonds)
```

```
## tibble [53,940 x 10] (S3: tbl_df/tbl/data.frame)
##  $ carat  : num [1:53940] 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
##  $ depth  : num [1:53940] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num [1:53940] 55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int [1:53940] 326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num [1:53940] 3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num [1:53940] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num [1:53940] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

数值型变量 carat 作为 x 轴

```
ggplot(diamonds, aes(x = carat))
ggplot(diamonds, aes(x = carat, y = price))
ggplot(diamonds, aes(x = carat, color = cut))
ggplot(diamonds, aes(x = carat), color = "steelblue")
```

图 11.40 的基础上添加数据图层

```
sub_diamonds <- diamonds[sample(1:nrow(diamonds), 1000), ]
ggplot(sub_diamonds, aes(x = carat, y = price)) +
  geom_point()
```

给散点图11.41上色

```
ggplot(sub_diamonds, aes(x = carat, y = price)) +
  geom_point(color = "steelblue")
```

```
ggplot(sub_diamonds, aes(x = carat, y = price)) +
  geom_point(color = "steelblue") +
  scale_y_continuous(
    labels = scales::unit_format(unit = "k", scale = 1e-3),
    breaks = seq(0, 20000, 4000)
  )
```

让另一变量 cut 作为颜色分类指标

© 黄湘云

(a) 指定 x 轴

(b) 数值变量 price 作为纵轴

(c) 有序分类变量 cut 指定颜色

(d) 指定统一颜色

图 11.40: 绘图过程

© 黄湘云

图 11.41: 添加数据图层

图 11.42: 散点图配色

© 黄湘云

图 11.43: 格式化坐标轴刻度标签

```
ggplot(sub_diamonds, aes(x = carat, y = price, color = cut)) +
  geom_point()
```

当然还有一种类似的表示就是分组，默认情况下，ggplot2 将所有观测点视为一组，以分类变量 cut 来分组

```
ggplot(sub_diamonds, aes(x = carat, y = price, group = cut)) +
  geom_point()
```

在图11.45 上没有体现出来分组的意思，下面以 cut 分组线性回归为例

```
ggplot(sub_diamonds, aes(x = carat, y = price)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
ggplot(sub_diamonds, aes(x = carat, y = price, group = cut)) +
  geom_point() +
  geom_smooth(method = "lm")
```

我们当然可以选择更加合适的拟合方式，如局部多项式平滑 `loess` 但是该方法不太适用观测值比较多的情况，因为它会占用比较多的内存，建议使用广义可加模型作平滑拟合

```
ggplot(sub_diamonds, aes(x = carat, y = price, group = cut)) +
  geom_point() +
  geom_smooth(method = "loess")
```

© 黄湘云

图 11.44: 分类散点图

图 11.45: 分组

© 黄湘云

图 11.46: 分组线性回归

图 11.47: 分组线性回归

© 黄湘云

图 11.48: 局部多项式平滑

```
ggplot(sub_diamonds, aes(x = carat, y = price, group = cut)) +
  geom_point() +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"))
```

ggfortify 包支持更多的统计分析结果的可视化。

为了更好地区分开组别，我们在图11.49的基础上分面或者配色

```
ggplot(sub_diamonds, aes(x = carat, y = price, group = cut)) +
  geom_point() +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs")) +
  facet_grid(~cut)
```

```
ggplot(sub_diamonds, aes(x = carat, y = price, group = cut, color = cut)) +
  geom_point() +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"))
```

在分类散点图的另一种表示方法就是分面图，以 cut 变量作为分面的依据

```
ggplot(sub_diamonds, aes(x = carat, y = price)) +
  geom_point() +
  facet_grid(~cut)
```

给图 11.52 上色

© 黄湘云

图 11.49: 数据分组应用广义可加平滑

图 11.50: 分组分面

© 黄湘云

图 11.51: 分组配色

图 11.52: 分面散点图

© 黄湘云


```
ggplot(sub_diamonds, aes(x = carat, y = price)) +
  geom_point(color = "steelblue") +
  facet_grid(~cut)
```

图 11.53: 给分面散点图上色

在图11.53的基础上，给不同的类上不同的颜色

```
ggplot(sub_diamonds, aes(x = carat, y = price, color = cut)) +
  geom_point() +
  facet_grid(~cut)
```

去掉图例，此时图例属于冗余信息了

```
ggplot(sub_diamonds, aes(x = carat, y = price, color = cut)) +
  geom_point(show.legend = FALSE) +
  facet_grid(~cut)
```

四块土地，所施肥料不同，肥力大小顺序 4 < 2 < 3 < 1 小麦产量随肥力的变化

```
data(Wheat2, package = "nlme") # Wheat Yield Trials
library(colorspace)
ggplot(Wheat2, aes(longitude, latitude)) +
  geom_point(aes(size = yield, colour = Block)) +
  scale_color_discrete_sequential(palette = "Viridis") +
  scale_x_continuous(breaks = seq(0, 30, 5)) +
  scale_y_continuous(breaks = seq(0, 50, 10))
```

© 黄湘云

图 11.54: 给不同的类上不同的颜色

图 11.55: 去掉图例

©黄湘云

图 11.56: 多个图例

```
ggplot(mtcars, aes(x = hp, y = mpg, color = factor(am))) +
  geom_point()
```

图层、分组、分面和散点图介绍完了，接下来就是其它统计图形，如箱线图，小提琴图和条形图

```
dat <- as.data.frame(cbind(rep(1948 + seq(12), each = 12), rep(seq(12), 12), AirPassengers))
colnames(dat) <- c("year", "month", "passengers")

ggplot(data = dat, aes(x = as.factor(year), y = as.factor(month))) +
  stat_sum(aes(size = passengers), colour = "lightblue") +
  scale_size(range = c(1, 10), breaks = seq(100, 650, 50)) +
  labs(x = "Year", y = "Month", colour = "Passengers") +
  theme_minimal()
```

11.4.5 条形图

条形图特别适合分类变量的展示，我们这里展示钻石切割质量 cut 不同等级的数量，当然我们可以直接展示各类的数目，在图层 geom_bar 中指定 stat="identity"

```
# 需要映射数据框的两个变量，相当于自己先计算了每类的数量
with(diamonds, table(cut))
```

```
## cut
```

©黄湘云

图 11.57: 分类散点图

图 11.58: 1948 年至 1960 年航班乘客人数变化

© 基础统计

```
##      Fair      Good Very Good   Premium     Ideal
##      1610      4906     12082     13791     21551
```

```
cut_df <- as.data.frame(table(diamonds$cut))
ggplot(cut_df, aes(x = Var1, y = Freq)) + geom_bar(stat = "identity")
```

```
ggplot(diamonds, aes(x = cut)) + geom_bar()
```

还有另外三种表示方法

```
ggplot(diamonds, aes(x = cut)) + geom_bar(stat = "count")
```

© 黄湘云

图 11.59: 频数条形图

© 基础编


```
ggplot(diamonds, aes(x = cut, y = ..count..)) + geom_bar()
```

```
ggplot(diamonds, aes(x = cut, y = stat(count))) + geom_bar()
```

© 黄湘云

我们还可以在图 11.59 的基础上再添加一个分类变量钻石的纯净度 clarity，形成堆积条形图

```
ggplot(diamonds, aes(x = cut, fill = clarity)) + geom_bar()
```

再添加一个分类变量钻石颜色 color 比较好的做法是分面

```
ggplot(diamonds, aes(x = color, fill = clarity)) +
  geom_bar() +
  facet_grid(~cut)
```

实际上，绘制图11.61包含了对分类变量的分组计数过程，如下

```
with(diamonds, table(cut, color))
```

```
##            color
## cut            D    E    F    G    H    I    J
##   Fair       163  224  312  314  303  175  119
##   Good       662  933  909  871  702  522  307
##   Very Good 1513 2400 2164 2299 1824 1204  678
##   Premium   1603 2337 2331 2924 2360 1428  808
##   Ideal     2834 3903 3826 4884 3115 2093  896
```

还有一种堆积的方法是按比例，而不是按数量，如图11.62

```
ggplot(diamonds, aes(x = color, fill = clarity)) +
  geom_bar(position = "fill") +
  facet_grid(~cut)
```

接下来就是复合条形图

© 黄湘云

图 11.60: 堆积条形图

图 11.61: 分面堆积条形图

©黄湘云

图 11.62: 比例堆积条形图

```
ggplot(diamonds, aes(x = color, fill = clarity)) +
  geom_bar(position = "dodge")
```

再添加一个分类变量，就是需要分面大法了，图 11.63 展示了三个分类变量，其实我们还可以再添加一个分类变量用作分面的列依据

```
ggplot(diamonds, aes(x = color, fill = clarity)) +
  geom_bar(position = "dodge") +
  facet_grid(rows = vars(cut))
```

图 11.64 展示的数据如下

```
with(diamonds, table(color, clarity, cut))
```

```
## , , cut = Fair
##
##       clarity
## color   I1  SI2  SI1  VS2  VS1 VVS2 VVS1   IF
##     D    4   56   58   25    5    9    3    3
##     E    9   78   65   42   14   13    3    0
##     F   35   89   83   53   33   10    5    4
##     G   53   80   69   45   45   17    3    2
##     H   52   91   75   41   32   11    1    0
##     I   34   45   30   32   25    8    1    0
```

©黄湘云

图 11.63: 复合条形图

图 11.64: 分面复合条形图

©清华大学出版社

```
##     J   23   27   28   23   16    1    1    0
##
## , , cut = Good
##
##      clarity
## color   I1  SI2  SI1  VS2  VS1 VVS2 VVS1   IF
##     D    8  223  237  104   43   25   13    9
##     E   23  202  355  160   89   52   43    9
##     F   19  201  273  184  132   50   35   15
##     G   19  163  207  192  152   75   41   22
##     H   14  158  235  138   77   45   31    4
##     I    9   81  165  110  103   26   22    6
##     J    4   53   88   90   52   13    1    6
##
## , , cut = Very Good
##
##      clarity
## color   I1  SI2  SI1  VS2  VS1 VVS2 VVS1   IF
##     D    5  314  494  309  175  141   52   23
##     E   22  445  626  503  293  298  170   43
##     F   13  343  559  466  293  249  174   67
##     G   16  327  474  479  432  302  190   79
##     H   12  343  547  376  257  145  115   29
##     I    8  200  358  274  205   71   69   19
##     J    8  128  182  184  120   29   19    8
##
## , , cut = Premium
##
##      clarity
## color   I1  SI2  SI1  VS2  VS1 VVS2 VVS1   IF
##     D   12  421  556  339  131   94   40   10
##     E   30  519  614  629  292  121  105   27
##     F   34  523  608  619  290  146   80   31
##     G   46  492  566  721  566  275  171   87
##     H   46  521  655  532  336  118  112   40
##     I   24  312  367  315  221   82   84   23
##     J   13  161  209  202  153   34   24   12
##
## , , cut = Ideal
##
##      clarity
## color   I1  SI2  SI1  VS2  VS1 VVS2 VVS1   IF
##     D   13  356  738  920  351  284  144   28
##     E   18  469  766 1136  593  507  335   79
##     F   42  453  608  879  616  520  440  268
```

```
##     G   16  486  660  910  953  774  594  491
##     H   38  450  763  556  467  289  326  226
##     I   17  274  504  438  408  178  179   95
##     J    2  110  243  232  201   54   29   25
```

```
# 漫谈条形图 https://cosx.org/2017/10/discussion-about-bar-graph
set.seed(2020)
dat <- data.frame(
  age = rep(1:30, 2),
  gender = rep(c("man", "woman"), each = 30),
  num = sample(x = 1:100, size = 60, replace = T)
)
# 重叠
p1 <- ggplot(data = dat, aes(x = age, y = num, fill = gender)) +
  geom_col(position = "identity", alpha = 0.5)
# 堆积
p2 <- ggplot(data = dat, aes(x = age, y = num, fill = gender)) +
  geom_col(position = "stack")
# 双柱
p3 <- ggplot(data = dat, aes(x = age, y = num, fill = gender)) +
  geom_col(position = "dodge")
# 百分比
p4 <- ggplot(data = dat, aes(x = age, y = num, fill = gender)) +
  geom_col(position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(y = "%")
(p1 + p2) / (p3 + p4)
```

以数据集 diamonds 为例，按照纯净度 clarity 和切工 cut 分组统计钻石的数量，再按切工分组统计不同纯净度的钻石数量占比，如表 11.2 所示

```
library(data.table)
diamonds <- as.data.table(diamonds)
dat <- diamonds[, .(cnt = .N), by = .(cut, clarity)] %>%
  .[, pct := cnt / sum(cnt), by = .(cut)] %>%
  .[, pct_pp := paste0(cnt, " (", scales::percent(pct, accuracy = 0.01), ")") ]
# 分组计数 with(diamonds, table(clarity, cut))
dcast(dat, formula = clarity ~ cut, value.var = "pct_pp") %>%
  knitr::kable(align = "crrrrr", caption = "数值和比例组合呈现")
```

分别以堆积条形图和百分比堆积条形图展示，添加注释到条形图上，见 11.66

```
p1 = ggplot(data = dat, aes(x = cut, y = cnt, fill = clarity)) +
  geom_col(position = "dodge") +
  geom_text(aes(label = cnt), position = position_dodge(1), vjust = -0.5) +
  geom_text(aes(label = scales::percent(pct, accuracy = 0.1)),
    position = position_dodge(1), vjust = 1, hjust = 0.5
  ) +
```

© 黄湘云

图 11.65: 条形图的四种常见形态

表 11.2: 数值和比例组合呈现

clarity	Fair	Good	Very Good	Premium	Ideal
I1	210 (13.04%)	96 (1.96%)	84 (0.70%)	205 (1.49%)	146 (0.68%)
SI2	466 (28.94%)	1081 (22.03%)	2100 (17.38%)	2949 (21.38%)	2598 (12.06%)
SI1	408 (25.34%)	1560 (31.80%)	3240 (26.82%)	3575 (25.92%)	4282 (19.87%)
VS2	261 (16.21%)	978 (19.93%)	2591 (21.45%)	3357 (24.34%)	5071 (23.53%)
VS1	170 (10.56%)	648 (13.21%)	1775 (14.69%)	1989 (14.42%)	3589 (16.65%)
VVS2	69 (4.29%)	286 (5.83%)	1235 (10.22%)	870 (6.31%)	2606 (12.09%)
VVS1	17 (1.06%)	186 (3.79%)	789 (6.53%)	616 (4.47%)	2047 (9.50%)
IF	9 (0.56%)	71 (1.45%)	268 (2.22%)	230 (1.67%)	1212 (5.62%)

© 清华大学出版社

```
  scale_fill_brewer(palette = "Spectral") +
  labs(fill = "clarity", y = "", x = "cut") +
  theme_minimal() +
  theme(legend.position = "top")

p2 = ggplot(data = dat, aes(y = cut, x = cnt, fill = clarity)) +
  geom_col(position = "fill") +
  geom_text(aes(label = cnt), position = position_fill(1), vjust = -0.5) +
  geom_text(aes(label = scales::percent(pct, accuracy = 0.1)),
    position = position_fill(1), vjust = 1, hjust = 0.5
  ) +
  scale_fill_brewer(palette = "Spectral") +
  scale_x_continuous(labels = scales::percent) +
  labs(fill = "clarity", y = "", x = "cut") +
  theme_minimal() +
  theme(legend.position = "top")

p1 / p2
```

借助 plotly 制作相应的动态百分比堆积条形图

```
ggplot(data = diamonds, aes(x = cut, fill = clarity)) +
  geom_bar(position = "dodge2") +
  scale_fill_brewer(palette = "Spectral")

# 百分比堆积条形图
plotly::plot_ly(dat,
  x = ~cut, color = ~clarity, y = ~pct,
  colors = "Spectral", type = "bar",
  text = ~ paste0(
    cnt, "颗 <br>",
    "占比: ", scales::percent(pct, accuracy = 0.1), "<br>"
  ),
  hoverinfo = "text"
) %>%
  plotly::layout(
    barmode = "stack",
    yaxis = list(tickformat = ".0%")
  ) %>%
  plotly::config(displayModeBar = FALSE)

# `type = "histogram"` 以 cut 和 clarity 分组计数
plotly::plot_ly(diamonds,
  x = ~cut, color = ~clarity,
  colors = "Spectral", type = "histogram"
) %>%
```

© 黄湘云

图 11.66: 添加注释到条形图

© 黄湘云


```
  plotly::config(displayModeBar = FALSE)

# 堆积图
plotly::plot_ly(diamonds,
  x = ~cut, color = ~clarity,
  colors = "Spectral", type = "histogram"
) %>%
  plotly::layout(
    barmode = "stack",
    yaxis = list(title = "cnt"),
    legend = list(title = list(text = "clarity"))
  ) %>%
  plotly::config(displayModeBar = FALSE)
```

11.4.6 直方图

直方图用来查看连续变量的分布

```
ggplot(diamonds, aes(price)) + geom_histogram(bins = 30)
```

图 11.67: 钻石价格的分布

堆积直方图

© 黄湘云


```
ggplot(diamonds, aes(x = price, fill = cut)) + geom_histogram(bins = 30)
```

图 11.68: 钻石价格随切割质量的分布

基础 R 包与 Ggplot2 包绘制的直方图的对比，Base R 绘图速度快，代码更加稳定，Ggplot2 代码简洁，更美观

```
par(mar = c(2.1, 2.1, 1.5, 0.5))
plot(c(50, 350), c(0, 10),
  type = "n", font.main = 1,
  xlab = "", ylab = "", frame.plot = FALSE, axes = FALSE,
  # xlab = "hp", ylab = "Frequency",
  main = paste("Histogram with Base R", paste(rep(" ", 60), collapse = ""))
)
axis(
  side = 1, at = seq(50, 350, 50), labels = seq(50, 350, 50),
  tick = FALSE, las = 1, padj = 0, mgp = c(3, 0.1, 0)
)
axis(
  side = 2, at = seq(0, 10, 2), labels = seq(0, 10, 2),
  # col = "white", 坐标轴的颜色
  # col.ticks 刻度线的颜色
  tick = FALSE, # 取消刻度线
  las = 1, # 水平方向
  hadj = 1, # 右侧对齐
```

©黄湘云

```
  mgp = c(3, 0.1, 0) # 纵轴边距线设置为 0.1
)
abline(h = seq(0, 10, 2), v = seq(50, 350, 50), col = "gray90", lty = "solid")
abline(h = seq(1, 9, 2), v = seq(75, 325, 50), col = "gray95", lty = "solid")
hist(mtcars$hp,
  col = "#56B4E9", border = "white",
  freq = TRUE, add = TRUE
  # labels = TRUE, axes = TRUE, ylim = c(0, 10.5),
  # xlab = "hp",main = "Histogram with Base R"
)
mtext("hp", 1, line = 1.0)
mtext("Frequency", 2, line = 1.0)

ggplot(mtcars) +
  geom_histogram(aes(x = hp), fill = "#56B4E9", color = "white", breaks = seq(50, 350, 50)) +
  scale_x_continuous(breaks = seq(50, 350, 50)) +
  scale_y_continuous(breaks = seq(0, 12, 2)) +
  labs(x = "hp", y = "Frequency", title = "Histogram with Ggplot2") +
  theme_minimal(base_size = 12)
```

(a) Base R 直方图

(b) Ggplot2 直方图

图 11.69: 直方图

11.4.7 箱线图

以 PlantGrowth 数据集为例展示箱线图，在两组不同实验条件下，植物生长的情况，纵坐标是干燥植物的量，横坐标表示不同的实验条件。这是非常典型的适合用箱线图来表达数据的场合，Y 轴对应数值型变量，X 轴对应分类变量，在 R 语言中，分类变量的类型是 factor

```
data("PlantGrowth")
str(PlantGrowth)
```

```
## 'data.frame':    30 obs. of  2 variables:
##  $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
```

© 黄湘云


```
##  $ group : Factor w/ 3 levels "ctrl","trt1",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
ggplot(data = PlantGrowth, aes(x = group, y = weight)) + geom_boxplot()
```

PlantGrowth 数据量比较小，此时比较适合采用抖动散点图，抖动是为了避免点之间相互重叠，为了增加不同类别之间的识别性，我们可以用不同的点的形状或者不同的颜色来表示类别

```
ggplot(data = PlantGrowth, aes(x = group, y = weight, shape = group)) + geom_jitter()
```

© 黄湘云

```
ggplot(data = PlantGrowth, aes(x = group, y = weight, color = group)) + geom_jitter()
```

© 黄湘云


```
boxplot(weight ~ group,
  data = PlantGrowth,
  ylab = "Dried weight of plants", col = "lightgray",
  notch = FALSE, varwidth = TRUE
)
```

以钻石切割质量 cut 为分面依据，以钻石颜色类别 color 为 x 轴，钻石价格为 y 轴，绘制箱线图11.70

```
ggplot(diamonds, aes(x = color, y = price, color = cut)) +
  geom_boxplot(show.legend = FALSE) +
  facet_grid(~cut)
```

我们当然还可以添加钻石的纯净度 clarity 作为分面依据，那么箱线图可以为图 11.71

```
ggplot(diamonds, aes(x = color, y = price, color = cut)) +
  geom_boxplot(show.legend = FALSE) +
  facet_grid(clarity ~ cut)
```

经过观察，我们发现水平分类过多，考虑用切割质量 cut 替换钻石颜色 color 绘图，但是由于分类过细，图信息展示不简练，反而不好，如图 11.72

```
ggplot(diamonds, aes(x = cut, y = price, color = cut)) +
  geom_boxplot(show.legend = FALSE) +
```

© 黄湘石

图 11.70: 箱线图

图 11.71: 复合分面箱线图

© 基础出版


```
  facet_grid(clarity ~ color)
ggplot(diamonds, aes(x = cut, y = price, color = color)) +
  geom_boxplot(show.legend = FALSE) +
  facet_grid(clarity ~ color)
```

11.4.8 函数图

蝴蝶图的参数方程如下

$$x = \sin t\left(\mathrm{e}^{\cos t} - 2\cos 4t + \sin^5\left(\frac{t}{12}\right)\right) \tag{11.1}$$

$$y = \cos t\left(\mathrm{e}^{\cos t} - 2\cos 4t + \sin^5\left(\frac{t}{12}\right)\right), t \in [-\pi, \pi] \tag{11.2}$$

11.4.9 密度图

```
ggplot(mpg, aes(cty)) +
  geom_density(aes(fill = factor(cyl)), alpha = 0.8) +
  labs(
    title = "Density plot",
    subtitle = "City Mileage Grouped by Number of cylinders",
    caption = "Source: mpg",
    x = "City Mileage",
    fill = "# Cylinders"
  )
```

添加透明度，解决遮挡

```
ggplot(diamonds, aes(x = price, fill = cut)) + geom_density()
```

```
ggplot(diamonds, aes(x = price, fill = cut)) + geom_density(alpha = 0.5)
```

堆积密度图

```
ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_density(position = "stack")
```

条件密度估计

```
# You can use position="fill" to produce a conditional density estimate
ggplot(diamonds, aes(carat, stat(count), fill = cut)) +
  geom_density(position = "fill")
```

岭线图是密度图的一种变体，可以防止密度曲线重叠在一起

```
ggplot(diamonds) +
  ggridges::geom_density_ridges(aes(x = price, y = color, fill = color))
```

© 黄湘石

(a) 切割质量 cut 上色

(b) 钻石颜色配色

图 11.72: 箱线图配色

© 黄湘云

图 11.73: 按汽缸数分组的城市里程

图 11.74: 密度图

© 黄湘云

图 11.75: 添加透明度的密度图

图 11.76: 堆积密度图

© 黄湘云

图 11.77: 条件密度估计图

© 基础设施

二维的密度图又是一种延伸

```
ggplot(diamonds, aes(x = carat, y = price)) +
  geom_density_2d(aes(color = cut)) +
  facet_grid(~cut)
```

stat 函数，特别是 nlevel 参数，在密度曲线之间填充我们又可以得到热力图

```
ggplot(diamonds, aes(x = carat, y = price)) +
  stat_density_2d(aes(fill = stat(nlevel)), geom = "polygon") +
  facet_grid(. ~ cut)
```

© 黄湘云

gemo_hex 也是二维密度图的一种变体，特别适合数据量比较大的情形

```
ggplot(diamonds, aes(x = carat, y = price)) + geom_hex() +
  scale_fill_viridis_c()
```

© 黄湘云

heatmaps in ggplot2 二维密度图

```
ggplot(faithful, aes(x = eruptions, y = waiting)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon") +
  xlim(1, 6) +
  ylim(40, 100)

ggplot(faithful, aes(x = eruptions, y = waiting)) +
  stat_density2d(aes(fill = stat(level)), geom = "polygon") +
  scale_fill_viridis_c(option = "viridis") +
  xlim(1, 6) +
  ylim(40, 100)
```

提示

`MASS::kde2d()` 实现二维核密度估计，**ggplot2** 包提供了两种等价的绘图方式

1. `stat_density_2d()` 和 `..`
2. `stat_density2d()` 和 `stat()`

```
plotly::plot_ly(
  data = faithful, x = ~eruptions,
  y = ~waiting, type = "histogram2dcontour"
) %>%
  plotly::config(displayModeBar = FALSE)

# plot_ly(faithful, x = ~waiting, y = ~eruptions) %>%
```

© 黄湘云

(a) 默认调色板 (b) viridis 调色板

图 11.78: 二维密度图

```
#   add_histogram2d() %>%
#   add_histogram2dcontour()
```

延伸一下，热力图

```
library(KernSmooth)
den <- bkde2D(x = faithful, bandwidth = c(0.7, 7))
# 热力图
p1 <- plotly::plot_ly(x = den$x1, y = den$x2, z = den$fhat) %>%
  plotly::config(displayModeBar = FALSE) %>%
  plotly::add_heatmap()

# 等高线图
p2 <- plotly::plot_ly(x = den$x1, y = den$x2, z = den$fhat) %>%
  plotly::config(displayModeBar = FALSE) %>%
  plotly::add_contour()

htmltools::tagList(p1, p2)
```

11.4.10 提琴图

2004 年 Daniel Adler 开发 vioplot 包实现提琴图的绘制，它可能是最早实现此功能的 R 包，随后 10 余年没有更新却一直坚挺在 CRAN 上，非常难得，好在 Thomas Kelly 已经接手维护。另一款绘制提琴图的 R 包是 Peter Kampstra 开发的 beanplot [Kampstra, 2008]，也存在很多年了，不过随着时间的变迁，比较现代的方式是 **ggplot2** 带来的 `geom_violin()` 扔掉了很多依赖，也是各种图形的汇集地，可以看作是最佳实践。提琴图比起箱线图优势在于呈现更多的分布信息，其次在于更加美观，但是就目前来说箱线图的受众比提琴图要多很多，毕竟前者是包含更多统计信息，如图11.79 所示。

```
boxplot(count ~ spray, data = InsectSprays)
vioplot::vioplot(count ~ spray, data = InsectSprays, col = "lightgray")
ggplot(InsectSprays, aes(x = spray, y = count)) +
```

© 黄湘云


```
  geom_violin(fill = "lightgray") +
  theme_minimal()
beanplot::beanplot(count ~ spray, data = InsectSprays, col = "lightgray")
```

(a) 简单箱线图

(b) vioplot 绘制的提琴图

(c) ggplot2 绘制的提琴图

(d) beanplot 绘制的提琴图

图 11.79: 几种不同的提琴图

ggnormalviolin 包在给定均值和标准差的情况下，绘制正态分布的概率密度曲线，如图 11.80 所示。

```
library(ggnormalviolin)
with(
  aggregate(
    data = iris, Sepal.Length ~ Species,
    FUN = function(x) c(dist_mean = mean(x), dist_sd = sd(x))
  ),
  cbind.data.frame(Sepal.Length, Species)
```

© 王建硕


```
) %>%
  ggplot(aes(x = Species, mu = dist_mean, sigma = dist_sd, fill = Species)) +
  geom_normalviolin() +
  theme_minimal()
```

图 11.80: 正态分布的概率密度曲线

11.4.11 抖动图

抖动图适合数据量比较小的情况

```
ggplot(mpg, aes(x = class, y = hwy, color = class)) + geom_jitter()
```

© 黄湘云

抖不抖，还是抖一下

```
ggplot(iris, aes(x = Species, y = Sepal.Length)) +
  geom_point(aes(fill = Species), size = 5, shape = 21, colour = "grey20") +
  # geom_boxplot(outlier.colour = NA, fill = NA, colour = "grey20") +
  labs(title = "Not Jittered")
```

© 黄湘云

```
ggplot(iris, aes(x = Species, y = Sepal.Length)) +
  geom_point(aes(fill = Species),
    size = 5, shape = 21, colour = "grey20",
    position = position_jitter(width = 0.2, height = 0.1)
  ) +
  # geom_boxplot(outlier.colour = NA, fill = NA, colour = "grey20") +
  labs(title = "Jittered")
```

© 黄湘云

在数据量比较大的时候，可以用箱线图、密度图、提琴图

```
ggplot(sub_diamonds, aes(x = cut, y = price)) + geom_jitter()
```

上色和分面都不好使的抖动图，因为区分度变小

```
ggplot(sub_diamonds, aes(x = color, y = price, color = color)) +
  geom_jitter() +
  facet_grid(clarity ~ cut)
```

箱线图此时不宜分的过细

```
ggplot(diamonds, aes(x = color, y = price, color = color)) +
  geom_boxplot() +
  facet_grid(cut ~ clarity)
```

所以这样更好，先按纯净度分面，再对比不同的颜色，钻石价格的差异

```
ggplot(diamonds, aes(x = color, y = price, color = color)) +
  geom_boxplot() +
  facet_grid(~clarity)
```

最好只比较一个维度，不同颜色钻石的价格对比

```
ggplot(diamonds, aes(x = color, y = price, color = color)) +
  geom_boxplot()
```

设置随机数种子，抖动图是可重复的。

© 黄湘云

图 11.81: 抖动图的反例

```
ggplot(iris, aes(x = Species, y = Sepal.Width, color = Species)) +
  geom_boxplot(width = 0.65) +
  geom_point(position = position_jitter(seed = 37, width = 0.25))
```

© 黄湘云

图 11.82: 根据钻石颜色上色

© 黄湘云

图 11.83: 箱线图

© 黄湘云

图 11.84: 钻石按纯净度分面

© 黄湘云

图 11.85: 不同颜色钻石的价格比较

© 黄湘云

11.4.12 蜂群图

在样本点有限的情况下，用蜜蜂图代替普通的抖动图，可视化效果会好很多，如图 11.86 所示。Erik Clarke 开发的 ggbeeswarm 包可以将随机抖动的散点图朝着比较规律的方向聚合，又不丢失数据本身的准确性。

```
library(ggbeeswarm)
p1 <- ggplot(iris, aes(Species, Sepal.Length)) +
  geom_jitter() +
  theme_minimal()
p2 <- ggplot(iris, aes(Species, Sepal.Length)) +
  geom_quasirandom() +
  theme_minimal()
p1 + p2
```

图 11.86: 蜜蜂图可视化效果比抖动图好

11.4.13 玫瑰图

南丁格尔风玫瑰图[9] 可以作为堆积条形图，分组条形图

```
ggplot(diamonds, aes(x = color, fill = clarity)) +
  geom_bar()
```

```
ggplot(diamonds, aes(x = color, fill = clarity)) +
  geom_bar() +
  coord_polar()
```

```
# 风玫瑰图 http://blog.csdn.net/Bone_ACE/article/details/47624987
set.seed(2018)
# 随机生成100次风向，并汇集到16个区间内
```

[9] https://mbostock.github.io/protovis/ex/crimea-rose-full.html

© 黄湘云

图 11.87: 堆积条形图转风玫瑰图

图 11.88: 堆积条形图转风玫瑰图

© 基础统计


```
direction <- cut_interval(runif(100, 0, 360), n = 16)
# 随机生成100次风速，并划分成4种强度
mag <- cut_interval(rgamma(100, 15), 4)
dat <- data.frame(direction = direction, mag = mag)
# 将风向映射到X轴，频数映射到Y轴，风速大小映射到填充色，生成条形图后再转为极坐标形式即可
p <- ggplot(dat, aes(x = direction, y = ..count.., fill = mag))
p + geom_bar(colour = "white") +
  coord_polar() +
  theme(axis.ticks = element_blank(), axis.text.y = element_blank()) +
  labs(x = "", y = "", fill = "Magnitude")
```

图 11.89: 风玫瑰图

```
p + geom_bar(position = "fill") +
  coord_polar() +
  theme(axis.ticks = element_blank(), axis.text.y = element_blank()) +
  labs(x = "", y = "", fill = "Magnitude")
```

© 黄湘云

11.4.14 瓦片图

```
p1 <- expand.grid(months = month.abb, years = 1949:1960) %>%
  transform(num = as.vector(AirPassengers)) %>%
  ggplot(aes(x = years, y = months, fill = num)) +
  scale_fill_continuous(type = "viridis") +
  geom_tile(color = "white", size = 0.4) +
  scale_x_continuous(
    expand = c(0.01, 0.01),
    breaks = seq(1949, 1960, by = 1), labels = 1949:1960
  ) +
  theme_minimal(base_size = 10.54, base_family = "Noto Serif SC") +
  theme(legend.position = "top") +
  labs(x = "年", y = "月", fill = "人数")

p2 <- expand.grid(months = month.abb, years = 1949:1960) %>%
  transform(num = as.vector(AirPassengers)) %>%
  ggplot(aes(x = years, y = months, color = num)) +
  geom_point(pch = 15, size = 8) +
  scale_color_distiller(palette = "Spectral") +
  scale_x_continuous(
    expand = c(0.01, 0.01),
    breaks = seq(1949, 1960, by = 1), labels = 1949:1960
```

```
  ) +
  theme_minimal(base_size = 10.54, base_family = "Noto Serif SC") +
  theme(legend.position = "top") +
  labs(x = "年", y = "月", color = "人数")
p1 + p2
```

图 11.90: 1949-1960 年国际航线乘客数量的月度趋势

11.4.15 日历图

airquality 数据集记录了 1973 年 5 月至 9 月纽约的空气质量，包括气温（华氏度）、风速（米/小时）、紫外线强度、臭氧含量四个指标，图 11.91 展示了每日的气温变化。

```
airquality %>%
  transform(Date = seq.Date(
    from = as.Date("1973-05-01"),
    to = as.Date("1973-09-30"), by = "day"
  )) %>%
  transform(
    Week = as.integer(format(Date, "%W")),
    Year = as.integer(format(Date, "%Y")),
    Weekdays = factor(weekdays(Date, abbreviate = T),
      levels = c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")
    )
  ) %>%
  ggplot(aes(x = Week, y = Weekdays, fill = Temp)) +
  scale_fill_distiller(name = "Temp (F)", palette = "Spectral") +
  geom_tile(color = "white", size = 0.4) +
  facet_wrap("Year", ncol = 1) +
```

© 黄湘云


```
  scale_x_continuous(
    expand = c(0, 0),
    breaks = seq(1, 52, length = 12),
    labels = month.abb
  )
```

图 11.91: 1973 年 5 月至 9 月纽约的气温变化

注意

图 11.91 横轴的刻度标签换成了月份，一个月为四周，一年 52 ~ 53 周，每周的第一天约定为星期一，1973 年 05 月 01 日为星期二。代码中颇为技巧的在于 `format()` 函数从 Date 日期类型的数据提取第几周，用 `weekdays()` 函数提取星期几，而 month.abb 则是一个内置常量，12 个月份的英文缩写。在调用其它 R 包处理日期数据时要特别小心，要留意一周的的第一天是星期几，有的是星期一，有的是星期日，这往往和宗教信仰相关，星期日在西方也叫礼拜天。上面 Base R 提供的日期函数认为一周的第一天是星期一，而调用 **data.table** 的话，默认一周是从星期日（礼拜天）开始的。

```
# https://d.cosx.org/d/421230
weekdays(Sys.Date(), abbreviate = TRUE)
```

```
## [1] "Wed"
```

```
data.table::wday(Sys.Date())
```

```
## [1] 4
```

```
library(gert)
library(ggplot2)
git_config_set("user.name", "XiangyunHuang")
git_config_set("user.email", "xiangyunfaith@outlook.com")

dat <- git_log(max = 1000)
# format(time, "%a") 本地 MacOS 环境 "六" "四" "五" 表示星期
# Sys.getlocale("LC_TIME") # "zh_CN.UTF-8"
```

© 基础统计

```
lvls <- if(!is.na(Sys.getenv("CI", NA))) {
  c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")
} else {
  c("日", "一", "二", "三", "四", "五", "六")
}

dat <- transform(dat,
  date = format(time, "%Y-%m-%d"),
  year = format(time, "%Y") ,
  month = format(time, "%m"),
  weekday = format(time, "%a"), # factor(format(time, "%a"), levels = lvls),
  week = as.integer(format(time, "%W"))
)
```

本书的活跃情况

```
dat1 <- aggregate(formula = commit ~ year + month, data = dat, FUN = length)
# 条形图
ggplot(data = dat1, aes(x = month, y = commit, fill = year)) +
  geom_bar(stat = "identity", position = "identity")
```

```
# 日历图
dat2 <- aggregate(formula = commit ~ year + week + weekday, data = dat, FUN = length)
```

© 统计出版社


```
dat2 <- transform(dat2, colorBin = cut(commit, breaks = c(0, 5, 10, 15, 20, 25)))

ggplot(data = dat2, aes(x = week, y = weekday, fill = colorBin)) +
  scale_fill_brewer(name = "commit", palette = "Greens") +
  geom_tile(color = "white", size = 0.4) +
  facet_wrap("year", ncol = 1) +
  scale_x_continuous(
    expand = c(0, 0),
    breaks = seq(1, 52, length = 12),
    labels = month.abb
  ) +
  labs(x = "", y = "") +
  theme_minimal(base_family = "Noto Sans SC")
```

图 11.92: 《现代统计图形》的活跃情况

11.4.16 岭线图

ggridges 包，于淼 对此图形的来龙去脉做了比较系统的阐述，详见统计之都主站文章叠嶂图的前世今生

```
library(ggridges)
ggplot(lincoln_weather, aes(x = `Mean Temperature [F]`, y = Month, fill = stat(x))) +
  geom_density_ridges_gradient(scale = 3, rel_min_height = 0.01, gradient_lwd = 1.) +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_discrete(expand = expansion(mult = c(0.01, 0.25))) +
  scale_fill_viridis_c(name = "Temp. [F]", option = "C") +
  labs(
    title = 'Temperatures in Lincoln NE',
    subtitle = 'Mean temperatures (Fahrenheit) by month for 2016'
```

© 黄湘云


```
  ) +
  theme_ridges(font_size = 13, grid = TRUE) +
  theme(axis.title.y = element_blank())
```

图 11.93: 2016 年在内布拉斯加州林肯市的天气变化

通过数据可视化的手段帮助肉眼检查两组数据的分布

```
p1 <- ggplot(sleep, aes(x = extra, y = group, fill = group)) +
  geom_density_ridges() +
  theme_ridges()

p2 <- ggplot(diamonds, aes(x = price, y = color, fill = color)) +
  geom_density_ridges() +
  theme_ridges()

p1 / p2
```

ridgeline 提供 Base R 绘图方案

11.4.17 椭圆图

type 指定多元分布的类型，`type = "t"` 和 `type = "norm"` 分别表示 t 分布和正态分布，`geom = "polygon"`，以 `eruptions > 3` 分为两组

© 黄湘云

图 11.94: 比较数据的分布

© 黄湘云

图 11.95: 岭线图

```
ggplot(faithful, aes(x = waiting, y = eruptions)) +
  geom_point() +
  stat_ellipse()

ggplot(faithful, aes(waiting, eruptions, color = eruptions > 3)) +
  geom_point() +
  stat_ellipse(type = "norm", linetype = 2) +
  stat_ellipse(type = "t") +
  theme(legend.position = "none")

ggplot(faithful, aes(waiting, eruptions, fill = eruptions > 3)) +
  stat_ellipse(geom = "polygon") +
  theme(legend.position = "none")
```

11.4.18 Q-Q 图

quantile-quantile Q-Q 正态分布图的 ggplot2 实现 qqplotr

11.4.19 包络图

ggpubr 包提供了 `stat_chull()` 图层

© 黄湘云

(a) 简单椭圆图

(b) 正态和 t 分布

(c) 填充几何图形

图 11.96: 几种不同的椭圆图

© 黄湘云


```
library(ggpubr)
ggscatter(mpg, x = "displ", y = "hwy", color = "drv")+
 stat_chull(aes(color = drv, fill = drv), alpha = 0.1, geom = "polygon")
```

图 11.97: 包络图

其背后的原理如下

```
stat_chull
```

```
## function (mapping = NULL, data = NULL, geom = "path", position = "identity",
##     na.rm = FALSE, show.legend = NA, inherit.aes = TRUE, ...)
## {
##     layer(stat = StatChull, data = data, mapping = mapping, geom = geom,
##         position = position, show.legend = show.legend, inherit.aes = inherit.aes,
##         params = list(na.rm = na.rm, ...))
## }
## <bytecode: 0x560a4bca89d0>
## <environment: namespace:ggpubr>
```

```
StatChull <- ggproto("StatChull", Stat,
  compute_group = function(data, scales) {
    data[chull(data$x, data$y), , drop = FALSE]
  },
  required_aes = c("x", "y")
)

stat_chull <- function(mapping = NULL, data = NULL, geom = "polygon",
```

© 科学出版社


```
                        position = "identity", na.rm = FALSE, show.legend = NA,
                        inherit.aes = TRUE, ...) {
  layer(
    stat = StatChull, data = data, mapping = mapping, geom = geom,
    position = position, show.legend = show.legend, inherit.aes = inherit.aes,
    params = list(na.rm = na.rm, ...)
  )
}

ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  stat_chull(fill = NA, colour = "black")

ggplot(mpg, aes(displ, hwy, colour = drv)) +
  geom_point() +
  stat_chull(fill = NA)
```

11.4.20 拟合图

```
xx <- -9:9
yy <- sqrt(abs(xx))
plot(xx, yy,
  col = "red",
  xlab = expression(x),
  ylab = expression(sqrt(abs(x)))
)
lines(spline(xx, yy, n = 101, method = "fmm", ties = mean), col = "pink")

myspline <- function(formula, data, ...) {
  dat <- model.frame(formula, data)
  res <- splinefun(dat[[2]], dat[[1]])
  class(res) <- "myspline"
  res
}

predict.myspline <- function(object, newdata, ...) {
  object(newdata[[1]])
}

data.frame(x = -9:9) %>%
  transform(y = sqrt(abs(x))) %>%
  ggplot(aes(x = x, y = y)) +
  geom_point(color = "red", pch = 1, size = 2) +
  stat_smooth(method = myspline, formula = y~x, se = F, color = "pink") +
```

© 谢湘


```
  labs(x = expression(x), y = expression(sqrt(abs(x)))) +
  theme_minimal()
```

图 11.98: 自定义样条函数

下面以真实数据集 trees 为例，介绍 `geom_smooth()` 支持的拟合方法，比如 `"lm"` 线性回归和 `"nls"` 非线性回归

```
ggplot(trees, aes(x = log(Girth), y = log(Volume))) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE)

ggplot(trees, aes(x = Girth, y = Volume)) +
  geom_point() +
  geom_smooth(
    method = "nls", formula = y ~ a * x^2 + b, se = F,
    method.args = list(start = list(a = 5, b = -36))
  )
```

11.4.21 地形图

区域之间以轮廓分割，轮廓之间以相同的颜色填充，Cleveland 把这个叫做 level plot，**lattice** 包中 `levelplot()` 函数正来源于此。

Auckland's Maunga Whau Volcano 是火山喷发后留下的碴堆，位于新西兰奥克兰伊甸山郊区。Ross Ihaka 收集了它的地形数据，命名为 volcano，打包在 R 软件环境中，见图 11.100

```
filled.contour(volcano,
  color.palette = terrain.colors,
  plot.title = title(
    main = "The Topography of Maunga Whau",
    xlab = "Meters North", ylab = "Meters West"
  ),
```

© 黄湘云

图 11.99: 平滑方法

```
  plot.axes = {
    axis(1, seq(100, 800, by = 100))
    axis(2, seq(100, 600, by = 100))
  },
  key.title = title(main = "Height\n(meters)"),
  key.axes = axis(4, seq(90, 190, by = 10))
)
```

11.4.22 树状图

数据集 GNI2014 来自 **treemap** 包，是一个 data.frame 类型的数据对象，记录了 2014 年每个国家的人口总数 population 和国民人均收入 GNI，数据样例见下方：

```
library(treemap)
data(GNI2014, package = "treemap")
subset(GNI2014, subset = grepl(x = country, pattern = 'China'))
```

```
##    iso3              country continent population   GNI
## 7   MAC     Macao SAR, China      Asia     559846 76270
## 33  HKG Hong Kong SAR, China      Asia    7061200 40320
## 87  CHN                China      Asia 1338612970  7400
```

数据呈现明显的层级结构，从大洲到国家记录人口数量和人均收入，矩阵树图以方块大小表示人口数量，以颜色深浅表示人均收入，见图11.101

```
treemap(GNI2014,
  index = c("continent", "iso3"),
  vSize = "population",
  vColor = "GNI",
  type = "value",
```

© 黄湘云

图 11.100: image 图形

© 黄湘云


```
  format.legend = list(scientific = FALSE, big.mark = " ")
)
```

图 11.101: 矩阵树图

treemapify 包基于 ggplot2 制作树状图，类似地，该 R 包内置了数据集 G20，记录了世界主要经济体 G20 (https://en.wikipedia.org/wiki/G20) 的经济和人口信息，国家 GDP（单位：百万美元）`gdp_mil_usd` 和人类发展指数 `hdi`。相比于 GNI2014，它还包含了两列标签信息：经济发展阶段和所处的半球。图 @(fig:treemap-ggplot2) 以南北半球 hemisphere 分面，以色彩填充区域 region，以 `gdp_mil_usd` 表示区域大小

```
library(treemapify)
ggplot(G20, aes(
  area = gdp_mil_usd, fill = region,
  label = country, subgroup = region
)) +
  geom_treemap() +
  geom_treemap_text(grow = T, reflow = T, colour = "black") +
  facet_wrap(~hemisphere) +
  scale_fill_brewer(palette = "Set1") +
  theme(legend.position = "bottom") +
  labs(
    title = "The G-20 major economies by hemisphere",
```

© 清华大学出版社


```
    caption = "The area of each tile represents the country's GDP as a
      proportion of all countries in that hemisphere",
    fill = "Region"
  )
```

图 11.102: 世界主要经济体 G20 的人口和经济信息

11.4.23 留存图

```
cohort <- data.frame(
  cohort = rep(1:5, times = 5:1),
  week = c(1:5, 1:4, 1:3, 1:2, 1),
  value = c(
    75, 73, 54, 23, 3,
    98, 94, 70, 25,
    52, 5, 3,
    44, 15,
    88
  )
)
```

© 黄湘石


```
ggplot(cohort, aes(x = week, y = cohort, fill = value)) +
  geom_tile(color = "white") +
  geom_text(aes(label = value), color = "white") +
  scale_y_reverse() +
  scale_fill_binned(type = "viridis")
```

留存是 Cohort 分析 中的一种情况，还有转化等，首先定义你的问题，确定度量问题的指标，确定和问题相关的 Cohort （比如时间、空间和用户属性等关键的影响因素），然后数据处理、可视化获得 Cohort 分析结果，最后在实际决策和行动中检验分析结论。

11.4.24 瀑布图

瀑布图 waterfall 与上月相比，谁增谁减，用瀑布图分别表示占比和绝对数值。瀑布图 waterfall

```
balance <- data.frame(
  event = c(
    "Starting\nCash", "Sales", "Refunds",
    "Payouts", "Court\nLosses", "Court\nWins", "Contracts", "End\nCash"
  ),
  change = c(2000, 3400, -1100, -100, -6600, 3800, 1400, -2800)
)

balance$balance <- cumsum(c(0, balance$change[-nrow(balance)])) # 累计值
```

©黄天宇

```
balance$time <- 1:nrow(balance)
balance$flow <- factor(sign(balance$change)) # 变化为正还是为负

ggplot(balance) +
  geom_hline(yintercept = 0, colour = "white", size = 2) +
  geom_rect(aes(
    xmin = time - 0.45, xmax = time + 0.45,
    ymin = balance, ymax = balance + change, fill = flow
  )) +
  geom_text(aes(
    x = time,
    y = pmin(balance, balance + change) - 50,
    label = scales::dollar(change)
  ),
  hjust = 0.5, vjust = 1, size = 3
  ) +
  scale_x_continuous(
    name = "",
    breaks = balance$time,
    labels = balance$event
  ) +
  scale_y_continuous(
    name = "Balance",
    labels = scales::dollar
  ) +
  scale_fill_brewer(palette = "Spectral") +
  theme_minimal()
```

```
library(ggplot2)
# AtherEnergy/ggTimeSeries
# 个人收入，国家地区收入
library(ggTimeSeries) # https://github.com/AtherEnergy/ggTimeSeries
dat <- data.frame(year = 2000:2021, dpc = 10:31)
ggplot(data = dat, aes(x = year, y = dpc)) +
  stat_waterfall()
```

11.4.25 桑基图

ggalluvial

```
titanic_wide <- data.frame(Titanic)
```

```
head(titanic_wide)
```

```
##   Class    Sex   Age Survived Freq
## 1   1st   Male Child       No    0
```

© 黄湘云

图 11.103: 瀑布图

```
## 2   2nd   Male Child       No    0
## 3   3rd   Male Child       No   35
## 4  Crew   Male Child       No    0
## 5   1st Female Child       No    0
## 6   2nd Female Child       No    0
```

```
library(ggalluvial)
ggplot(data = titanic_wide,
       aes(axis1 = Class, axis2 = Sex, axis3 = Age,
           y = Freq)) +
  scale_x_discrete(limits = c("Class", "Sex", "Age"), expand = c(.2, .05)) +
  xlab("Demographic") +
  geom_alluvium(aes(fill = Survived)) +
  geom_stratum() +
  geom_text(stat = "stratum", aes(label = after_stat(stratum))) +
  theme_minimal() +
  ggtitle("passengers on the maiden voyage of the Titanic",
          "stratified by demographics and survival")
```

11.4.26 词云图

词云 ggwordcloud

© 黄湘云

图 11.104: 桑基图

11.4.27 甘特图

描述项目进展的甘特图 ganttrify

11.4.28 马赛克图

```
library(ggmosaic)
ggplot(data = as.data.frame(UCBAdmissions)) +
  geom_mosaic(aes(weight = Freq, x = product(Gender, Admit), fill = Dept)) +
  coord_flip() +
  theme_minimal() +
  labs(x = "Admit", y = "Gender")
```

```
## Warning: `unite_()` was deprecated in tidyr 1.2.0.
## Please use `unite()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

11.4.29 凹凸图

ggbump 排序随位置的变化

© 黄湘云

图 11.105: UCBAdmissions 马赛克图

```
# remotes::install_github("davidsjoberg/ggbump")
library(ggbump)
# 代码修改自 https://github.com/davidsjoberg/ggbump
df <- data.frame(
  season = c(
    "Spring", "Pre-season", "Summer", "Season finale", "Autumn", "Winter",
    "Spring", "Pre-season", "Summer", "Season finale", "Autumn", "Winter",
    "Spring", "Pre-season", "Summer", "Season finale", "Autumn", "Winter",
    "Spring", "Pre-season", "Summer", "Season finale", "Autumn", "Winter"
  ),
  rank = c(
    1, 3, 4, 2, 1, 4,
    2, 4, 1, 3, 2, 3,
    4, 1, 2, 4, 4, 1,
    3, 2, 3, 1, 3, 2
  ),
  player = c(
    rep("David", 6),
    rep("Anna", 6),
    rep("Franz", 6),
    rep("Ika", 6)
```

© 王建硕


```
  )
)

# Create factors and order factor
df <- transform(df, season = factor(season, levels = unique(season)))

# Add manual axis labels to plot
ggplot(df, aes(season, rank, color = player)) +
  geom_bump(size = 2, smooth = 20, show.legend = F) +
  geom_point(size = 5, aes(shape = player)) +
  theme_minimal(base_size = 10, base_line_size = 0) +
  theme(panel.grid.major = element_blank(),
        axis.ticks = element_blank()) +
  scale_color_manual(values = RColorBrewer::brewer.pal(name = "Set2", n = 4))
```

图 11.106: 凹凸图

11.4.30 水流图

常用于时间序列数据展示的堆积区域图，ggstream 和 streamgraph

```
library(ggstream)

ggplot(blockbusters, aes(year, box_office, fill = genre)) +
  geom_stream() +
  theme_minimal()
```

© 黄湘云

图 11.107: 堆积区域图

11.4.31 时间线

```
# 交互动态图 https://github.com/shosaco/vistime
# 刘思Ⓕ 2018 数据科学的时间轴 https://bjt.name/2018/11/18/timeline.html
x <- read.table(
  textConnection("
The Future of Data Analysis,1962
Relational Database,1970
Data science(Peter Naur),1974
Two-Way Communication,1975
Exploratory Data Analysis,1977
Business Intelligence,1989
The First Database Report,1992
The World Wide Web Explodes,1995
Data Mining and Knowledge Discovery,1997
S(ACM Software System Award),1998
Statistical Modeling: The Two Cultures,2001
Hadoop,2006
Data scientist,2008
NOSQL,2009
Deep Learning,2015
"),
  sep = ","
)
names(x) <- c("Event", "EventDate")
x$EventDate <- as.Date(paste(x$EventDate, "/01/01", sep = ""))
```

© 基础统计

```
library(timelineS)
timelineS(x,
  labels = paste(x[[1]], format(x[[2]], "%Y")),
  line.color = "blue", label.angle = 15
)
```

图 11.108: 数据科学的时间轴

```
library(timeline)
data(ww2, package = 'timeline')
timeline(ww2, ww2.events, event.spots=2, event.label='', event.above=FALSE)
```

```
# 适合放在动态幻灯片
# 美团风格的写轮眼
# 时间线
library(vistime)
# presidents and vice presidents
pres <- data.frame(
  Position = rep(c("President", "Vice"), each = 3),
  Name = c("Washington", rep(c("Adams", "Jefferson"), 2), "Burr"),
  start = c("1789-03-29", "1797-02-03", "1801-02-03"),
  end = c("1797-02-03", "1801-02-03", "1809-02-03"),
  color = c("#cbb69d", "#603913", "#c69c6e")
)

hc_vistime(pres, col.event = "Position", col.group = "Name",
           title = "Presidents of the USA")
```

© 基础图

11.4.32 三元图

Ternary 使用基础图形库，而 ggtern 使用 ggplot2 绘制

```
library(ggtern)
library(ggalt)
data("Fragments")
ggtern(Fragments, aes(
  x = Qm, y = Qp, z = Rf + M,
  fill = GrainSize, shape = GrainSize
)) +
  geom_encircle(alpha = 0.5, size = 1) +
  geom_point() +
  labs(
    title = "Example Plot",
    subtitle = "using geom_encircle"
  ) +
  theme_bw() +
  theme_legend_position("tr")
```

11.4.33 向量场图

```
library(ggquiver)
```

11.4.34 四象限图

```
dat <- data.frame(
  perc = c(54, 18, 5, 15),
  wall_policy = c("oppose", "favor", "oppose", "favor"),
  dreamer_policy = c("favor", "favor", "oppose", "oppose"),
  stringsAsFactors = FALSE
) %>%
  transform(
    xmin = ifelse(wall_policy == "oppose", -sqrt(perc), 0),
    xmax = ifelse(wall_policy == "favor", sqrt(perc), 0),
    ymin = ifelse(dreamer_policy == "oppose", -sqrt(perc), 0),
    ymax = ifelse(dreamer_policy == "favor", sqrt(perc), 0)
  )

ggplot(data = dat) +
  geom_rect(aes(
    xmin = xmin, xmax = xmax,
    ymin = ymin, ymax = ymax
  ), fill = "grey") +
```

© 黄湘云

```
  geom_text(aes(
    x = xmin + 0.5 * sqrt(perc),
    y = ymin + 0.5 * sqrt(perc),
    label = perc
  ),
  color = "white", size = 10
  ) +
  coord_equal() +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  theme_minimal() +
  labs(x = "", y = "", title = "")
```

图 11.109: 四象限图

11.4.35 韦恩图

ggVennDiagram

11.4.36 龙卷风图

© 王建平


```
dat <- data.frame(
  variable = c("A", "B", "A", "B"),
  Level = c("Top-2", "Top-2", "Bottom-2", "Bottom-2"),
  value = c(.8, .7, -.2, -.3)
)
ggplot(dat, aes(x = variable, y = value, fill = Level)) +
  geom_bar(position = "identity", stat = "identity") +
  scale_y_continuous(labels = abs) +
  coord_flip() +
  theme_minimal()
```

图 11.110: 龙卷风图展示变量重要性

Tornado diagram 主要用于敏感性分析，比较不同变量的重要性程度。条形图 `geom_bar()` 图层的变体，模型权重可视化的手段，仅限于广义线性模型。

11.4.37 聚类图

ggdendro 的 `dendro_data()` 函数支持 `tree` 、`hclust` 、`dendrogram` 和 `rpart` 结果的整理，进而绘图

```
library(ggdendro)
hc <- hclust(dist(USArrests), "ave")
hcdata <- dendro_data(hc, type = "rectangle")
ggplot() +
  geom_segment(data = segment(hcdata),
               aes(x = x, y = y, xend = xend, yend = yend)
```

© 黄湘云

```
  ) +
  geom_text(data = label(hcdata),
            aes(x = x, y = y, label = label, hjust = 0),
            size = 3
  ) +
  coord_flip() +
  scale_y_reverse(expand = c(0.2, 0)) +
  theme_minimal()
```

11.4.38 主成分图

借助 **autoplotly** 包 [Tang, 2018] 可将函数 `stats::prcomp` 生成的结果转化为交互图形

```
pca <- prcomp(iris[c(1, 2, 3, 4)])
plot(pca)
```

© 黄湘云

```
library(autoplotly)
autoplotly(pca,
  data = iris, colour = "Species",
  label = TRUE, label.size = 3, frame = TRUE
)
```

ggfortify [Tang et al., 2016] 包将主成分分析图转化为静态图形

```
library(ggfortify)
autoplot(pca, data = iris, colour = 'Species')
```

11.4.39 组合图

组合的意思是将不同种类的图形绘制在一个区域中，比如密度曲线和地毯图[10]组合。**GGally**、**ggupset**、ggcharts 和 **ggpubr** 高度定制了一些组合统计图形，以 ggpubr 为例，见图 11.112。

```
library(ggpubr)
ggdensity(sleep,
  x = "extra", add = "mean", rug = TRUE, color = "group",
  fill = "group", palette = c("#00AFBB", "#E7B800")
)
```

上面介绍的都是已经固化的组合方式，一般地，将多个图形组合到一个图中，可以有很多办法，比如 Claus Wilke 开发的 **cowplot**，在他的书里 Fundamentals of Data Visualization 大量使用，后起之秀 **patchwork**

[10]其实是轴须图 rug plot，只因样子看起来像铺在地上的毛毯，故而称之为地毯图，对应于 R 内置的 `rug()` 函数或 ggplot2 提供的图层 `geom_rug()`，更多解释详见 https://en.wikipedia.org/wiki/Rug_plot。

© 黄湘云

图 11.111: 主成分分析

图 11.112: 组合图形

© 黄湘云

表 11.3: 吲哚美辛在人体中的代谢情况

Subject	0.25	0.5	0.75	1	1.25	2	3	4	5	6	8
1	1.50	0.94	0.78	0.48	0.37	0.19	0.12	0.11	0.08	0.07	0.05
2	2.03	1.63	0.71	0.70	0.64	0.36	0.32	0.20	0.25	0.12	0.08
3	2.72	1.49	1.16	0.80	0.80	0.39	0.22	0.12	0.11	0.08	0.08
4	1.85	1.39	1.02	0.89	0.59	0.40	0.16	0.11	0.10	0.07	0.07
5	2.05	1.04	0.81	0.39	0.30	0.23	0.13	0.11	0.08	0.10	0.06
6	2.31	1.44	1.03	0.84	0.64	0.42	0.24	0.17	0.13	0.10	0.09

则提供更加简洁的组合语法，非常受欢迎，更加底层的拼接方法可以去看 一页多图 和 R 内置的 grid 系统。

11.4.40 动态图

av 包基于 FFmpeg 将静态图片合成视频，而 **gifski** 包基于 gifski 将静态图片合成 GIF 动画，**animation** 包 [Xie, 2013] 将 Base R 绘制的图形转化为动画或视频，mapmate 制作地图相关的三维可视化图形，**gganimate** 包支持将 ggplot2 生成的图形，**magick** 可以将一系列静态图形合成动态图形，借助 **gifski** 包转化为动态图片或视频。推荐读者从 gganimate 案例合集 开始制作动态图形。**rgl** 可以制作真三维动态图形，支持缩放、拖拽、旋转等操作，**rayshader** 还支持转化 ggplot2 对象为 3D 图形。

数据集 Indometh 记录了药物在人体中的代谢情况，给 6 个人分别静脉注射了吲哚美辛，每隔一段时间抽血检查药物在血浆中的浓度，收集的数据见表 11.3

```
reshape(Indometh, v.names = "conc", idvar = "Subject",
        timevar = "time", direction = "wide", sep = "") %>%
  knitr::kable(.,
    caption = "吲哚美辛在人体中的代谢情况",
    row.names = FALSE, col.names = gsub("(conc)", "", names(.)),
    align = "c"
  )
```

如图 11.113 所示，药物在人体中浓度变化情况

```
p <- ggplot(
  data = Indometh,
  aes(x = time, y = conc, color = Subject)
) +
  geom_point() +
  geom_line() +
  theme_minimal() +
  labs(
    x = "time (hr)",
    y = "plasma concentrations of indometacin (mcg/ml)"
  )
p
```

© 黄湘云

图 11.113: 药物在人体中的代谢情况

```
library(gganimate)
p + transition_reveal(time)
```

© 黄湘云

提示

书籍目标输出格式是 PDF，则在代码块选项设置里必须指定参数 `fig.show='animate'` 否则插入的只是图片而不是动画，目标格式是 HTML 网页，就不必指定参数，默认会将图片合成 GIF 动态图，嵌入 PDF 里面的动画需要 Acrobat Reader 阅读器才能正确地显示。

动态图形制作的原理，简单来说，就是将一帧帧静态图形以较快的速度播放，人眼形成视觉残留，以为是连续的画面，相比于 animation，**gganimate** 借助 **tweenr** 包添加了过渡效果，动态图形显得非常自然。下面以 cup 函数[11]为例

$$f(x;\theta,\phi)=\theta x\log(x)-\frac{1}{\phi}e^{-\phi^4(x-\frac{1}{e})^4},\quad \theta\in(2,3),\phi\in(30,50),x\in(0,1)$$

函数图像随着 θ 和 ϕ 的变化情况见图 11.114。

```
library(tweenr)
cup_curve <- function(n = 100, theta = 3, phi = 30, cup = "A") {
  data.frame(x = seq(0.00001, 1, length.out = n), cup = cup) %>%
    transform(y = theta * x * log(x, base = 10)
              - 1 / phi * exp(-(phi * x - phi / exp(1))^4))
}
mapply(
  FUN = cup_curve, theta = c(E = 3, D = 2.8, C = 2.5, B = 2.2, A = 2),
  phi = c(30, 33, 36, 40, 50), cup = c("E", "D", "C", "B", "A"),
  MoreArgs = list(n = 50), SIMPLIFY = FALSE, USE.NAMES = TRUE
```

[11]函数来自余光创的博客 — 3D 版邪恶的曲线，此处借用 gganimate 将其动态化，前方高能，少儿不宜，R 还能这么不正经的玩。

© 王建硕

```
) %>%
  tween_states(
    data = .,
    tweenlength = 2, statelength = 1,
    ease = rep("cubic-in-out", 4), nframes = 100
  ) %>%
  ggplot(data = ., aes(x, y, color = cup, frame = .frame)) +
  geom_path() +
  coord_flip() +
  theme_void()
```

图 11.114: 添加过渡效果

© 黄湘云

第十二章　交互图形

提示

plotly 包的函数使用起来还是比较复杂的，特别是需要打磨细节以打造数据产品时，此外，其依赖相当重，仅数据处理就包含两套方法 — dplyr 和 data.table，引起很多函数冲突，可谓「苦其久矣」！因此，准备另起炉灶，开发一个新的 R 包 qplotly，取意 quick plotly，以 `qplot_ly()` 替代 `plot_ly()`。类似简化 API 的工作有 simplevis、autoplotly、ggfortify 和 plotme。
plotly 团队开发了 plotly.js 库，且维护了 R 接口文档 (https://plotly.com/r/)，Carson Sievert 开发了 plotly 包，配套书 Interactive web-based data visualization with R, plotly, and shiny。Paul C. Bauer 的书 Applied Data Visualization 介绍 plotly https://bookdown.org/paul/applied-data-visualization/what-is-plotly.html

echarts4r 包基于 Apache ECharts (incubating)，ECharts 的 Python 接口 pyecharts 也非常受欢迎，基于 apexcharts.js 的 apexcharter。ECharts2Shiny 包将 ECharts 嵌入 shiny 框架中。

timevis 创建交互式的时间线的时序可视化，它基于 Vis 的 vis-timeline 模块，支持 shiny 集成。dygraphs 包基于 dygraphs 可视化库，将时序数据可视化，更多情况见 https://dygraphs.com/。leaflet 提供 leaflet 的 R 接口。rAmCharts4 基于 amCharts 4 库，apexcharter 提供 apexcharts.js 的 R 接口。还有 billboarder 等。更完整地，请看 Etienne Bacher 维护的 R 包列表 r-js-adaptation 。

对于想了解 htmlwidgets 框架，JavaScript 响应式编程的读者，推荐 John Coene 新书 JavaScript for R

提示

学习 plotly 和 highcharter 为代表的基于 JavaScript 的 R 包，共有四重境界：第一重是照着帮助文档的示例，示例有啥我们做啥；第二重是明白帮助文档中 R 函数和 JavaScript 函数的对应关系，能力达到 JS 库的功能边界；第三重是深度自定义一些扩展性的 JS 功能，放飞自我；第四重是重新造轮子，为所欲为。下面的介绍希望能帮助读者到达第二重境界。

plotly 是一个功能非常强大的绘制交互式图形的 R 包。它支持下载图片、添加水印、自定义背景图片、工具栏和注释[1] 等一系列细节的自定义控制。下面结合 JavaScript 库 plotly.js 一起介绍，帮助文档 `?config` 没有太详细地介绍，所以我们看看 `config()` 函数中参数 `...` 和 JavaScript 库 plot_config.js 中的功能函数是怎么对应的。图中图片下载按钮对应 `toImageButtonOptions` 参数，看 toImageButtonOptions 源代码，可知，它接受任意数据类型，对应到 R 里面就是列表。`watermark` 和 `displaylogo` 都是传递布尔值 (TRUE/FALSE)，具体根据 JavaScript 代码中的 valType （参数值类型）决定，其它参数类似。另一个函数 layout 和函数 `config()` 是类似的，怎么传递参数值是根据 JavaScript 代码来的。

```
toImageButtonOptions: {
    valType: 'any',
```

[1] https://plotly.com/r/reference/#layout-scene-annotations-items-annotation-font

© 基础设施

```
    dflt: {},
    description: [
        'Statically override options for toImage modebar button',
        'allowed keys are format, filename, width, height, scale',
        'see ../components/modebar/buttons.js'
    ].join(' ')
},
displaylogo: {
    valType: 'boolean',
    dflt: true,
    description: [
        'Determines whether or not the plotly logo is displayed',
        'on the end of the mode bar.'
    ].join(' ')
},
watermark: {
    valType: 'boolean',
    dflt: false,
    description: 'watermark the images with the company\'s logo'
},
```

```
library(plotly, warn.conflicts = FALSE)
plot_ly(diamonds,
  x = ~clarity, y = ~price,
  color = ~clarity, colors = "Set1", type = "box"
) %>%
  config(
    toImageButtonOptions = list(
      format = "svg", width = 450, height = 300,
      filename = paste("plot", Sys.Date(), sep = "_")
    ),
    modeBarButtons = list(list("toImage")),
    watermark = FALSE,
    displaylogo = FALSE,
    locale = "zh-CN",
    staticPlot = TRUE,
    showLink = FALSE,
    modeBarButtonsToRemove = c(
      "hoverClosestCartesian", "hoverCompareCartesian",
      "zoom2d", "zoomIn2d", "zoomOut2d",
      "autoScale2d", "resetScale2d", "pan2d",
      "toggleSpikelines"
    )
  ) %>%
  layout(
```

© 王建硕


```
    template = "plotly_dark",
    images = list(
      source = "https://images.plot.ly/language-icons/api-home/r-logo.png",
      xref = "paper",
      yref = "paper",
      x = 1.00,
      y = 0.25,
      sizex = 0.2,
      sizey = 0.2,
      opacity = 0.5
    ),
    annotations = list(
      text = "DRAFT",                # 水印文本
      textangle = -30,               # 逆时针旋转 30 度
      font = list(
        size = 40,                   # 字号
        color = "gray",              # 颜色
        family = "Times New Roman"   # 字族
      ),
      opacity = 0.2,                 # 透明度
      xref = "paper",
      yref = "paper",
      x = 0.5,
      y = 0.5,
      showarrow = FALSE              # 去掉箭头指示
    )
  )
```

表 12.1: 交互图形的设置函数 `config()` 各个参数及其作用（部分）

参数	作用
displayModeBar	是否显示交互图形上的工具条，默认显示 `TRUE`[2]。
modeBarButtons	工具条上保留的工具，如下载 `"toImage"`，缩放 `"zoom2d"`[3]。
modeBarButtonsToRemove	工具条上要移除的工具，如下载和缩放图片 `c("toImage", "zoom2d")`。
toImageButtonOptions	工具条上下载图片的选项设置，包括名称、类型、尺寸等。[4]
displaylogo	是否交显示互图形上 Plotly 的图标，默认显示 `TRUE`[5]。
staticPlot	是否将交互图形转为静态图形，默认 `FALSE`。
locale	本土化语言设置，比如 `"zh-CN"` 表示中文。

[2]https://plotly-r.com/control-modebar.html。

[3]完整的列表见 https://github.com/plotly/plotly.js/blob/master/src/components/modebar/buttons.js。

[4]设置下载图片的尺寸，还可设置为 PNG 格式，SVG 格式图片，可借助 **rsvg** 的 `rsvg_pdf()` 函数转化为 PDF 格式 https://github.com/ropensci/plotly/issues/1556#issuecomment-505833092。

[5]https://plotly.com/r/logos/。

© 黄湘云

12.1 散点图

表 12.2: 散点图类型

类型	名称
scattercarpet	地毯图
scatterternary	三元图
scatter3d	三维散点图
scattergeo	地图散点图
scattermapbox	地图散点图 Mapbox
scatter	散点图
scattergl	散点图 GL
scatterpolar	极坐标散点图
scatterpolargl	极坐标散点图 GL

plotly.js 提供很多图层用于绘制各类图形 https://github.com/plotly/plotly.js/tree/master/src/traces

```
# 折线图
plot_ly(Orange,
  x = ~age, y = ~circumference, color = ~Tree,
  type = "scatter", mode = "markers"
)
```

12.2 条形图

日常使用最多的图形无外乎散点图、柱形图（分组、堆积、百分比堆积等）

```
# 简单条形图
library(data.table)
diamonds <- as.data.table(diamonds)

p11 <- diamonds[, .(cnt = .N), by = .(cut)] %>%
  plot_ly(x = ~cut, y = ~cnt, type = "bar") %>%
  add_text(
    text = ~ scales::comma(cnt), y = ~cnt,
    textposition = "top middle",
    cliponaxis = FALSE, showlegend = FALSE
  )
# 分组条形图
p12 <- plot_ly(diamonds,
  x = ~cut, color = ~clarity,
  colors = "Accent", type = "histogram"
)
# 堆积条形图
```

© 基础统计


```
p13 <- plot_ly(diamonds,
  x = ~cut, color = ~clarity,
  colors = "Accent", type = "histogram"
) %>%
  layout(barmode = "stack")
# 百分比堆积条形图
# p14 <- plot_ly(diamonds,
#   x = ~cut, color = ~clarity,
#   colors = "Accent", type = "histogram"
# ) %>%
#   layout(barmode = "stack", barnorm = "percent") %>%
#   config(displayModeBar = F)

# 推荐使用如下方式绘制堆积条形图
dat = diamonds[, .(cnt = length(carat)), by = .(clarity, cut)] %>%
  .[, pct := round(100 * cnt / sum(cnt), 2), by = .(cut)]

p14 <- plot_ly(
  data = dat, x = ~cut, y = ~pct, color = ~clarity,
  colors = "Set3", type = "bar"
) %>%
  layout(barmode = "stack")

htmltools::tagList(p11, p12, p13, p14)
```

12.3 折线图

其它常见的图形还要折线图、直方图、箱线图和提琴图

```
# 折线图
plot_ly(Orange,
  x = ~age, y = ~circumference, color = ~Tree,
  type = "scatter", mode = "markers+lines"
)
```

12.4 双轴图

双轴图

模拟一组数据

```
set.seed(2020)
dat <- data.frame(
  dt = seq(from = as.Date("2020-01-01"), to = as.Date("2020-01-31"), by = "day"),
```

© 基础设施

```
  search_qv = sample(100000:1000000, size = 31, replace = T)
) %>%
  transform(valid_click_qv = sapply(search_qv, rbinom, n = 1, prob = 0.5)) %>%
  transform(qv_ctr = valid_click_qv / search_qv)
```

hoverinfo = "text" 表示 tooltips 使用指定的 text 映射，而 visible = "legendonly" 表示图层默认隐藏不展示，只在图例里显示，有时候很多条线，默认只是展示几条而已。举例如下

```
plot_ly(data = dat) %>%
  add_bars(
    x = ~dt, y = ~search_qv, color = I("gray80"), name = "搜索 QV",
    text = ~ paste0(
      "日期: ", dt, "<br>",
      "点击 QV: ", format(valid_click_qv, big.mark = ","), "<br>",
      "搜索 QV: ", format(search_qv, big.mark = ","), "<br>",
      "QV_CTR: ", scales::percent(qv_ctr, accuracy = 0.01), "<br>"
    ),
    hoverinfo = "text"
  ) %>%
  add_bars(
    x = ~dt, y = ~valid_click_qv, color = I("gray60"), name = "点击 QV",
    text = ~ paste0(
      "日期: ", dt, "<br>",
      "点击 QV: ", format(valid_click_qv, big.mark = ","), "<br>",
      "搜索 QV: ", format(search_qv, big.mark = ","), "<br>",
      "QV_CTR: ", scales::percent(qv_ctr, accuracy = 0.01), "<br>"
    ), visible = "legendonly",
    hoverinfo = "text"
  ) %>%
  add_lines(
    x = ~dt, y = ~qv_ctr, name = "QV_CTR", yaxis = "y2", color = I("gray40"),
    text = ~ paste("QV_CTR: ", scales::percent(qv_ctr, accuracy = 0.01), "<br>"),
    hoverinfo = "text",
    line = list(shape = "spline", width = 3, dash = "line")
  ) %>%
  layout(
    title = "",
    yaxis2 = list(
      tickfont = list(color = "black"),
      overlaying = "y",
      side = "right",
      title = "QV_CTR (%) ",
      # ticksuffix = "%", # 设置坐标轴单位
      tickformat = '.1%', # 设置坐标轴刻度
      showgrid = F, automargin = TRUE
    ),
```

© 基础统计


```
    xaxis = list(title = "日期", showgrid = F, showline = F),
    yaxis = list(title = " ", showgrid = F, showline = F),
    margin = list(r = 20, autoexpand = T),
    legend = list(
      x = 0, y = 1, orientation = "h",
      title = list(text = " ")
    )
  )
```

12.5 直方图

```
plot_ly(iris,
  x = ~Sepal.Length, colors = "Greys",
  color = ~Species, type = "histogram"
)
```

12.6 箱线图

```
# 箱线图
plot_ly(diamonds,
  x = ~clarity, y = ~price, colors = "Greys",
  color = ~clarity, type = "box"
)
```

12.7 提琴图

```
plot_ly(sleep,
  x = ~group, y = ~extra, split = ~group,
  type = "violin",
  box = list(visible = T),
  meanline = list(visible = T)
)
```

plotly 包含图层 27 种，见表 12.3

12.8 气泡图

简单图形 scatter，分布图几类，其中 scatter、heatmap、scatterpolar 支持 WebGL 绘图引擎

© 黄湘云

表 12.3: 图层

A	B	C
add_annotations	add_histogram	add_polygons
add_area	add_histogram2d	add_ribbons
add_bars	add_histogram2dcontour	add_scattergeo
add_boxplot	add_image	add_segments
add_choropleth	add_lines	add_sf
add_contour	add_markers	add_surface
add_data	add_mesh	add_table
add_fun	add_paths	add_text
add_heatmap	add_pie	add_trace

```
# https://plotly.com/r/bubble-charts/
dat <- diamonds[, .(
  carat = mean(carat),
  price = sum(price),
  cnt = .N
), by = .(cut)]

plot_ly(
  data = dat, colors = "Greys",
  x = ~carat, y = ~price, color = ~cut, size = ~cnt,
  type = "scatter", mode = "markers",
  marker = list(
    symbol = "circle", sizemode = "diameter",
    line = list(width = 2, color = "#FFFFFF"), opacity = 0.4
  ),
  text = ~ paste(
    sep = " ", "重量: ", round(carat, 2), "克拉",
    "<br>价格:", round(price / 10^6, 2), "百万"
  ),
  hoverinfo = 'text'
) %>%
  add_annotations(
    x = ~carat, y = ~price, text = ~cnt,
    showarrow = F, font = list(family = "sans")
  ) %>%
  layout(
    xaxis = list(hoverformat = ".2f"),
    yaxis = list(hoverformat = ".0f")
  )
```

© 基础统计

12.9 曲线图

```
plot_ly(
  x = c(1, 2.2, 3), y = c(5.3, 6, 7),
  type = "scatter", color = I("gray40"),
  mode = "markers+lines", line = list(shape = "spline")
) %>%
  add_annotations(
    x = 2, y = 6, size = I(100),
    text = TeX("x_i \\sim N(\\mu, \\sigma)")
  ) %>%
  layout(
    xaxis = list(showgrid = F, title = TeX("\\mu")),
    yaxis = list(showgrid = F, title = TeX("\\alpha"))
  ) %>%
  config(mathjax = 'cdn')
```

12.10 堆积图

```
plot_ly(
  data = PlantGrowth, y = ~weight,
  color = ~group, colors = "Greys",
  type = "scatter", line = list(shape = "spline"),
  mode = "lines", fill = "tozeroy"
)
```

12.11 热力图

其他基础图形

```
plot_ly(z = volcano, type = 'heatmap', colors = "Greys")
```

12.12 地图 I

`plot_mapbox()` 使用 Mapbox 提供的地图服务，因此，需要注册一个账户，获取 MAPBOX_TOKEN

```
data("quakes")
plot_mapbox(
  data = quakes, colors = "Greys",
  lon = ~long, lat = ~lat,
  color = ~mag, size = 2,
  type = "scattermapbox",
```

© 基础统计

```
  mode = "markers",
  marker = list(opacity = 0.5)
) %>%
  layout(
    title = "Fiji Earthquake",
    mapbox = list(
      zoom = 3,
      center = list(
        lat = ~ median(lat - 5),
        lon = ~ median(long)
      )
    )
  ) %>%
  config(
    mapboxAccessToken = Sys.getenv("MAPBOX_TOKEN")
  )
```

```
plotly::plot_ly(
  data = quakes,
  lon = ~long, lat = ~lat,
  type = "scattergeo", mode = "markers",
  text = ~ paste0(
    "站点: ", stations, "<br>",
    "震级: ", mag
  ),
  marker = list(
    color = ~mag,
    size = 10, opacity = 0.8,
    line = list(color = "white", width = 1)
  )
) %>%
  plotly::layout(geo = list(
    showland = TRUE,
    landcolor = plotly::toRGB("gray95"),
    subunitcolor = plotly::toRGB("gray85"),
    countrycolor = plotly::toRGB("gray85"),
    countrywidth = 0.5,
    subunitwidth = 0.5,
    lonaxis = list(
      showgrid = TRUE,
      gridwidth = 0.5,
      range = c(160, 190),
      dtick = 5
    ),
    lataxis = list(
```

© 王建平


```
      showgrid = TRUE,
      gridwidth = 0.5,
      range = c(-40, -10),
      dtick = 5
    )
  ))
```

```
dat <- data.frame(state.x77,
  stats = rownames(state.x77),
  stats_abbr = state.abb
)

plotly::plot_ly(
  data = dat,
  type = "choropleth",
  locations = ~stats_abbr,
  locationmode = "USA-states",
  colorscale = "Viridis",
  z = ~Income
) |>
  plotly::layout(
    geo = list(scope = "usa"),
    title = "1974年美国各州的人均收入",
    legend = list(title = "收入")
  ) |>
  plotly::config(displayModeBar = FALSE)
```

12.13 拟合图

```
plot_ly(economics,
  type = "scatter",
  x = ~date,
  y = ~uempmed,
  name = "observed unemployment",
  mode = "markers+lines",
  marker = list(
    color = "red"
  ),
  line = list(
    color = "red",
    dash = "dashed"
  )
) %>%
  add_trace(
```

© 黄湘云

```
    x = ~date,
    y = ~fitted(loess(uempmed ~ as.numeric(date))),
    name = "fitted unemployment",
    mode = "markers+lines",
    marker = list(
      color = "orange"
    ),
    line = list(
      color = "orange"
    )
  ) %>%
  layout(
    title = "失业时间",
    xaxis = list(
      title = "日期",
      showgrid = F
    ),
    yaxis = list(
      title = "失业时间（周）"
    ),
    legend = list(
      x = 0, y = 1, orientation = "v",
      title = list(text = "")
    )
  )
```

12.14 轨迹图

rasterly 百万量级的散点图

```
library(rasterly)
plot_ly(quakes, x = ~long, y = ~lat) %>%
  add_rasterly_heatmap()

quakes %>%
  rasterly(mapping = aes(x = long, y = lat)) %>%
  rasterly_points()
```

```
library(plotly)
# 读取数据
# uber 轨迹数据来自 https://github.com/plotly/rasterly
ridesDf <- readRDS(file = 'data/uber.rds')

ridesDf %>%
  rasterly(mapping = aes(x = Lat, y = Lon)) %>%
```

© 黄湘云


```
  rasterly_points()
```

图 12.1: 轨迹数据

12.15 三维图 (plotly)

```
plot_ly(z = ~volcano) %>%
  add_surface()

plot_ly(x = c(0, 0, 1), y = c(0, 1, 0), z = c(0, 0, 0)) %>%
  add_mesh()

# https://plot.ly/r/reference/#scatter3d
transform(mtcars, am = ifelse(am == 0, "Automatic", "Manual")) %>%
  plot_ly(x = ~wt, y = ~hp, z = ~qsec,
          color = ~am, colors = c("#BF382A", "#0C4B8E")) %>%
  add_markers() %>%
  layout(scene = list(
    xaxis = list(title = "Weight"),
    yaxis = list(title = "Gross horsepower"),
    zaxis = list(title = "1/4 mile time")
```

© 清华大学出版社

```
  ))
```

12.16 甘特图

项目管理必备，如图所示，本项目拆分成 7 个任务，一共使用 3 种项目资源

```
# https://plotly.com/r/gantt/
# 项目拆解为一系列任务，每个任务的开始时间，持续时间和资源类型
df <- data.frame(
  task = paste("Task", 1:8),
  start = as.Date(c(
    "2016-01-01", "2016-02-20", "2016-01-01",
    "2016-04-10", "2016-06-09", "2016-04-10",
    "2016-09-07", "2016-11-26"
  )),
  duration = c(50, 25, 100, 60, 30, 150, 80, 10),
  resource = c("A", "B", "C", "C", "C", "A", "B", "B")
) %>%
  transform(end = start + duration) %>%
  transform(y = 1:nrow(.))

plot_ly(data = df) %>%
  add_segments(
    x = ~start, xend = ~end,
    y = ~y, yend = ~y,
    color = ~resource,
    mode = "lines",
    colors = "Greys",
    line = list(width = 20),
    showlegend = F,
    hoverinfo = "text",
    text = ~ paste(
      " 任务: ", task, "<br>",
      "启动时间: ", start, "<br>",
      "周期: ", duration, "天<br>",
      "资源: ", resource
    )
  ) %>%
  layout(
    xaxis = list(
      showgrid = F,
      title = list(text = "")
    ),
    yaxis = list(
```

© 黄湘云


```
      showgrid = F,
      title = list(text = ""),
      tickmode = "array",
      tickvals = 1:nrow(df),
      ticktext = unique(df$task),
      domain = c(0, 0.9)
    ),
    annotations = list(
      list(
        xref = "paper", yref = "paper",
        x = 0.80, y = 0.1,
        text = paste0(
          "项目周期: ", sum(df$duration), " 天<br>",
          "资源类型: ", length(unique(df$resource)), " 个<br>"
        ),
        font = list(size = 12),
        ax = 0, ay = 0,
        align = "left"
      ),
      list(
        xref = "paper", yref = "paper",
        x = 0.1, y = 1,
        xanchor = "left",
        text = "项目资源管理",
        font = list(size = 20),
        ax = 0, ay = 0,
        align = "left",
        showarrow = FALSE
      )
    )
  )
```

12.17 帕雷托图

帕雷托图 20/80 法则

```
# 数据来自 https://github.com/plotly/datasets
dat <- data.frame(
  complaint = c(
    "Small portions", "Overpriced",
    "Wait time", "Food is tasteless", "No atmosphere", "Not clean",
    "Too noisy", "Food is too salty", "Unfriendly staff", "Food not fresh"
  ),
  count = c( 621L, 789L, 109L, 65L, 45L, 30L, 27L, 15L, 12L, 9L)
```

© 基础统计

```
)

dat <- dat[order(-dat$count), ] %>%
  transform(cumulative = round(100 * cumsum(count) / sum(count), digits = 2))

# complaint 按 count 降序排列
dat$complaint <- reorder(x = dat$complaint, X = dat$count, FUN = function(x) 1/(1 + x))

plot_ly(data = dat) %>%
  add_bars(
    x = ~complaint, y = ~count,
    showlegend = F, color = I("gray60")
  ) %>%
  add_lines(
    x = ~complaint, y = ~cumulative, yaxis = "y2",
    showlegend = F, color = I("gray40")
  ) %>%
  layout(
    yaxis2 = list(
      tickfont = list(color = "black"),
      overlaying = "y",
      side = "right",
      title = "累积百分比 (%) ",
      showgrid = F
    ),
    xaxis = list(title = "投诉类型", showgrid = F, showline = F),
    yaxis = list(title = "数量", showgrid = F, showline = F)
  )
```

提示

reorder() 对 complaint 按照降序还是升序由 FUN 函数的单调性决定，单调增对应升序，单调减对应降序

12.18 时间线

```
library(vistime)

pres <- data.frame(
  Position = rep(c("President", "Vice"), each = 3),
  Name = c("Washington", rep(c("Adams", "Jefferson"), 2), "Burr"),
  start = c("1789-03-29", "1797-02-03", "1801-02-03"),
  end = c("1797-02-03", "1801-02-03", "1809-02-03"),
  color = c("#cbb69d", "#603913", "#c69c6e"),
```

© 基础


```
  fontcolor = c("black", "white", "black")
)

vistime(pres, col.event = "Position", col.group = "Name")
```

12.19 漏斗图

```
dat <- data.frame(
  category = c("访问", "下载", "潜客", "报价", "下单"),
  value = c(39, 27.4, 20.6, 11, 2)
) %>%
  transform(percent = value / cumsum(value))

plot_ly(data = dat) %>%
  add_trace(
    type = "funnel",
    y = ~category,
    x = ~value,
    color = ~category,
    colors = "Set2",
    text = ~ paste0(value, "<br>", sprintf("%.2f%%", 100*percent)) ,
    hoverinfo = "text",
    showlegend = FALSE
  ) %>%
  layout(yaxis = list(
    categoryarray = ~category,
    title = ""
  ))
```

```
plotly::plot_ly(data = dat) %>%
  plotly::add_trace(
    type = "funnel",
    y = ~category,
    x = ~value,
    marker = list(color = RColorBrewer::brewer.pal(n = 5, name = "Set2")),
    textposition = "auto",
    textinfo = "value+percent previous",
    hoverinfo = "none"
  ) %>%
  plotly::layout(yaxis = list(categoryarray = ~category, title = ""))
```

© 基础统计

12.20 雷达图

```
plot_ly(
  type = "scatterpolar", mode = "markers", fill = "toself"
) %>%
  add_trace(
    r = c(39, 28, 8, 7, 28, 39), color = I("gray40"),
    theta = c("数学", "物理", "化学", "英语", "生物", "数学"),
    name = "学生 A"
  ) %>%
  add_trace(
    r = c(1.5, 10, 39, 31, 15, 1.5), color = I("gray80"),
    theta = c("数学", "物理", "化学", "英语", "生物", "数学"),
    name = "学生 B"
  ) %>%
  layout(
    polar = list(
      radialaxis = list(
        visible = T,
        range = c(0, 50)
      )
    )
  )
```

12.21 瀑布图

盈亏图

```
library(plotly)
library(dplyr)

dat <- data.frame(
  x = c(
    "销售", "咨询", "净收入",
    "购买", "其他费用", "税前利润"
  ),
  y = c(60, 80, 10, -40, -20, 0),
  measure = c(
    "relative", "relative", "relative",
    "relative", "relative", "total"
  )
) %>%
  mutate(text = case_when(
    y > 0 ~ paste0("+", y),
```

© 王建硕


```
    y == 0 ~ "",
    y < 0 ~ as.character(y)
  )) %>%
  mutate(x = factor(x, levels = c(
    "销售", "咨询", "净收入",
    "购买", "其他费用", "税前利润"
  )))

n_rows <- nrow(dat)
dat[nrow(dat), "text"] <- "累计"

# measure 取值为 'relative'/'total'/'absolute'
plotly::plot_ly(dat,
  x = ~x, y = ~y, measure = ~measure, type = "waterfall",
  text = ~text, textposition = "outside",
  name = "收支", hoverinfo = "final",
  connector = list(line = list(color = "gray")),
  increasing = list(marker = list(color = "#66C2A5")),
  decreasing = list(marker = list(color = "#FC8D62")),
  totals = list(marker = list(color = "#8DA0CB"))
) %>%
  plotly::layout(
    title = "2018 年收支状态",
    xaxis = list(title = "业务"),
    yaxis = list(title = "金额"),
    showlegend = FALSE
  )
```

12.22 树状图

plotly 绘制 treemap 和 sunburst 图比较复杂，接口不友好，plotme 正好弥补不足。

12.23 旭日图

plotme

12.24 调色板

```
plot_ly(iris,
  x = ~Petal.Length, y = ~Petal.Width,
  mode = "markers", type = "scatter",
```

© 王建硕

```
  color = ~ Sepal.Length > 6, colors = c("#132B43", "#56B1F7")
)
plot_ly(iris,
  x = ~Petal.Length, y = ~Petal.Width, color = ~ Sepal.Length > 6,
  mode = "markers", type = "scatter"
)

plot_ly(iris,
  x = ~Petal.Length, y = ~Petal.Width, color = ~ Sepal.Length > 6,
  mode = "markers", type = "scatter", colors = "Set2"
)

plot_ly(iris,
  x = ~Petal.Length, y = ~Petal.Width, color = ~ Sepal.Length > 6,
  mode = "markers", type = "scatter", colors = "Set1"
)
```

构造 20 个类别超出 Set1 调色板的范围，会触发警告说 Set1 没有那么多色块，但还是返回足够多的色块，也可以使用 `viridis`、`plasma`、`magma` 或 `inferno` 调色板

```
dat <- data.frame(
  dt = rep(seq(
    from = as.Date("2021-01-01"),
    to = as.Date("2021-01-31"), by = "day"
  ), each = 20),
  bu = rep(LETTERS[1:20], 31),
  qv = rbinom(n = 20 * 31, size = 10000, prob = runif(20 * 31))
)
# viridis
plot_ly(dat,
  x = ~dt, y = ~qv, color = ~bu,
  mode = "markers", type = "scatter", colors = "viridis"
)
```

12.25 时序图

dygraphs 专门用来绘制交互式时间序列图形，下面以美团股价为例，展示时间窗口筛选、坐标轴名称、刻度标签、注释、事件标注、缩放等功能

```
meituan <- quantmod::getSymbols("3690.HK", auto.assign = FALSE, src = "yahoo")
library(dygraphs)
# 缩放
dyUnzoom <- function(dygraph) {
  dyPlugin(
    dygraph = dygraph,
```

© 王沛勇


```
    name = "Unzoom",
    path = system.file("plugins/unzoom.js", package = "dygraphs")
  )
}

# 年月
getYearMonth <- '
  function(d) {
    var monthNames = ["01", "02", "03", "04", "05", "06","07", "08", "09", "10", "11", "12"];
    date = new Date(d);
    return date.getFullYear() + "-" + monthNames[date.getMonth()];
  }'

dygraph(meituan[, "3690.HK.Adjusted"], main = "美团股价走势") |>
  dyRangeSelector(dateWindow = c(format(Sys.Date(), "%Y-01-01"), as.character(Sys.Date()))) |>
  dyAxis(name = "x", axisLabelFormatter = getYearMonth) |>
  dyAxis("y", valueRange = c(0, 500), label = "美团股价") |>
  dyEvent("2020-01-23", "武汉封城", labelLoc = "bottom") |>
  dyShading(from = "2020-01-23", to = "2020-04-08", color = "#FFE6E6") |>
  dyAnnotation("2020-01-23", text = "武汉封城", tooltip = "武汉封城", width = 60) |>
  dyAnnotation("2020-04-08", text = "武汉解封", tooltip = "武汉解封", width = 60) |>
  dyHighlight(highlightSeriesOpts = list(strokeWidth = 2)) |>
  dySeries(label = "调整股价") |>
  dyLegend(show = "follow", hideOnMouseOut = FALSE) |>
  dyOptions(fillGraph = TRUE, drawGrid = FALSE, gridLineColor = "lightblue") |>
  dyUnzoom()
```

12.26 导出静态图形

orca (Open-source Report Creator App) 软件针对 plotly.js 库渲染的图形具有很强的导出功能，安装 orca 后，`plotly::orca()` 函数可以将基于 htmlwidgets 的 plotly 图形对象导出为 PNG、PDF 和 SVG 等格式的高质量静态图片。

```
p <- plot_ly(x = 1:10, y = 1:10, color = 1:10)
orca(p, "plot.svg")
```

12.27 静态图形转交互图形

函数 `ggplotly()` 将 ggplot 对象转化为交互式 plotly 对象

```
gg <- ggplot(faithful, aes(x = eruptions, y = waiting)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon") +
  xlim(1, 6) +
  ylim(40, 100)
```

© 王建平

静态图形

```
gg
```

转化为 plotly 对象

```
ggplotly(gg)
```

添加动态点的注释，比如点横纵坐标、坐标文本，整个注释标签的样式（如背景色）

```
ggplotly(gg, dynamicTicks = "y") %>%
  style(., hoveron = "points", hoverinfo = "x+y+text",
        hoverlabel = list(bgcolor = "white"))
```

12.28 地图 II

leaflet 包制作地图，斐济是太平洋上的一个岛国，处于板块交界处，经常发生地震，如下图所示，展示 1964 年来 1000 次震级大于 4 级的地震活动。

```
library(leaflet)
data(quakes)
# Pop 提示
quakes$popup_text <- lapply(paste(
  "编号:", "<strong>", quakes$stations, "</strong>", "<br>",
```

© 谢湘


```
  "震深:", quakes$depth, "<br>",
  "震级:", quakes$mag
), htmltools::HTML)
# 构造调色板
pal <- colorBin("Spectral", bins = pretty(quakes$mag), reverse = TRUE)
p <- leaflet(quakes) |>
  addProviderTiles(providers$CartoDB.Positron) |>
  addCircles(lng = ~long, lat = ~lat, color = ~ pal(mag), label = ~popup_text) |>
  addLegend("bottomright",
    pal = pal, values = ~mag,
    title = "地震震级"
  ) |>
  addScaleBar(position = c("bottomleft"))
p
```

图 12.2: 斐济地震带

将上面的绘图部分保存为独立的 HTML 网页文件

```
library(htmlwidgets)
# p 就是绘图部分的数据对象
saveWidget(p, "fiji-map.html", selfcontained = T)
```

```
library(leaflet)
library(leaflet.extras)
```

© 基础图

```
quakes |>
  leaflet() |>
  addTiles() |>
  addProviderTiles(providers$OpenStreetMap.DE) |>
  addHeatmap(
    lng = ~long, lat = ~lat, intensity = ~mag,
    max = 100, radius = 20, blur = 10
  )
```

图 12.3: 斐济地震带热力图

leafletCN 提供汉化

```
# 地图默认放大倍数
zoom          <- 4
# 地图可以放大的倍数区间
minZoom       <- 1
maxZoom       <- 18

library(leaflet)
library(leafletCN)
library(maptools)
library(leaflet.extras)
```

仅供读者 ©


```
# 热力图 heatmap
leaflet(res, options = leafletOptions(minZoom = minZoom, maxZoom = maxZoom)) |>
  amap() |>
  # setView(lng = mean(data$long), lat = mean(data$lat), zoom = zoom) |>
  setView(lng = 109, lat = 38, zoom = 4) |>
  addHeatmap(
    lng = ~long2, lat = ~lat2, intensity = ~uv, max = max(res$uv),
    blur = blur, minOpacity = minOpacity, radius = radius
  )

quakes$popup_text <- lapply(paste(
  "编号:", "<strong>", quakes$stations, "</strong>", "<br>",
  "震深:", quakes$depth, "<br>",
  "震级:", quakes$mag
), htmltools::HTML)
# 构造调色板
pal <- colorBin("Spectral", bins = pretty(quakes$mag), reverse = TRUE)

leaflet(quakes) |>
  addProviderTiles(providers$CartoDB.Positron) |>
  addCircles(
    lng = ~long, lat = ~lat,
    color = ~ pal(mag), label = ~popup_text
  ) |>
  setView(178, -20, 5) |>
  addHeatmap(
    lng = ~long, lat = ~lat, intensity = ~mag,
    blur = 20, max = 0.05, radius = 15
  ) |>
  addLegend("bottomright",
    pal = pal, values = ~mag,
    title = "地震震级"
  ) |>
  addScaleBar(position = c("bottomleft"))
```

12.29 动画

袁凡 用 R 中 echarts4r 包绘制柱状图的笔记

```
# https://d.cosx.org/d/422311
library(purrr)
library(echarts4r)

data("gapminder", package = "gapminder")
```

© 王沛扬

```
titles <- map(unique(gapminder$year), function(x) {
  list(
    text = "Gapminder",
    left = "center"
  )
})

years <- map(unique(gapminder$year), function(x) {
  list(
    subtext = x,
    left = "center",
    top = "center",
    z = 0,
    subtextStyle = list(
      fontSize = 100,
      color = "rgb(170, 170, 170, 0.5)",
      fontWeight = "bolder"
    )
  )
})

# 添加一列颜色，各大洲和颜色的对应关系可自定义，调整 levels 或 labels 里面的顺序即可，也可不指定 levels ，
gapminder <- gapminder |>
  transform(
    color = factor(
      continent,
      levels = c("Asia", "Africa", "Americas", "Europe", "Oceania"),
      labels = RColorBrewer::brewer.pal(n = 5, name = "Spectral")
    )
  )

gapminder |>
  group_by(year) |>
  e_charts(x = gdpPercap, timeline = TRUE) |>
  e_scatter(
    serie = lifeExp, size = pop, bind = country,
    symbol_size = 5, name = ""
  ) |>
  e_add("itemStyle", color) |>
  e_y_axis(
    min = 20, max = 85, nameGap = 30,
    name = "Life Exp", nameLocation = "center"
  ) |>
  e_x_axis(
```

© 王建硕


```
    type = "log", min = 100, max = 100000,
    nameGap = 30, name = "GDP / Cap", nameLocation = "center"
  ) |>
  e_timeline_serie(title = titles) |>
  e_timeline_serie(title = years, index = 2) |>
  e_timeline_opts(playInterval = 1000) |>
  e_grid(bottom = 100) |>
  e_tooltip()
```

```
# params.name 对应 bind
# params.value[0] 对应 x
# params.value[1] 对应 serie
# params.value[2] 对应 size
# tooltips 自定义
# https://stackoverflow.com/questions/50554304/displaying-extra-variables-in-tooltips-echarts4r
# 百分数处理
# https://stackoverflow.com/questions/11832914/how-to-round-to-at-most-2-decimal-places-if-necessary
mtcars |>
  tibble::rownames_to_column("model") |>
  e_charts(x = wt) |>
  e_scatter(serie = mpg, size = qsec, bind = model) |>
  e_tooltip(formatter = htmlwidgets::JS("
          function(params) {
              return (
                  '<strong>' + params.name + '</strong>' +
                  '<br />wt: ' + params.value[0] +
                  '<br />mpg: ' + params.value[1] +
                  '<br />qsec- ' + params.value[2]
              )
          }
          "))
```

12.30 三维图 (rgl)

ggrgl

```
library(rgl)
lat <- matrix(seq(90, -90, len = 50) * pi / 180, 50, 50, byrow = TRUE)
long <- matrix(seq(-180, 180, len = 50) * pi / 180, 50, 50)

r <- 6378.1 # radius of Earth in km
x <- r * cos(lat) * cos(long)
y <- r * cos(lat) * sin(long)
z <- r * sin(lat)
```

© 清华大学出版社

```
# 调整视角
rgl.viewpoint( theta = 0, phi = 15, fov = 60, zoom = 0.5, interactive = TRUE)

persp3d(x, y, z,
  col = "white", xlab = "", ylab = "", zlab = "",
  texture = system.file("textures/world.png", package = "rgl"),
  specular = "black", axes = FALSE, box = FALSE,
  normal_x = x, normal_y = y, normal_z = z
)
```

12.31 网络图

gephi 探索和可视化网络图 GraphViz

```
# library(igraph)
```

12.31.1 networkD3

networkD3 D3 非常适合绘制网络图，如网络、树状、桑基图

```
library(networkD3)
data(MisLinks, MisNodes) # 加载数据
head(MisLinks) # 边
```

```
##   source target value
## 1      1      0     1
## 2      2      0     8
## 3      3      0    10
## 4      3      2     6
## 5      4      0     1
## 6      5      0     1
```

```
head(MisNodes) # 节点
```

```
##              name group size
## 1          Myriel     1   15
## 2        Napoleon     1   20
## 3 Mlle.Baptistine     1   23
## 4    Mme.Magloire     1   30
## 5    CountessdeLo     1   11
## 6        Geborand     1    9
```

构造网络图

```
forceNetwork(
  Links = MisLinks, Nodes = MisNodes, Source = "source",
  Target = "target", Value = "value", NodeID = "name",
```

© 草稿


```
  Group = "group", opacity = 0.4
)
```

12.31.2 visNetwork

visNetwork 使用 vis-network.js 库绘制网络关系图 https://datastorm-open.github.io/visNetwork

```
library(visNetwork)
```

调用函数 `visTree()` 可视化分类模型结果

```
library(rpart)
library(sparkline) # 函数 visTree 需要导入 sparkline 包
res <- rpart(Species~., data=iris)
visTree(res, main = "鸢尾花分类树", width = "100%")
```

节点、边的属性都可以映射数据指标

12.31.3 r2d3

D3 是非常流行的 JavaScript 库，r2d3 提供了 R 接口

```
library(r2d3)
```

更加具体的使用介绍，一个复杂的案例，如何从简单配置过来，以条形图为例，D3 是一个相当强大且成熟的库，提供的案例功能要覆盖 plotly

此为试读，需要完整PDF请访问：www.ertongbook.com

r2d3 提供了两个样例 JS 库 baranims.js 和 barchart.js

```
list.files(system.file("examples/", package = "r2d3"))
```

```
## [1] "baranims.js" "barchart.js"
```

```
library(r2d3)
r2d3(
  data = c(0.3, 0.6, 0.8, 0.95, 0.40, 0.20),
  script = system.file("examples/barchart.js", package = "r2d3")
)
```

```
r2d3(
  data = c(0.3, 0.6, 0.8, 0.95, 0.40, 0.20),
  script = system.file("examples/baranims.js", package = "r2d3")
)
```

TODO: 提供一个 R 包和 HTML Widgets 小练习：给 roughViz.js 写个 R 包装 https://d.cosx.org/d/421030-r-html-widgets-roughviz-js-r https://github.com/XiangyunHuang/roughviz

12.32 Python 交互图形

Plotly 的图形库

```
import plotly.express as px

px.scatter(
    px.data.iris(),
    x="sepal_width",
    y="sepal_length",
    color="species",
    trendline="ols",
    template="simple_white",
    labels={
        "sepal_length": "Sepal Length (cm)",
        "sepal_width": "Sepal Width (cm)",
        "species": "Species of Iris",
    },
    title="Edgar Anderson's Iris Data",
    color_discrete_sequence=px.colors.qualitative.Set2
)
```

不能同时使用 Python 版和 R 版的 Plotly.js 库，因为版本不一致产生冲突，而不能显示图形。

©黄海广

图 12.4: 插入图片

12.33 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] sparkline_2.0    rpart_4.1.16     visNetwork_2.1.0 networkD3_0.4
## [5] r2d3_0.2.6       dygraphs_1.1.1.6 plotly_4.10.0    ggplot2_3.3.5
```

© 黄湘云

```
## [9] reticulate_1.24
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8.3      lattice_0.20-45   tidyr_1.2.0       ps_1.6.0
##  [5] png_0.1-7         sysfonts_0.8.8    zoo_1.8-9         assertthat_0.2.1
##  [9] digest_0.6.29     utf8_1.2.2        R6_2.5.1          evaluate_0.15
## [13] httr_1.4.2        pillar_1.7.0      rlang_1.0.2       lazyeval_0.2.2
## [17] curl_4.3.2        rstudioapi_0.13   data.table_1.14.2 callr_3.7.0
## [21] Matrix_1.4-1      rmarkdown_2.13    labeling_0.4.2    webshot_0.5.2
## [25] stringr_1.4.0     htmlwidgets_1.5.4 igraph_1.2.11     munsell_0.5.0
## [29] compiler_4.1.3    xfun_0.30         pkgconfig_2.0.3   htmltools_0.5.2
## [33] tidyselect_1.1.2  tibble_3.1.6      bookdown_0.25     fansi_1.0.3
## [37] viridisLite_0.4.0 crayon_1.5.1      dplyr_1.0.8       withr_2.5.0
## [41] MASS_7.3-56       grid_4.1.3        jsonlite_1.8.0    gtable_0.3.0
## [45] lifecycle_1.0.1   DBI_1.1.2         magrittr_2.0.3    scales_1.1.1
## [49] cli_3.2.0         stringi_1.7.6     farver_2.1.0      ellipsis_0.3.2
## [53] generics_0.1.2    vctrs_0.4.0       tools_4.1.3       glue_1.6.2
## [57] purrr_0.3.4       processx_3.5.3    fastmap_1.1.0     yaml_2.3.5
## [61] colorspace_2.0-3  isoband_0.2.5     knitr_1.38
```

© 黄湘云

第三部分

动态文档

© 黄湘云

介绍

图 12.5: R Markdown 极其周边生态

WrapRmd 将 R Markdown 里很长的文本自动断行，但不产生空行。regexplain 帮助检查正则表达式，rdoc 支持 R 帮助文档的语法高亮。shinyComponents 实现在 R Markdown 中写 shiny 。wordcountaddin 统计 R Markdown 文档中的单词数量。styler 格式化 R Markdown 文档中的代码块。reprex 添加代码执行的软件环境，提供可重复的例子，方便在论坛/Github 上发问。carbonate 将源代码截图。downloadthis 在 R Markdown 文档中添加下载按钮。icon 添加各种各样的图标，thematic 定制 R Markdown 主题。datadrivencv、vitae 制作基于 R Markdown 文档的简历。addinslist 收集了一系列 RStudio 插件，提高写作和编码的效率。posterdown 写宣传海报，redoc 实现 R Markdown 和 Microsoft Word 两种文档格式之间互相转化，rrtools 写可重复性的研究论文和报告，提供一套自动化的软件环境的配置，节省科研人员的时间。butteRfly 快速获取 Github 等社交网络上活动记录，以日历图的形式展现出来。flow 可以非常方便地制作函数内部调用执行的流程图。

minidown 提供轻量级的 CSS 框架打磨的网页模版，rmdformats 和 prettydoc 提供不同主题样式的网页输出，govdown 提供 GOV.UK 风格的网页模版。

uiucthemes 伊利诺伊大学主题的 R Markdown 模版，rmdshower 提供 shower 引擎打造的幻灯片，而 xaringan 是基于 remark.js。xaringanthemer 和 xaringanExtra 包含丰富的 xaringan 的主题。

slidex 可以将 PowerPoint 幻灯片转化为粗燥的 xaringan 幻灯片。

gluedown 用 R 代码写格式化的 Markdown 文本，

- Reproducible Research Data and Project Management in R https://annakrystalli.me/rrresearchACCE20/

© 黄湘云

- Higher, further, faster with Marvelous R Markdown https://bit.ly/marvelRMD
- R Markdown for Scientists https://rmd4sci.njtierney.com/
- Getting Used to R, RStudio, and R Markdown https://rbasics.netlify.app/
- R Markdown 指南手册 https://www.dataquest.io/blog/r-markdown-guide-cheatsheet/
- Statistical Inference via Data Science: A ModernDive into R and the tidyverse https://moderndive.com/
- 参数化报告 https://github.com/jenniferthompson/ParamRmdExample 和 https://elastic-lovelace-155848.netlify.app/gallery/themes/flatly.html
- Sharing analyses with R Markdown https://andrewbtran.github.io/NICAR/2018/workflow/docs/02-rmarkdown.html
- Introduction to the Normal Distribution https://tinystats.github.io/teacups-giraffes-and-statistics/02_bellCurve.html
- 混合效应模型的 workshop https://github.com/singmann/mixed_model_workshop
- 基于 thematic 和 bslib 包美化 Rmd 文档 https://www.tillac-data.com/2020-fast-rmd-theming-with-thematic-and-bootstraplib/
- 借助 flipbookr 在 xaringan 制作的幻灯片里逐行展示代码执行的效果，特别适合用于 ggplot2 的教学 https://evamaerey.github.io/little_flipbooks_library/flipbookr/skeleton
- 制作 note/tips 等自定义块 https://desiree.rbind.io/post/2019/making-tip-boxes-with-bookdown-and-rmarkdown/
- learnr: Interactive Tutorials with R Markdown https://rstudio.github.com/learnr/
- r2d3: R Interface to D3 Visualizations https://rstudio.github.io/r2d3/
- radix: Radix combines the technical authoring features of Distill with R Markdown, enabling a fully reproducible workflow based on literate programming https://github.com/radixpub/radix-r
- revealjs: R Markdown Format for reveal.js Presentations https://github.com/rstudio/revealjs
- xaringan: Presentation Ninja 幻灯忍者写轮眼 https://slides.yihui.name/xaringan/

©黄湘云

第十三章　文档元素

```
library(knitr)
library(nomnoml)
library(magrittr)
library(rmarkdown)
```

knitr 将 R Markdown 文件转化为 Markdown 文件，Pandoc 可以将 Markdown 文件转化为 HTML5、Word、PowerPoint 和 PDF 等文档格式。

图 13.1: rmarkdown 支持的输出格式

rmarkdown 自 2014 年 09 月 17 日在 CRAN 上发布第一个正式版本以来，逐渐形成了一个强大的生态系统，世界各地的开发者贡献各种各样的扩展功能，见图 13.2

13.1　控制选项

Using SQL in RStudio

```
library(DBI)
conn <- DBI::dbConnect(RSQLite::SQLite(),
  dbname = system.file("db", "datasets.sqlite", package = "RSQLite")
)
```

Base R 内置的数据集都整合进 RSQLite 的样例数据库里了，

```
dbListTables(conn)
```

```
##  [1] "BOD"              "CO2"              "ChickWeight"      "DNase"
##  [5] "Formaldehyde"     "Indometh"         "InsectSprays"     "LifeCycleSavings"
##  [9] "Loblolly"         "Orange"           "OrchardSprays"    "PlantGrowth"
## [13] "Puromycin"        "Theoph"           "ToothGrowth"      "USArrests"
## [17] "USJudgeRatings"   "airquality"       "anscombe"         "attenu"
## [21] "attitude"         "cars"             "chickwts"         "esoph"
```

©黄湘云

图 13.2: rmarkdown 生态系统

图 13.3: R Markdown 概念图

```
## [25] "faithful"         "freeny"          "infert"          "iris"
## [29] "longley"          "morley"          "mtcars"          "npk"
## [33] "pressure"         "quakes"          "randu"           "rock"
## [37] "sleep"            "stackloss"       "swiss"           "trees"
## [41] "warpbreaks"       "women"
```

随意选择 5 行数据记录，将结果保存到变量 iris_preview

```
SELECT * FROM iris LIMIT 5;
```

查看变量 iris_preview 的内容

```
iris_preview
```

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

结束后关闭连接

```
dbDisconnect(conn = conn)
```

13.2 Markdown

Markdown 为核心，Pandoc's Markdown 和 R Markdown 仅介绍扩展的功能，三剑客 Markdown & Pandoc's Markdown & R Markdown，Markdown for scientific writing

首先介绍 Markdown 在强调、标题、列表、断行、链接、图片、引用、代码块、LaTeX 公式等使用方式，然后在 Markdown 的基础上介绍 Pandoc's Markdown 功能有加强的地方，R Markdown 在 Pandoc's Markdown 的基础上介绍功能有加强的地方。

Markdown 基础语法见 RStudio IDE 自带的 Markdown 手册：RStudio 顶部菜单栏 -> Help -> Markdown Quick Reference，这里主要介绍一下 Markdown 高级语法，特别是 Pandoc's Markdown，其实是 Pandoc 提供了很多对 Markdown 的扩展支持，下面介绍一下被 Pandoc 加强后的 Markdown 表格、图片和公式的使用

13.2.1 列表

- 有序的列表
  1. 第一条
  2. 第二条
- 无序的列表
  - 第一条
  - 第二条

© 吉祥出版

  - here is my first list item.
  - and my second.

- 嵌套的列表
  1. 有序
  2. Item 2
  3. Item 3
     - Item 3a
     - Item 3b
  - 无序
  - Item 2
    * Item 2a
    * Item 2b

定义型列表中包含代码

Term 1 Definition 1

Term 2 with *inline markup* Definition 2

```
{ some code, part of Definition 2 }
```

Third paragraph of definition 2.

定义类型的列表，紧凑形式

Term 1 Definition 1
Term 2 Definition 2a
Definition 2b

无序列表

- fruits
  - apples
    * macintosh
    * red delicious
  - pears
  - peaches
- vegetables
  - broccoli
  - chard

对应 LaTeX 列表环境里的有序环境，通篇计数

(1) My first example will be numbered (1).
(2) My second example will be numbered (2).

Explanation of examples.

(3) My third example will be numbered (3).

`(@)` 环境可以引用

(4) 这是一个好例子

内部稿 ©

正如 (4) 所指出的那样，

列表里包含代码块

- item one
- item two

```
{ my code block }
```

显示反引号 `

13.2.2 引用

注意在引用末尾空两格，出处另起一行，引用名人名言:

> It's always better to give than to receive.
>
> Trellis graphics are a bit like hash functions: you can be close to the target, but get a far-off result.[1]
>
> — Dieter Menne
>
> If you imagine that this pen is Trellis, then Lattice is not this pen.[2]
>
> — Paul Murrell
>
> You're overlooking something like line 800 of the documentation for xyplot. [...] It's probably in the R-FAQ as well, since my original feeling was that this behaviour was chosen in order to confuse people and see how many people read the FAQ... :) [3]
>
> — Barry Rowlingson

13.2.3 表格

插入表格很简单的，复杂的表格制作可以借助 R 包 knitr 提供的 kable 函数以及 kableExtra 包[4]，此外谢益辉的书籍 bookdown: Authoring Books and Technical Documents with R Markdown 中也有一节专门介绍表格 https://bookdown.org/yihui/bookdown/tables.html

`kable` 支持多个表格并排，

```
knitr::kable(
  list(
    head(iris[, 1:2], 3),
    head(mtcars[, 1:3], 5)
  ),
  caption = 'A Tale of Two Tables.', booktabs = TRUE
)
```

在表格中引入数学符号

[1] (about problems with creating a suitable lattice panel function) R-help (August 2008)

[2] (on the difference of Lattice (which eventually was called grid) and Trellis) DSC 2001, Wien (March 2001)

[3] (about the fact that lattice objects have to be print()ed) R-help (May 2005)

[4] https://xiangyunhuang.github.io/bookdown-kableExtra/

© 黄湘云

表 13.1: A Tale of Two Tables.

Sepal.Length	Sepal.Width
5.1	3.5
4.9	3.0
4.7	3.2

	mpg	cyl	disp
Mazda RX4	21.0	6	160
Mazda RX4 Wag	21.0	6	160
Datsun 710	22.8	4	108
Hornet 4 Drive	21.4	6	258
Hornet Sportabout	18.7	8	360

kableExtra 、broom 和 pixiedust 包实现表格样式的精细调整，如黄湘云制作的 样例

13.2.4 图片

利用 `knitr::include_graphics` 函数在代码块中插入图片是很简单的，如图13.4所示，图、表的标题很长或者需要插入脚注，可以使用 [文本引用][text-references]

```
## Warning in knitr::include_graphics(path = system.file("help/figures",
## "mai.png", : It is highly recommended to use relative paths for images. You had
## absolute paths: "/opt/R/4.1.3/lib/R/library/graphics/help/figures/mai.png"
```

图 13.4: (ref:footnote)

```
par(mar = c(4.1, 4.1, 0.5, 0.5))
plot(rnorm(10), xlab = "", ylab = "")
```

控制图片插入的宽度参考谢益辉的博客：CSS 的位置属性以及如何居中对齐超宽元素 https://yihui.name/cn/2018/05/css-position/

- One
- Two
- Three

© 谢益辉

图 13.5: 测试文本引用 (ref:text-references) 图表标题很长可使用 [文本引用][text-references] (ref:footnote) 表格标题里插入脚注，但是 ebooks 不支持这样插入脚注 [^longnote] [^longnote]: Here's one with multiple blocks. [text-references]: https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html#text-references

图 13.6: 还可以在列表环境中插入图片

根据代码动态生成图片，并插入文档中；外部图片插入文档中

```
plot(AirPassengers)
```

```
plot(pressure)
plot(AirPassengers)
```

```
plot(pressure)
plot(AirPassengers)
```

```
plot(pressure)
plot(AirPassengers)
plot(pressure)
plot(AirPassengers)
```

13.2.5 公式

行内公式一对美元符号 α 或者 $\alpha + \beta$，行间公式

$$\alpha$$

或者

$$\alpha + \beta$$

对公式编号，如公式 (13.1)

© 黄湘云

图 13.7: 时间序列图

$$L(\beta, \boldsymbol{\theta}) = f(y; \beta, \boldsymbol{\theta}) = \int_{\mathbb{R}^n} N(t; D\beta, \Sigma(\boldsymbol{\theta})) f(y|t) dt \tag{13.1}$$

多行公式分别编号，如公式(13.2) 和公式(13.3)

$$\log\{\frac{p_i}{1-p_i}\} = T_i = d(x_i)'\beta + S(x_i) + Z_i \tag{13.2}$$

$$\log(\lambda_i) = T_i = d(x_i)'\beta + S(x_i) + Z_i \tag{13.3}$$

多行公式中对某一（些）行编号，如公式 (13.5) 和公式 (13.6)

$$g(X_n) = g(\theta) + g'(\tilde{\theta})(X_n - \theta) \tag{13.4}$$

$$\sqrt{n}[g(X_n) - g(\theta)] = g'\left(\tilde{\theta}\right)\sqrt{n}[X_n - \theta] \tag{13.5}$$

$$\log(\lambda_i) = T_i = d(x_i)'\beta + S(x_i) + Z_i \tag{13.6}$$

多行公式共用一个编号，如公式 (13.7)

© 黄湘云

(a) 压力与温度的关系

(b) 时间序列图

图 13.8: 2 行 1 列布局

© 黄湘云

(a) 压力与温度的关系 (b) 时间序列图

图 13.9: 1 行 2 列布局

(a) 压力与温度的关系 (b) 压力与温度的关系

(c) 压力与温度的关系 (d) 压力与温度的关系

图 13.10: 2x2 图布局

© 黄湘云

$$
\begin{aligned}
L(\beta, \boldsymbol{\theta}) &= \int_{\mathbb{R}^n} \frac{N(t; D\beta, \Sigma(\boldsymbol{\theta}))f(y|t)}{N(t; D\beta_0, \Sigma(\boldsymbol{\theta}_0))f(y|t)} f(y,t)dt \\
&\propto \int_{\mathbb{R}^n} \frac{N(t; D\beta, \Sigma(\boldsymbol{\theta}))}{N(t; D\beta_0, \Sigma(\boldsymbol{\theta}_0))} f(t|y)dt \\
&= E_{T|y}\left[\frac{N(t; D\beta, \Sigma(\boldsymbol{\theta}))}{N(t; D\beta_0, \Sigma(\boldsymbol{\theta}_0))}\right]
\end{aligned} \tag{13.7}
$$

推荐在 `equation` 公式中，使用 `split` 环境，意思是一个公式很长，需要拆成多行，如公式(13.8)

$$
\begin{aligned}
\mathrm{Var}(\hat{\beta}) &= \mathrm{Var}((X'X)^{-1}X'y) \\
&= (X'X)^{-1}X'\mathrm{Var}(y)((X'X)^{-1}X')' \\
&= (X'X)^{-1}X'\mathrm{Var}(y)X(X'X)^{-1} \\
&= (X'X)^{-1}X'\sigma^2 IX(X'X)^{-1} \\
&= (X'X)^{-1}\sigma^2
\end{aligned} \tag{13.8}
$$

注意，`\mathbf` 只对字母 a, b, c, A, B, C 加粗，mathjax 不支持公式中使用 `\bm` 对 $\theta, \alpha, \beta, \ldots, \gamma$ 加粗，应该使用 `\boldsymbol`

13.3 表格

knitr 的 `kable()` 函数提供了制作表格的基本功能 https://bookdown.org/yihui/rmarkdown-cookbook/tables.html，flextable 支持更加细粒度的表格定制功能。beautifyR 整理 Markdown 表格非常方便，datapasta 快速复制粘贴 data.frame 和 tibble 类型的数据表格。rpivotTable 不更新了，pivottabler 在更新，内容似乎更好。remedy 提供了更加通用的 Markdown 写作功能，简化创作的技术难度。

13.4 流程图

nomnoml 流程图、思维导图

```
nomnoml::nomnoml("
#stroke: #26A63A
#.box: fill=#8f8 dashed visual=note
#direction: down

[Sweave-test-1.Rnw] -> utils::Sweave() [Sweave-test-1.tex|Sweave-test-1-006.pdf|Sweave-test-1-007.pdf]
[Sweave-test-1.Rnw] -> utils::Stangle() [Sweave-test-1.R]
[Sweave-test-1.tex] -> tools::texi2pdf() [Sweave-test-1.pdf]
[Sweave-test-1.tex] -> tools::texi2dvi() [Sweave-test-1.dvi]
")
```

© 黄湘云

表 13.2: knitr 支持的引擎

awk	bash	coffee
gawk	groovy	haskell
lein	mysql	node
octave	perl	psql
Rscript	ruby	sas
scala	sed	sh
stata	zsh	asis
asy	block	block2
bslib	c	cat
cc	comment	css
ditaa	dot	embed
exec	fortran	fortran95
go	highlight	js
julia	python	R
Rcpp	sass	scss
sql	stan	targets
tikz	verbatim	theorem
lemma	corollary	proposition
conjecture	definition	example
exercise	hypothesis	proof
remark	solution	nomnoml

13.5 编程语言引擎

语法高亮

Pandoc 通过 LaTeX 环境 lstlisting 支持语法高亮，比如

© 黄湘云

表 13.3: Pandoc 支持的语法高亮

ABAP	IDL	Plasm	ACSL	inform	POV	Ada
Java	Prolog	Algol	JVMIS	Promela	Ant	ksh
Python	Assembler	Lisp	R	Awk	Logo	Reduce
bash	make	Rexx	Basic	Mathematica	RSL	C
Matlab	Ruby	C++	Mercury	S	Caml	MetaPost
SAS	Clean	Miranda	Scilab	Cobol	Mizar	sh
Comal	ML	SHELXL	csh	Modula-2	Simula	Delphi
MuPAD	SQL	Eiffel	NASTRAN	tcl	Elan	Oberon-2
TeX	erlang	OCL	VBScript	Euphoria	Octave	Verilog
Fortran	Oz	VHDL	GCL	Pascal	VRML	Gnuplot
Perl	XML	Haskell	PHP	XSLT	HTML	PL/I

13.6 快速创建书籍项目

在指定目录创建 Book 项目

```
bookdown:::bookdown_skeleton("~/bookdown-demo")
```

项目根目录的文件列表

```
directory/
├── index.Rmd
├── 01-intro.Rmd
├── 02-literature.Rmd
├── 03-method.Rmd
├── 04-application.Rmd
├── 05-summary.Rmd
├── 06-references.Rmd
├── _bookdown.yml
├── _output.yml
├── book.bib
├── preamble.tex
├── README.md
└── style.css
```

13.7 Markdown 生态系统

大量基于 Markdown 的软件工具，比如 Wiki Gollum、Typora 和 VS Code 等

Pandoc's Markdown 在 Markdown 的基础上添加的功能

© 王沛沛

13.8 R Markdown 生态系统

R Markdown 站在巨人的肩膀上，这些巨人有 Markdown、Pandoc 和 LaTeX 等。

markdown 简洁设计哲学，Sweave 文学编程思想，期间各种工具粉墨登场，最后分别回到 Pandoc 和 R Markdown

表 13.4: R Markdown 生态系统

Package	Title
addinsOutline	RStudio Addins for Show Outline of a R Markdown/LaTeX Project
blogdown	Create Blogs and Websites with R Markdown
bookdown	Authoring Books and Technical Documents with R Markdown
bsplus	Adds Functionality to the R Markdown + Shiny Bootstrap Framework
chronicle	Grammar for Creating R Markdown Reports
distill	R Markdown Format for Scientific and Technical Writing
flexdashboard	R Markdown Format for Flexible Dashboards
govdown	GOV.UK Style Templates for R Markdown
jds.rmd	R Markdown Templates for Journal of Data Science
komaletter	Simply Beautiful PDF Letters from Markdown
liftr	Containerize R Markdown Documents for Continuous Reproducibility
mailmerge	Mail Merge Using R Markdown Documents and 'gmailr'
memoiR	R Markdown and Bookdown Templates to Publish Documents
memor	A rmarkdown Template that Can be Highly Customized
officedown	Enhanced R Markdown Format for 'Word' and 'PowerPoint'
pagedown	Paginate the HTML Output of R Markdown with CSS for Print
parsermd	Formal Parser and Related Tools for R Markdown Documents
posterdown	Generate PDF Conference Posters Using R Markdown
prereg	R Markdown Templates to Preregister Scientific Studies
prettydoc	Creating Pretty Documents from R Markdown
quarto	R Interface to 'Quarto' Markdown Publishing System
reportfactory	Lightweight Infrastructure for Handling Multiple R Markdown Documents
revealjs	R Markdown Format for reveal.js Presentations
rmdfiltr	Lua-Filters for R Markdown
rmdformats	HTML Output Formats and Templates for rmarkdown Documents
rmdplugr	Plugins for R Markdown Formats
rmdshower	R Markdown Format for shower Presentations
rticles	Article Formats for R Markdown
siteymlgen	Automatically Generate _site.yml File for R Markdown
stevetemplates	Steve's R Markdown Templates
tufte	Tufte's Styles for R Markdown Documents
tufterhandout	Tufte-style html document format for rmarkdown
uiucthemes	R Markdown Themes for UIUC Documents and Presentations
vitae	Curriculum Vitae for R Markdown
webexercises	Create Interactive Web Exercises in R Markdown (Formerly 'webex')

© 草稿

Package	Title
ymlthis	Write YAML for R Markdown, bookdown, blogdown, and More

13.9 支持网页图形

```
library(ggplot2)
p1 <- ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point() +
  theme_minimal()

p2 <- ggplot(data = iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +
  geom_point() +
  theme_minimal()

library(patchwork)
p1 + p2
```

图 13.11: 组合 ggplot2 图形

ggiraph 将 ggplot 对象转化为网页

```
library(ggiraph)
girafe(code = print(p1 + p2), width_svg = 8, height_svg = 3)
```

13.10 支持 Shiny App

将动态图形嵌入 Shiny App 中

© 谢益辉

图 13.12: A Shiny app created via the ggiraph package; you can see a live version at https://xiangyun.shinyapps.io/01-iris-ggiraph/.

© 黄湘云

第十四章　便携式文档

14.1　文档汉化

从 R Markdown 到 beamer 幻灯片，如何迁移 LaTeX 模版

默认的 PDF 文档 PDF 文档案例

详见PDF 文档案例

14.2　添加水印

draftwatermark

14.3　双栏排版

普通单栏排版改为双栏排版，只需添加文档类选项 `"twocolumn"`，将 YAML 元数据中的

```
classoption: "UTF8,a4paper,fontset=adobe,zihao=false"
```

变为

```
classoption: "UTF8,a4paper,fontset=adobe,zihao=false,twocolumn"
```

其中，参数 `UTF8` 设定文档编码类型，`a4paper` 设置版面为 A4 纸大小，`fontset=adobe` 指定中文字体为 Adobe 字体，`zihao=false` 不指定字体大小，使用文档类 ctexart 默认的字号，

14.4　参数化报告

参数化文档案例

进一步将文档类型做成参数化，实现在运行时自由选择，只需将如下两行替换掉上述一行

```
params:
  classoption: twocolumn
classoption: "`r params$classoption`"
```

如果想要双栏的排版风格，编译时传递 documentclass 参数值，覆盖掉默认的参数值即可

© 黄湘云


```
rmarkdown::render(
  input = "examples/pdf-document.Rmd",
  params = list(classoption = c("twocolumn"))
)
```

14.5 学术幻灯片

beamer 幻灯片也是一种 PDF 文档 PDF 文档案例

Dirk Eddelbuettel 将几个大学的 beamer 幻灯片转化成 R Markdown 模板，收录在 binb 包里，方便调用。伊利诺伊大学的 James J Balamuta 在 R Markdown 基础上专门为自己学校开发了一套的幻灯片模版，全部打包在 uiucthemes 包里。

komaletter 用 Markdown 写信件

memor `memor::pdf_memo()`

hrbrthemes 提供两个文档模版 `hrbrthemes::ipsum_pdf()` 和 `hrbrthemes::ipsum()`

此汉风主题由 林莲枝 开发，LaTeX 宏包已发布在 CTAN 上，使用此幻灯片主题需要将相关的 LaTeX 宏包一块安装。

```
tlmgr install pgfornament pgfornament-han needspace xpatch
```

14.6 文档模版

字体设置

```
---
output:
  pdf_document:
    extra_dependencies:
      DejaVuSansMono:
       - scaled=0.9
      DejaVuSerif:
       - scaled=0.9
      DejaVuSans:
       - scaled=0.9
---
```

```
---
output:
  pdf_document:
    extra_dependencies:
      sourcecodepro:
       - scale=0.85
      sourceserifpro:
       - rmdefault
      sourcesanspro:
       - sfdefault
---
```

14.7 引用文献

Getting started with Zotero, Better BibTeX, and RMarkdown

knitcitations 包可以根据文献数字对象标识符（英文 Digital Object Identifier，简称 DOI）生成引用，以文章《A Probabilistic Grammar of Graphics》[Pu and Kay, 2020] 为例，其 DOI 为 `10.1145/3313831.3376466`，

内部使用 ©

总之，DOI 就像是文章的身份证，是一一对应的关系[1]。

```
library(knitcitations)
citep(x ='10.1145/3313831.3376466')
```

```
[1] "(Pu and Kay, 2020)"
```

在表格的格子中引用参考文献

```
data.frame(
  author = c("Yihui Xie", "Yihui Xie", "Yihui Xie"),
  citation = c("[@xie2019]", "[@xie2015]", "[@xie2016]")
) |>
  knitr::kable(format = "pandoc")
```

author	citation
Yihui Xie	[Xie, 2019]
Yihui Xie	[Xie, 2015]
Yihui Xie	[Xie, 2016]

citr 包提供了快速查找参考文献的 RStudio 插件，不用去原始文献库 `*.bib` 搜索查找，也会自动生成引用，非常方便，极大地提高了工作效率。**citr** 还支持集成 Zotero 文献管理软件，可以直接从 Zotero 中导入参考文献数据库。rbbt 包也提供了类似的功能，只要系统安装 Zotero 软件及其插件 Better Bibtex for Zotero connector。

14.8 自定义块

```
tinytex::tlmgr_install(c('awesomebox', 'fontawesome5'))
```

安装 awesomebox 包，开发仓库在 https://github.com/milouse/latex-awesomebox，这个 LaTeX 宏包的作用是提供几类常用的块，比如提示、注意、警告等

注意这是注意

提示这是提示信息

警告这是警告信息

[1] https://zh.wikipedia.org/wiki/DOI

© 黄湘石

重要这是重要信息

© 黄湘云

第十五章　网页文档

丘怡轩开发的 prettydoc 包提供了一系列模版，方便快速提高网页逼格。另有 Atsushi Yasumoto 开发的 minidown 包非常轻量，但是常用功能都覆盖了。

15.1　幻灯片

谢益辉开发的 xaringan 用于制作网页幻灯片，xaringanthemer 为 xaringan 提供主题定制，xaringanExtra 在 xaringan 之上提供各种功能扩展，xaringanBuilder 为 xaringan 提供多种输出格式。

15.2　电子邮件

emayili 是非常轻量的实现邮件发送的 R 包，其它功能类似的 R 包有 blastula mailR。Rahul Premraj 基于 rJava 开发的 mailR 虽然还未在 CRAN 上正式发布，但是已得到很多人的关注，也被广泛的使用，目前作者已经不维护了，继续使用有一定风险。RStudio 公司 Richard Iannone 新开发的 blastula 扔掉了 Java 的重依赖，更加轻量化、现代化，支持发送群组邮件[1]。curl 包提供的函数 `send_mail()` 本质上是在利用 curl 软件发送邮件，举个例子，邮件内容如下：

```
From: "黄湘云" <邮箱地址>
To: "黄湘云" <邮箱地址>
Subject: 测试邮件

你好：

这是一封测试邮件！
```

将邮件内容保存为 mail.txt 文件，然后使用 curl 命令行工具将邮件内容发出去。

```
curl --url 'smtp://公司邮件服务器地址:开放的端口号' \
  --ssl-reqd --mail-from '发件人邮箱地址' \
  --mail-rcpt '收件人邮箱地址' \
  --upload-file data/mail.txt \
  --user '发件人邮箱地址:邮箱登陆密码'
```

注意

Gmail 出于安全性考虑，不支持这种发送邮件的方式，会将邮件内容阻挡，进而接收不到邮件。

[1] https://thecoatlessprofessor.com/programming/r/sending-an-email-from-r-with-blastula-to-groups-of-students/

© 黄湘云

下面以 blastula 包为例怎么支持 Gmail/Outlook/QQ 等邮件发送，先安装系统软件依赖，CentOS 8 上安装依赖

```
sudo dnf install -y libsecret-devel libsodium-devel
```

然后安装 **keyring** 和 **blastula**

```
install.packages(c("keyring", "blastula"))
```

接着配置邮件帐户，这一步需要邮件账户名和登陆密码，配置一次就够了，不需要每次发送邮件的时候都配置一次

```
library(blastula)
create_smtp_creds_key(
  id = "outlook",
  user = "xiangyunfaith@outlook.com",
  provider = "outlook"
)
```

第二步，准备邮件内容，包括邮件主题、发件人、收件人、抄送人、密送人、邮件主体和附件等。

```
attachment <- "data/mail.txt" # 如果没有附件，引号内留空即可。
# 这个Rmd文件渲染后就是邮件的正文，交互图形和交互表格不适用
body <- "examples/html-document.Rmd"
# 渲染邮件内容，生成预览
email <- render_email(body) |>
  add_attachment(file = attachment)
email
```

最后，发送邮件

```
smtp_send(
  from = c("张三" = "xxx@outlook.com"), # 发件人
  to = c("李四" = "xxx@foxmail.com",
         "王五" = "xxx@gmail.com"), # 收件人
  cc = c("赵六" = "xxx@outlook.com"), # 抄送人
  subject = "这是一封测试邮件",
  email = email,
  credentials = creds_key(id = "outlook")
)
```

密送人实现群发单显，即一封邮件同时发送给多个人，每个收件人只能看到发件人地址而看不到其它收件人地址。

```
email <- compose_email(
  body = md("
Markdown 格式的邮件内容
")
)

smtp_send(
```

© 基础设施

```
  from = c("发件人" = "xx@outlook.com"),
  to = c("收件人" = "xx@outlook.com"),
  bcc = c(
    "抄送人" = "xx@outlook.com"
    ),
  subject = "邮件主题",
  email = email,
  credentials = creds_key(id = "outlook")
)
```

© 黄湘云

第十六章　办公文档

docxtools、officer 和 officedown 大大扩展了 rmarkdown 在制作 Word/PPT 方面的功能。

本节探索 Markdown + Pandoc 以 Word 格式作为最终交付的可能性。R Markdown 借助 Pandoc 将 Markdown 转化为 Word 文档，继承自 Pandoc 的扩展性，R Markdown 也支持自定义 Word 模版，那如何自定义呢？首先，我们需要知道 Pandoc 内建的 Word 模版长什么样子，然后我们依样画葫芦，制作适合实际需要的模版。获取 Pandoc 2.10.1 自带的 Word 和 PPT 模版，只需在命令行中执行

```
# DOCX 模版
pandoc -o custom-reference.docx --print-default-data-file reference.docx
# PPTX 模版
pandoc -o custom-reference.pptx --print-default-data-file reference.pptx
```

这里其实是将 Pandoc 自带的 docx 文档 reference.docx 拷贝一份到 custom-reference.docx，而后将 custom-reference.docx 文档自定义一番，但仅限于借助 MS Word 去自定义样式。Word 文档的 YAML 元数据定义详情见 https://pandoc.org/MANUAL.html#option--reference-doc，如何深度自定义文档模版见 https://bookdown.org/yihui/rmarkdown/word-document.html ，其它模版见 GitHub 仓库 pandoc-templates。这里提供一个Word 文档案例供读者参考。**bookdown** 提供的函数 `word_document2()` 相比于 **rmarkdown** 提供的 `word_document()` 支持图表的交叉引用，更多细节详见帮助 `?bookdown::word_document2`。

注意

R Markdown 文档支持带编号的 Word 文档格式输出要求 Pandoc 版本 2.10.1 及以上，rmarkdown 版本 2.4 及以上。

© 王湘君

第十七章　工作流

drake 一站式可重复性研究工作空间打造者，用户手册 https://books.ropensci.org/drake/ 和学习材料 https://github.com/wlandau/learndrake

© 黄湘云

第十八章　高级文档

18.1　编写书籍

此外，ElegantTufteBookdown 项目提供了 tufte 风格的书籍模板，本书配套的仓库目录 `examples/` 下准备了一系列常用模板。

18.2　个人网站

18.3　R 包文档

18.4　课程网站

© 黄湘云

第四部分

数据产品

© 黄湘云

介绍

数据产品

© 黄湘云

第十九章　交互表格

Greg Lin 开发的 reactable 包覆盖测试达到惊人的 99%，它基于 JavaScript 库 react-table，是 react 框架的衍生品，Nick Raienko 整理了一份超棒的 react 模块合集 也许机智如你，可以引入更多优秀的 react 模块到 R 语言社区。reactablefmtr 提供一些函数简化 reactable 定制表格的复杂性

谢益辉开发的 DT 包覆盖测试 31%，它基于 DataTables 库，是 jQuery 框架的衍生品。益辉评价 reactable 在多个方面优于 DT，比如行分组和聚合，嵌入 HTML widgets，甚至说要是 **reactable** 存在于 **DT** 之前，他就不会新开发 DT 这个 R 包了，不过这是后话了[1]。

Richard Iannone 开发的 gt 包覆盖测试 78%，类似 **ggplot2** 的设计哲学，试图打造制作表格的语法，相比于 **reactable** 和 **DT**，它不依赖于 JavaScript 库，更加轻量，一般来讲，持续维护更新重 JS 库依赖的 R 包比较累人，JS 库可能会不断重构，进而变动 API。

朱昊开发的 kableExtra 大大扩展了 **knitr** 包的 `kable()` 函数的功能，虽没有覆盖测试，但中英文文档特别详细，见官网 https://haozhu233.github.io/kableExtra/。

目前，Greg Lin、谢益辉和 Richard Iannone 都是 RStudio 公司雇员，他们背靠开源组织和大公司，开发的这些 R 包的生命力都比较强。**gt** 和 **kableExtra** 摆脱了 JavaScript 库的依赖，网页形式的表格可以嵌入到邮件内容中，这是一个不太引人注意的优势。**kableExtra** 还支持高度自定义的 LaTeX 输出，详见案例 https://github.com/XiangyunHuang/bookdown-kableExtra，**gt** 包据说未来也会支持，拭目以待吧，也许在成书之日能看到！

此外，还有任坤开发的 **formattable** 和 David Gohel 开发的 **flextable** 包等，一份综合介绍见博文 How to Make Beautiful Tables in R。

rtables 处于原型开发的阶段，针对复杂表格，有比较好的设计。tablesgg 使用 ggplot2 将表格渲染成图片。

19.1　DT 和 reactable

DT 基于 jQuery 的 JS 库 DataTables 提供了一个 R 的封装，封装工具和许多其他基于 JS 库的 R 包一样，比如即将介绍的 **reactable** 包，都依赖于 htmlwidgets。

```
library(magrittr)
```

```
if (!is.na(Sys.getenv('CI', NA))) {
  Sys.setenv(R_CRAN_WEB = "https://cloud.r-project.org/")
} else {
  Sys.setenv(R_CRAN_WEB = "https://mirrors.tuna.tsinghua.edu.cn/CRAN")
}
```

[1] https://bookdown.org/yihui/rmarkdown-cookbook/table-other.html

© 谢益辉


```
pdb <- tools::CRAN_package_db()
sub_pdb <- subset(pdb, subset = !duplicated(pdb[, "Package"]) & pdb[, "Package"] %in% .packages(T))
pkg_pdb <- subset(sub_pdb,
  subset = grepl("Yihui Xie", sub_pdb[, "Maintainer"]) | grepl("Hadley Wickham", sub_pdb[, "Maintaine
  select = c("Maintainer", "Package", "Version", "Published", "Title")
)

pkg_pdb <- transform(pkg_pdb, Title = gsub("(\\\n)", " ", Title))
```

```
library(DT)
```

```
datatable(pkg_pdb[order(pkg_pdb$Maintainer, decreasing = T), ],
  rownames = F, # 不显示行名
  extensions = c("Buttons", "RowGroup"),
  options = list(
    pageLength = 10, # 每页显示的行数
    language = list(url = "//cdn.datatables.net/plug-ins/1.10.11/i18n/Chinese.json"), # 汉化
    dom = "Brtp", # 去掉显示行数 i、过滤 f 的能力，翻页用 p 表示
    ordering = F, # 去掉列排序
    buttons = c("copy", "csv", "excel", "pdf", "print"), # 提供打印按钮
    rowGroup = list(dataSrc = 0), # 按 Maintainer 列分组
    columnDefs = list(
      list(className = "dt-center", targets = 0), # 不显示行名，则 targets 从 0 开始，否则从 1 开始
      list(visible = FALSE, targets = 0) # 不显示 Maintainer 列
    )
  ),
  caption = "谢大和哈神维护的 R 包"
)
```

```
colorize_num <- function(x) {
  ifelse(x > 0,
    sprintf("<span style='color:%s'>%s</span>", "green", x),
    sprintf("<span style='color:%s'>%s</span>", "red", x)
  )
}
colorize_pct <- function(x) {
  ifelse(x > 0,
    sprintf("<span style='color:%s'>%s</span>", "green", scales::percent(x, accuracy = 0.01)),
    sprintf("<span style='color:%s'>%s</span>", "red", scales::percent(x, accuracy = 0.01))
  )
}

colorize_pp <- function(x) {
  ifelse(x > 0,
    sprintf("<span style='color:%s'>%s</span>", "green", paste0(round(100*x, digits = 2), "PP")),
```

© 黄湘云

```
    sprintf("<span style='color:%s'>%s</span>", "red", paste0(round(100*x, digits = 2), "PP"))
  )
}

colorize_text <- function(x, color = "red") {
    sprintf("<span style='color:%s'>%s</span>", color, x )
}

library(tibble)

dat = tribble(
  ~name1, ~name2,
  as.character(htmltools::tags$b("加粗")), as.character(htmltools::a(href = "https://rstudio.com", "超链")
  as.character(htmltools::em("强调")), '<a href="#" onclick="alert(\'Hello World\');">Hello</a>',
  as.character(htmltools::span(style = 'color:red', "正常")), '正常'
)

datatable(
  data = dat,
  escape = F, # 设置 escape = F
  colnames = c(colorize_text("第1列", "red"), as.character(htmltools::em("第2列"))),
  caption = htmltools::tags$caption(
    style = "caption-side: top; text-align: center;",
    "表格 2: ", htmltools::em("表格标题")
  ), # 在表格底部显示标题，默认在表格上方显示标题
  # filter = "top", # 过滤框
  options = list(
    pageLength = 5, # 每页显示5行
    dom = "t"
  )
)
```

下面重点介绍 reactable 包，看看 React.js 和 Shiny 是如何集成的，这是比较高级的主题，主要参考 Alan Dipert 的演讲材料 Integrating React.js and Shiny。

```
library(reactable)
```

下面这个例子来自 React.js 官网 https://reactjs.org/

```
```js
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
```
```

© 王沛栋


```
  }
}

ReactDOM.render(
  <HelloMessage name="Taylor" />,
  document.getElementById('hello-example')
);
```

更多细节定制见 Thomas Mock 的博文 reactable - An Interactive Tables Guide

reactable 制作表格

```
library(shiny)
library(reactable)

ui <- fluidPage(
  reactableOutput("table")
)

server <- function(input, output) {
  output$table <- renderReactable({
    reactable(iris,
      filterable = TRUE, # 过滤
      searchable = TRUE, # 搜索
      showPageSizeOptions = TRUE, # 页面大小
      pageSizeOptions = c(5, 10, 15), # 页面大小可选项
      defaultPageSize = 10, # 默认显示10行
      highlight = TRUE, # 高亮选择
      striped = TRUE, # 隔行高亮
      fullWidth = FALSE, # 默认不要全宽填充，适应数据框的宽度
      defaultSorted = list(
        Sepal.Length = "asc", # 由小到大排序
        Petal.Length = "desc" # 由大到小
      ),
      columns = list(
        Sepal.Width = colDef(style = function(value) { # Sepal.Width 添加颜色标记
          if (value > 3.5) {
            color <- "#008000"
          } else if (value > 2) {
            color <- "#e00000"
          } else {
            color <- "#777"
          }
          list(color = color, fontWeight = "bold")
        })
```

© 非洲出版

```
      )
    )
  })
}

shinyApp(ui, server)
```

```
# 修改自 Code: https://gist.github.com/jthomasmock/f085dce3e70e42ca49b052bbe25de49f
library(reactable)
library(htmltools)

# barchart function from: https://glin.github.io/reactable/articles/building-twitter-followers.html
bar_chart <- function(label, width = "100%", height = "14px", fill = "#00bfc4", background = NULL) {
  bar <- div(style = list(background = fill, width = width, height = height))
  chart <- div(style = list(flexGrow = 1, marginLeft = "6px", background = background), bar)
  div(style = list(display = "flex", alignItems = "center"), label, chart)
}

data <- mtcars |>
  subset(select = c("cyl", "mpg")) |>
  subset(subset = sample(x = c(TRUE, FALSE), size = 6, replace = T))


reactable(
  data,
  defaultPageSize = 20,
  columns = list(
    cyl = colDef(align = "center"),
    mpg = colDef(
      name = "mpg",
      defaultSortOrder = "desc",
      minWidth = 250,
      cell = function(value, index) {
        width <- paste0(value * 100 / max(mtcars$mpg), "%")
        value <- format(value, width = 9, justify = "right", nsmall = 1)

        # output the value of another column
        # that aligns with current value
        cyl_val <- data$cyl[index]

        # Color based on the row's cyl value
        color_fill <- if (cyl_val == 4) {
          "#3686d3" # blue
        } else if (cyl_val == 6) {
```

© 谢湘


```
          "#88398a" # purple
        } else {
          "#fcab27" # orange
        }
        bar_chart(value, width = width, fill = color_fill, background = "#e1e1e1")
      },
      align = "left",
      style = list(fontFamily = "monospace", whiteSpace = "pre")
    )
  )
)
```

19.2 gt 和 kableExtra

如表 19.1 所示，我们可以自定义表格样式，比如配色，例子修改自 kableExtra 帮助文档 https://haozhu233.github.io/kableExtra/bookdown/cross-format-tables-in-bookdown.html，同时支持 HTML 和 LaTeX 输出，但是 LaTeX 输出需要在文档类选项中增加 table 选项，即 `classoption: "table"`，这样就可以加载 colortbl 宏包，进而提供 `\rowcolor` 等 LaTeX 命令，在表格中给每个格子定制颜色。我们推荐在 classoption 中添加 table 选项，而不是再次加载 xcolor 包，比如像这样 `\usepackage[table]{xcolor}`，这会在 R Markdown 中引起冲突 [2]。

```
library(kableExtra)

iris[1:10, ] %>%
  transform(
    Sepal.Length =
      cell_spec(Sepal.Length,
        bold = T,
        color = spec_color(Sepal.Length, end = 0.9)
      )
  ) %>%
  transform(Species = cell_spec(
    Species,
    color = "white", bold = T,
    background = spec_color(1:10,
      end = 0.9,
      option = "A", direction = -1
    )
  )) %>%
  kable(
    escape = F, align = "c", booktabs = T,
    caption = "自定义表格样式"
  ) %>%
```

[2] https://stackoverflow.com/questions/50094698/rmarkdown-beamer-presentation-option-clash-clash-for-xcolor

© 基础统计

表 19.1: 自定义表格样式

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa

```
  kable_styling(c("striped", "condensed"),
    latex_options = "striped",
    full_width = F
  )
```

一个非常基本的 gt 制作的表格

```
library(gt)
iris %>%
  head() %>%
  gt()
```

然后添加表格的标题和副标题，套上 `md()` 函数后，标题和副标题支持 Markdown 语法，告别 HTML 的制表方式吧！其它表格元素，如脚注支持和表格的列指标关联

```
library(data.table)

iris %>%
  as.data.table %>%
  .[, head(.SD, 2), by = .(Species)] %>%
  gt() %>%
  tab_header(
    title = md("**鸢尾花**数据集"),
    subtitle = "R 内置数据集"
  ) %>%
  data_color(
    columns = vars(Sepal.Length),
    colors = scales::col_numeric(palette = terrain.colors(5, rev = T), domain = NULL)
  ) %>%
  data_color(
    columns = vars(Species),
    colors = scales::col_factor(palette = hcl.colors(3), domain = NULL)
```

©黄湘云


```
  ) %>%
  tab_footnote(
    footnote = md("据说数据集最早收集自 Fisher's or Anderson's"),
    locations = cells_column_labels(columns = vars(Sepal.Length))
  ) %>%
  tab_footnote(
    footnote = "鸢尾花的类别",
    locations = cells_column_labels(
      columns = vars(Species)
    )
  )
```

更多细节的设置见 Thomas Mock 的博文gt - a (G)rammar of (T)ables

注意

当前 gt 包对 LaTeX 的支持比较弱，上述表格在 HTML 网页环境中可以看到的效果并不能一一对应到 LaTeX 输出中。且 gt 包生成 LaTeX 表格会自动加载宏包 amsmath、booktabs、caption 和 longtable，`gt_latex_dependencies()` 且不能控制

19.3 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] kableExtra_1.3.4 reactable_0.2.3  DT_0.22          magrittr_2.0.3
```

© 黄湘云 版权所有

```
##
## loaded via a namespace (and not attached):
##  [1] rstudioapi_0.13   knitr_1.38        xml2_1.3.3        sysfonts_0.8.8
##  [5] munsell_0.5.0     rvest_1.0.2       viridisLite_0.4.0 colorspace_2.0-3
##  [9] R6_2.5.1          rlang_1.0.2       fastmap_1.1.0     stringr_1.4.0
## [13] httr_1.4.2        tools_4.1.3       webshot_0.5.2     xfun_0.30
## [17] cli_3.2.0         systemfonts_1.0.4 htmltools_0.5.2   yaml_2.3.5
## [21] digest_0.6.29     lifecycle_1.0.1   bookdown_0.25     htmlwidgets_1.5.4
## [25] curl_4.3.2        glue_1.6.2        evaluate_0.15     rmarkdown_2.13
## [29] stringi_1.7.6     compiler_4.1.3    scales_1.1.1      svglite_2.1.0
```

© 基石

第二十章　交互报表

学习 shiny 应用开发，建议多看看 Learn Shiny。了解 shiny server ，推荐从 Shiny Server Professional Administrator's Guide 开始。了解 shiny 相关的生态，建议从 shiny 资源列表 https://github.com/grabear/awesome-rshiny 和 shiny 扩展合集 https://github.com/nanxstats/awesome-shiny-extensions 开始，希望读者能从中打造属于自己的最佳实践。

RStudio 首席技术官 CTO Joe Cheng 在 2019 年 RStudio 大会上介绍 企业级 shiny 应用原理、实践和工具 可以作为 shiny 从新技术到生产力的蜕变节点。支持高并发的异步编程，比如 Heather Nolis 和 Dr. Jacqueline Nolis 的报告介绍了日百万访问量下的 shiny 应用如何搭建[1]。Colin Fay, Sébastien Rochette, Vincent Guyader, Cervan Girard 的书 Engineering Production-Grade Shiny Apps、David Granjon 的书Outstanding User Interfaces with Shiny 和 Hadley Wickham 的书 Mastering Shiny 的问世宣告 shiny 的成熟稳定，以及生态的形成，在此之前 shiny 一直不被看好。shiny 生态意味着一个完整的工业级的应用圈，满足安全性、稳定性、高效性、维护性、扩展性的要求。

iSEE is winner of the Most Technically Impressive award of the 2019 Shiny Contest. 源码地址 https://github.com/iSEE/isee-shiny-contest

Six Years of Shiny in Research - Collaborative Development of Web Tools in R [Kasprzak et al., 2021]

以 RStudio 为核心，开发 Shiny 应用扩展的社区组织有 RStudio、Appsilon、RinteRface、ThinkR-open、dreamRs 和datastorm-open

20.1　开发流程

报表开发从数据仓库的 DWD 层开始，可能一些业务原因，我们需要从 ODS 层甚至从点击流的日志数据开始，经过数据清洗、提取、聚合成为支撑 BI 报表最底层的基础表，存储在 Hive 中，然后对这一系列的基础表根据 BI 展示的需要进行第二层聚合形成中间表，这两层数据根据业务情况做增量更新或者全量更新，并将中间表同步到 MySQL 仓库中，全量更新的情况，往往更新数据比较大，建议用 sqoop 做数据的同步。创建第二层的中间表稍有些灵活性，原则是在中间表之上对应的数据操作和可视化是容易实现且效率较高的，否则应该构造第三层的中间表，绝不能将大规模的数据集直接导入 R 中进行分析和可视化，拖慢前端展示的速度，占用过多的服务器资源。

[1] https://resources.rstudio.com/rstudio-conf-2020/we-re-hitting-r-a-million-times-a-day-so-we-made-a-talk-about-it-heather-nolis-dr-jacqueline-nolis

© 黄湘云

图 20.1: Shiny 生态系统

© 黄湘云

图 20.2: 开发 Shiny 应用扩展的组织

© 王慧勇

20.2 开发工具

除了在第 20.1 节介绍的和数据库紧密相关的工具外，我们还需要 Git 做代码管理，Azkaban 做任务调度（或者其它工具做任务调度器），RStudio IDE 做开发工具（或者 VS Code 等），Shiny Server 做报表支撑，做报表管理。具体到 shiny 页面开发，我们需要：

- RMySQL 做数据库连接，推荐 odbc 这个包，它支持连接相当广泛的数据库。
- data.table 或者 dplyr 做数据操作，推荐和管道操作 magrittr 一起使用，增加代码可读性。
- plotly 或者 highcharter 做数据可视化，reactable 和 DT 做数据呈现，也可以 ggplot2 和 plotly 的 `ggplotly()` 函数共同实现静态图到动态图的交互可视化。
- shiny 及其扩展工具做页面设计，比如 shinythemes 可以统一配色，dashboardthemes 提供更加深度的主题，shinytableau 提供仿 Tableau 的 dashboard 框架。sass 在 CSS 样式层面重定义网站风格，比如借助 sass 修改 Bootstrap 4，shiny 的布局其实就是魔改了 Bootstrap 库。
- 针对特定应用场景的其它交互可视化工具包，比如 leaflet 可以将地图嵌入 Shiny 应用，dygraphs 可以将时间序列塞进去。
- 其它加强 shiny 页面的小功能，比如 shinyFeedback 提供用户输入的反馈，miniUI 专为小屏幕设计，shinyMobile 在 IOS 和安卓手机上访问 shiny 应用，大大加强 miniUI 的功能，shinyWidgets 提供自定义 widget 的功能，shinymanager 支持单个 shiny 应用的权限管理，firebase 提供访问权限设置 https://firebase.john-coene.com/。
- shiny-server 以网络服务的方式支持 shiny 应用，是企业级 shiny 应用的核心，shinyproxy 提供企业级部署 shiny 应用的开源解决方案，ShinyStudio 打造基于容器架构的协作开发环境的开源解决方案，golem 构建企业级 shiny 应用的框架，RinteRface 开发的系列 R 包也试图打造一套完整的解决方案，并配有速查小抄 cheatsheets
- radiant 探索性分析解决方案

shinyauthr 应用授权

```
library(shiny)
```

20.3 基础知识

1920s 汽车数据分析和建模

20.4 基础组件

20.4.1 书签

链接可以指向页面状态

```
library(shiny)

ui <- function(request) {
  fluidPage(
    plotOutput("plot"),
    sliderInput("n", "Number of observations", 1, nrow(faithful), 100),
```

© 基础出版


```
    bookmarkButton()
  )
}

server <- function(input, output, session) {
  output$plot <- renderPlot({
    hist(faithful$eruptions[seq_len(input$n)], breaks = 40)
  })
}

enableBookmarking(store = "url")
shinyApp(ui, server)
```

20.4.2 表格

reactable 基于 JS 库 React Table 提供交互式表格渲染，和 shiny 无缝集成，是替代 DT 的不二选择，在 app.R 用 reactable 包的 `reactableOutput()` 和 `renderReactable()` 函数替代 shiny 里面的 `dataTableOutput()` 和 `renderDataTable()`。再也不用忍受 DT 和 shiny 的函数冲突了，且其覆盖测试达到 99%。

```
library(shiny)
library(data.table)
```

gt 高度自定义 gt 表格样式，支持 shiny 集成，data.table 提供高效的数据操作，formattable 支持自定义格子。

kableExtra 包

```
library(shiny)
library(data.table)
library(magrittr)
library(kableExtra)

ui <- fluidPage(
  title = "mtcars datasets",
  titlePanel("mtcars 数据集"),

  sidebarLayout(
    sidebarPanel(
      sliderInput("mpg", "mpg 范围",
        min = 11, max = 33, value = 15
      )
    ),

    mainPanel(
      tableOutput("mtcars_kable")
    )
```

© 黄湘云

```
  )
)

## 设置列序 https://stackoverflow.com/questions/19619666/change-column-position-of-data-table
server <- function(input, output) {
  output$mtcars_kable <- function() {
    # 转化数据类型
    mtcars_dt <- as.data.table(mtcars)
    # 添加新的列
    mtcars_dt[, car := rownames(mtcars)][mpg <= input$mpg] %>%
    setcolorder(., c("car", setdiff(names(.), "car"))) %>%
      knitr::kable("html") %>%
      kable_styling("striped", full_width = F) %>%
      add_header_above(c(" ", "Group 1" = 5, "Group 2" = 6))
  }
}

# 执行程序
shinyApp(ui = ui, server = server)
```

reactable 包

```
library(shiny)
library(reactable)

ui <- fluidPage(
  reactableOutput("table")
)

server <- function(input, output) {
  output$table <- renderReactable({
    reactable(iris,
      filterable = TRUE, # 过滤
      searchable = TRUE, # 搜索
      showPageSizeOptions = TRUE, # 页面大小
      pageSizeOptions = c(5, 10, 15), # 页面大小可选项
      defaultPageSize = 10, # 默认显示10行
      highlight = TRUE, # 高亮选择
      striped = TRUE, # 隔行高亮
      fullWidth = FALSE, # 默认不要全宽填充，适应数据框的宽度
      defaultSorted = list(
        Sepal.Length = "asc", # 由小到大排序
        Petal.Length = "desc" # 由大到小
      ),
      columns = list(
        Sepal.Width = colDef(style = function(value) { # Sepal.Width 添加颜色标记
```

© 基础


```
          if (value > 3.5) {
            color <- "#008000"
          } else if (value > 2) {
            color <- "#e00000"
          } else {
            color <- "#777"
          }
          list(color = color, fontWeight = "bold") # 字体加粗
        })

      )
    )
  })
}

shinyApp(ui, server)
```

下面介绍 DT

```
library(magrittr)
# ui.R 前端
library(shiny)
shinyUI(fluidPage(
  # 应用的标题名称
  titlePanel("鸢尾花数据集"),
  # 边栏
  fluidRow(
    column(
      12,
      DT::dataTableOutput("table")
    )
  )
))

# server.R 服务端
library(shiny)
shinyServer(function(input, output, session) {
  output$table <- iris %>%
    `colnames<-`(., gsub("\\.", "_", tolower(colnames(.)))) %>%
    DT::renderDataTable(.,
      options = list(
        pageLength = 5, # 每页显示5行
        initComplete = I("function(settings, json) {alert('Done.');}")
      ), server = F
    )
})
```

© 黄湘云

注意

加载 shiny 包后再加载 DT 包，函数 `dataTableOutput()` 和 `renderDataTable()` 显示冲突，因为两个 R 包都有这两个函数。在创建 shiny 应用的过程中，如果我们需要呈现动态表格，就需要使用 DT 包的 `DT::dataTableOutput()` 和 `DT::renderDataTable()` 否则会报错，详见 https://github.com/rstudio/shiny/issues/2653，DT 包官方文档 https://rstudio.github.io/DT/。

提示

在 server.R 里我们对数据集 iris 做了重命列名的操作，如果不使用管道操作，通常是下面这样操作。

```
colnames(iris) <- gsub("\\.", "_", tolower(colnames(iris)))
```

换成管道操作，函数 `colnames()` 要换成 `colnames<-`，这其实类似于 `1 + 2` 换成 `+(1, 2)`，保持函数在左边，参数值在右边的一致性。

设置页面默认显示的行数和列的宽度

```
# https://stackoverflow.com/questions/45509501/set-names-of-values-in-lengthmenu-page-length-menu-in-r-dt-
# 相关例子见 https://github.com/rstudio/shiny-examples/tree/master/018-datatable-options
# DT 选项 https://rstudio.github.io/DT/options.html

library(shiny)
library(DT)

ui <- fluidPage(
  DT::dataTableOutput("table")
)

server <- function(input, output) {
  output$table <- DT::renderDataTable({
    DT::datatable(iris, options = list(
      language = list(url = "//cdn.datatables.net/plug-ins/1.10.11/i18n/Chinese.json"),
      pageLength = 24, # 设置页面默认显示的行数
      lengthMenu = list(
        c(24, 48, 72, 96, -1),
        c("24", "48", "72", "96", "All")
      ),
      paging = T,
      # 设置第一列和第三列的宽度 https://rstudio.github.io/DT/options.html
      autoWidth = TRUE, columnDefs = list(list(width = '400px', targets = c(1, 3)))
    ))
  })
}

shinyApp(ui, server)
```

按指定格式显示数据

© 黄湘云


```
# data <- data.frame(x = c(100.0011, 80.0011, -90.0011, -110.0011, -70))
#
# library(shiny)
# runApp(list(
#   ui = fluidPage(dataTableOutput("num")),
#   server = function(input, output) {
#     output$num = renderDataTable(format(round(data, 3), nsmall = 3))
#   }
# ))

library(DT)

dat <- data.frame(x = c(100.0011, 80.0011, -90.0011, -110.0067, -70))

rowCallback <- c(
  "function(row, data, index){",
  "  var N = data.length;",
  "  for(var j=1; j<data.length; j++){",
  "    $('td:eq('+j+')',row)",
  "      .html(parseFloat(data[j]).toFixed(3));", # 四舍五入保留 3 位小数
  "  }",
  "}"
)

# https://github.com/rstudio/shiny/issues/2277
datatable(dat,
  options = list(
    rowCallback = JS(rowCallback)
  )
)
```

20.5 高级主题

异步编程，并发访问

```
## shiny 异步编程
## 解决问题，多人同时访问 Shiny 应用的情况下，必须等另一个人完成访问的情况下才能继续访问

library(shiny)
library(future)
library(promises)

plan(multiprocess)
```

© 基础

```
ui <- fluidPage(
  h2("测试异步下载"),
  tags$ol(
    tags$li("Verify that plot appears below"),
    tags$li("Verify that pressing Download results in 5 second delay, then rock.csv being downloaded"),
    tags$li("Check 'Throw on download?' checkbox and verify that pressing Download results in 5 second del
  ),
  hr(),
  checkboxInput("throw", "Throw on download?"),
  downloadButton("download", "下载 (等待5秒)"),
  plotOutput("plot")
)

server <- function(input, output, session) {
  output$download <- downloadHandler("rock.csv", function(file) {
    future({Sys.sleep(5)}) %...>%
      {
        if (input$throw) {
          stop("boom")
        } else {
          write.csv(rock, file)
        }
      }
  })

  output$plot <- renderPlot({
    plot(cars)
  })
}

shinyApp(ui, server)
```

20.6 部署应用

20.7 最佳实践

提升 shiny 仪表盘访问性能的 4 个建议

20.8 仪表盘

dashboard 翻译过来叫仪表盘，就是驾驶仓的那个玩意，形象地表达作为掌舵者应该关注的对象。R 包 shiny 出现后，仪表盘的制作显得非常容易，也很快形成了一个生态，比如 shinydashboard、flexdashboard

© 基础

等，此外 bs4Dash 基于 Bootstrap 4 的仪表盘，目前 shiny 和 rmarkdown 都在向 Bootstrap 4 升级，这是未来的方向。shinydashboardPlus 主要目的在于扩展 **shinydashboard** 包

shinydashboard 包

```
## app.R ##
library(shiny)
library(shinydashboard)

ui <- dashboardPage(
    dashboardHeader(title = "Basic dashboard"),
    ## Sidebar content
    dashboardSidebar(
        sidebarMenu(
            menuItem("Dashboard", tabName = "dashboard", icon = icon("dashboard")),
            menuItem("Widgets", tabName = "widgets", icon = icon("th"))
        )
    ),
    ## Body content
    dashboardBody(
        tabItems(
            # First tab content
            tabItem(tabName = "dashboard",
                    fluidRow(
                        box(plotOutput("plot1", height = 250)),

                        box(
                            title = "Controls",
                            sliderInput("slider", "Number of observations:", 1, 100, 50)
                        )
                    )
            ),

            # Second tab content
            tabItem(tabName = "widgets",
                    h2("Widgets tab content")
            )
        )
    )
)

server <- function(input, output) {
    set.seed(122)
    histdata <- rnorm(500)

    output$plot1 <- renderPlot({
```

© 基础统计

```
        data <- histdata[seq_len(input$slider)]
        hist(data)
    })
}

shinyApp(ui, server)
```

shinydashboardPlus 包

```
library(shiny)
library(shinydashboard)
library(shinydashboardPlus)

shinyApp(
  ui = dashboardPage(
    dashboardHeader(),
    dashboardSidebar(),
    dashboardBody(
      box(
        solidHeader = FALSE,
        title = "Status summary",
        background = NULL,
        width = 4,
        status = "danger",
        footer = fluidRow(
          column(
            width = 6,
            descriptionBlock(
              number = "17%",
              numberColor = "green",
              numberIcon = "fa fa-caret-up",
              header = "$35,210.43",
              text = "TOTAL REVENUE",
              rightBorder = TRUE,
              marginBottom = FALSE
            )
          ),
          column(
            width = 6,
            descriptionBlock(
              number = "18%",
              numberColor = "red",
              numberIcon = "fa fa-caret-down",
              header = "1200",
              text = "GOAL COMPLETION",
              rightBorder = FALSE,
```

© 王建硕


```
              marginBottom = FALSE
            )
          )
        )
      )
    ),
    title = "Description Blocks"
  ),
  server = function(input, output) { }
)
```

shinymaterial 包

```
library(shiny)
library(shinymaterial)

# https://ericrayanderson.github.io/shinymaterial/
# https://github.com/ericrayanderson/shinymaterial

# Wrap shinymaterial apps in material_page
ui <- material_page(
  title = "用户画像",
  nav_bar_fixed = TRUE,
  # 每个 sidebar 内容
  material_side_nav(
    fixed = TRUE,
    # Place side-nav tabs within side-nav
    material_side_nav_tabs(
      side_nav_tabs = c(
        "数据汇总" = "tab_1",
        "趋势信息" = "tab_2"
      ),
      icons = c("cast", "insert_chart")
    )
  ),
  # 每个 tab 页面的内容
  material_side_nav_tab_content(
    side_nav_tab_id = "tab_1",
    tags$h2("第一个tab页")
  ),
  material_side_nav_tab_content(
    side_nav_tab_id = "tab_2",
    tags$h2("第二个tab页")
  )
)
```

© 统计出版社

```
server <- function(input, output) {

}
shinyApp(ui = ui, server = server)
```

miniUI 包

```
library(shiny)
library(miniUI)
library(leaflet)
library(ggplot2)

ui <- miniPage(
  gadgetTitleBar("Shiny gadget example"),
  miniTabstripPanel(
    miniTabPanel("Parameters", icon = icon("sliders"),
                 miniContentPanel(
                   sliderInput("year", "Year", 1978, 2010, c(2000, 2010), sep = "")
                 )
    ),
    miniTabPanel("Visualize", icon = icon("area-chart"),
                 miniContentPanel(
                   plotOutput("cars", height = "100%")
                 )
    ),
    miniTabPanel("Map", icon = icon("map-o"),
                 miniContentPanel(padding = 0,
                                  leafletOutput("map", height = "100%")
                 ),
                 miniButtonBlock(
                   actionButton("resetMap", "Reset")
                 )
    ),
    miniTabPanel("Data", icon = icon("table"),
                 miniContentPanel(
                   DT::dataTableOutput("table")
                 )
    ),
    selected = "Map"
  )
)

server <- function(input, output, session) {
  output$cars <- renderPlot({
    require(ggplot2)
    ggplot(cars, aes(speed, dist)) + geom_point()
```

©黄湘云

```
  })

  output$map <- renderLeaflet({
    force(input$resetMap)

    leaflet(quakes, height = "100%") %>% addTiles() %>%
      addMarkers(lng = ~long, lat = ~lat)
  })

  output$table <- DT::renderDataTable({
    diamonds
  })

  observeEvent(input$done, {
    stopApp(TRUE)
  })
}

shinyApp(ui, server)
```

20.9 交互式数据报表 dash

```
library(dash)
library(dashHtmlComponents)
library(dashCoreComponents)
library(dashTable)
```

20.10 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
```

© 潘建平

```
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] data.table_1.14.2 shiny_1.7.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8.3     knitr_1.38       magrittr_2.0.3   sysfonts_0.8.8
##  [5] xtable_1.8-4     R6_2.5.1         rlang_1.0.2      fastmap_1.1.0
##  [9] stringr_1.4.0    tools_4.1.3      xfun_0.30        png_0.1-7
## [13] cli_3.2.0        htmltools_0.5.2  ellipsis_0.3.2   yaml_2.3.5
## [17] digest_0.6.29    lifecycle_1.0.1  bookdown_0.25    later_1.3.0
## [21] promises_1.2.0.1 curl_4.3.2       evaluate_0.15    mime_0.12
## [25] rmarkdown_2.13   stringi_1.7.6    compiler_4.1.3   httpuv_1.6.5
```

© 黄湘云

第五部分

统计基础

© 黄湘云

介绍

统计基础

© 黄湘云

第二十一章　抽样分布

分布我们已经听说过很多了，可是它们都是凭空臆测的吗？肯定不是，那它们是怎么产生的呢？谁提出了正态分布，他/她是怎么提出的？一定有故事背景，一定有数据记录，即观察值，我们的样本数据

抽样分布其中抽样二字更加贴近生活，说明它源于实际生产场景，而不是光靠大脑思维理论推导出来的东西，它是最本质的

21.1　正态分布

分三块介绍

- 历史背景
- 分布性质
- 应用场景

来源，为啥叫逻辑斯谛？历史故事

逻辑斯谛分布

1. 正态分布
2. t 分布
3. F 分布
4. χ^2 分布
5. 霍特林 T^2 分布 Hoteling's T^2 Distribution
6. 威沙特分布 Wishart Distribution

分一元和多元情况阐述正态分布、t 分布、F 分布、卡方分布及分布拟合

常见分布之间的关系图需要用 TikZ 来绘制

完整的关系图 http://www.math.wm.edu/~leemis/2008amstat.pdf 参考自 https://www.math.wustl.edu/~jmding/math494/dist.pdf

图来自 [Leemis, 1986]

fitdistrplus

21.2　指数族

谁提出的指数族，有哪些性质，指数族 quasi-poisson 是什么含义，拟族

©黄湘云

如何判别一个分布是否属于指数族

常见的高斯、二项、正态分布、伽马分布、泊松分布

指数族

推广到一般情况

> 三大抽样分布 t 分布，χ 分布和 F 分布，一元和多元情形，一元分布知识范围是本科，多元分布范围是研究生和博士，参考数理统计引论。一元分布多用于本科假设检验，多元分布常用于均值向量和协方差阵以及统计量的极限分布。介绍各个分布的形式、历史来源、各个特征量、密度、分布函数推导，数值计算

三大抽样的发现、历史、多元、非中心形式的推广

多元 t 分布函数 (MVT)

$$T(\mathbf{a},\mathbf{b},\Sigma,\nu)=\frac{2^{1-\frac{\nu}{2}}}{\Gamma(\frac{\nu}{2})}\int_0^\infty s^{\nu-1}e^{-\frac{s^2}{2}}\Phi(\frac{s\mathbf{a}}{\sqrt{\nu}},\frac{s\mathbf{b}}{\sqrt{\nu}},\Sigma)ds$$

多元正态分布函数 (MVN)

$$\Phi(\mathbf{a},\mathbf{b},\Sigma)=\frac{1}{\sqrt{|\Sigma|(2\pi)^m}}\int_{a_1}^{b_1}\int_{a_2}^{b_2}\cdots\int_{a_m}^{b_m}e^{-\frac{1}{2}x^\top\Sigma^{-1}x}dx$$

其中 $x=(x_1,x_2,\ldots,x_m)^\top,\forall i,-\infty\le a_i\le b_i\le\infty$，$\Sigma$ 是 $m\times m$ 对称非负定的矩阵

多元 t 分布分位数计算

```
library(mvtnorm)
n <- c(26, 24, 20, 33, 32)
V <- diag(1 / n)
df <- 130
C <- matrix(c(
  1, 1, 1, 0, 0, -1, 0, 0, 1, 0,
  0, -1, 0, 0, 1, 0, 0, 0, -1, -1,
  0, 0, -1, 0, 0
), ncol = 5)
cv <- C %*% V %*% t(C) ## covariance matrix
dv <- t(1 / sqrt(diag(cv)))
cr <- cv * (t(dv) %*% dv) ## correlation matrix
delta <- rep(0, 5)
Tn <- qmvt(0.95,
  df = df, delta = delta, corr = cr,
  abseps = 0.0001, maxpts = 100000, tail = "both"
)
Tn
```

```
## $quantile
## [1] 2.560887
##
## $f.quantile
```

© 草稿


```
## [1] 2.070871e-07
##
## attr(,"message")
## [1] "Normal Completion"
```

计算多元正态分布的概率，这个例子来自 https://stackoverflow.com/questions/36704081

```
# 模拟一个协方差矩阵
sigma <- as.matrix(read.csv(file = "data/sigma.csv", header = F, sep = ","))
rownames(sigma) <- colnames(sigma)
# matrixcalc::is.symmetric.matrix(sigma) # 判断 sigma 是否为对称的矩阵
# matrixcalc::is.positive.definite(sigma) # 判断 sigma 是否为正定的矩阵
# isTRUE(all.equal(sigma, t(sigma)))
m <- nrow(sigma)
Fn <- pmvnorm(
  lower = rep(-Inf, m), upper = rep(0, m),
  mean = rep(0, m), sigma = sigma
)
Fn
```

`mvrnorm()` 函数来自 **MASS** 包，模拟多元正态分布的样本

```
library(MASS)
n <- 1000 # 样本量
X <- mvrnorm(n, mu = rep(0, 2), Sigma = matrix(c(1, 0.8, 0.8, 1), ncol = 2, byrow = TRUE))
plot(X,
  pch = 20, panel.first = grid(), cex = 1,
  col = densCols(X, colramp = terrain.colors),
  xlab = expression(X[1]), ylab = expression(X[2])
)
points(x =0, y = 0, pch = 3, cex = 2)
```

```
f1 <- kde2d(X[, 1], X[, 2], n = 25)
filled.contour(f1, color.palette = terrain.colors)

library(shape)
persp(f1$z,
  xlab = expression(X[1]), ylab = expression(X[2]),
  zlab = expression(Z),
  col = drapecol(f1$z, col = terrain.colors(20)),
  theta = 30, phi = 20,
  r = 50, d = 0.1, expand = 0.5, ltheta = 90, lphi = 180,
  shade = 0.1, ticktype = "detailed", nticks = 5
)
```

Wishart 分布文献 [Eaton, 2007] 第八章

© 黄湘云

图 21.1: 二维正态分布

(a) 等高线图 (b) 透视图

图 21.2: 二维正态分布

© 基湖石

第二十二章　参数估计

> Jeremy Koster: My students were looking at the estimated varying intercepts for each higher-level group (or the "BLUP's", as some people seem to call them).
>
> Douglas Bates: As Alan James once said, "these values are just like the BLUPs - Best Linear Unbiased Predictors - except that they aren't linear and they aren't unbiased and there is no clear sense in which they are"best", but other than that ..."
>
> — Jeremy Koster and Douglas Bates [1]

22.1　点估计

- 矩估计
- 极大似然估计
- 最小二乘估计
- 同变估计
- 稳健估计

单参数和多参数模型的参数估计，比如指数分布、泊松分布、二项分布、正态分布，线性模型各个估计的推导过程

注意

应当考虑 $(X^\top X)^{-1}$ 不存在的情况，在均方误差最小的意义下，不必要求 β 的估计 $\hat{\beta}$ 满足无偏性的要求，所以介绍岭回归估计 $\hat{\beta}_{ridge}$、压缩估计 $\hat{\beta}_{jse}$、主成分估计 $\hat{\beta}_{pca}$ 和偏最小二乘估计 $\hat{\beta}_{pls}$。相比于 $\hat{\beta}_{pca}$，$\hat{\beta}_{pls}$ 考虑了响应变量的作用。《数理统计引论》第 5 章第 5 节线性估计类从改进 LS 估计出发，牺牲一部分估计的偏差，即采用有偏的估计，达到总体均方误差更小的效果 [陈希孺, 1981] James-Stein 估计可不可以看作一种压缩估计？从它牺牲一部分偏差，获取整体方差的降低来看和上面应该有某种联系

- 昔日因，今日意 讲线性混合效应模型和很多模型之间的联系
- 那些年，我们一起追的 EB James-Stein 估计和岭回归估计的联系
- 统计学习那些事 lasso 和 boosting 之间的联系

[1] https://stat.ethz.ch/pipermail/r-sig-mixed-models/2012q3/018817.html

22.1.1 矩估计

22.1.2 最小二乘估计

谈非线性最小二乘，这段话的意思是非线性模型不要谈 ANOVA 和 R^2 之类的东西

> As one of the developers of the `nls` function I would like to state that the lack of automatic ANOVA, R^2 and $adj.R^2$ from `nls` is a feature, not a bug :-)
>
> — Douglas Bates [2]

> 最小二乘估计是一种非参数估计方法（对数据分布没有假设，只要预测误差达到最小即可），而极大似然估计是一种参数估计方法（观测数据服从带参数的多元分布）

非线性最小二乘估计

```
# Nonlinear least-squares using nlm()
# demo(nlm)

# Helical Valley Function
# 非线性最小二乘

theta <- function(x1, x2) (atan(x2 / x1) + (if (x1 <= 0) pi else 0)) / (2 * pi)
## 更加简洁的表达
theta <- function(x1, x2) atan2(x2, x1) / (2 * pi)
# 目标函数
f <- function(x) {
  f1 <- 10 * (x[3] - 10 * theta(x[1], x[2]))
  f2 <- 10 * (sqrt(x[1]^2 + x[2]^2) - 1)
  f3 <- x[3]
  return(f1^2 + f2^2 + f3^2)
}

## explore surface {at x3 = 0}
x <- seq(-1, 2, length.out = 50)
y <- seq(-1, 1, length.out = 50)
z <- apply(as.matrix(expand.grid(x, y)), 1, function(x) f(c(x, 0)))

contour(x, y, matrix(log10(z), 50, 50))

nlm.f <- nlm(f, c(-1, 0, 0), hessian = TRUE)

points(rbind(nlm.f$estim[1:2]), col = "red", pch = 20)
```

```
### the Rosenbrock banana valley function 香蕉谷函数

fR <- function(x) {
```

[2] https://stat.ethz.ch/pipermail/r-help/2000-August/007778.html

© 统计出版社


```
  x1 <- x[1]
  x2 <- x[2]
  100 * (x2 - x1 * x1)^2 + (1 - x1)^2
}

## explore surface
fx <- function(x) { ## `vectorized' version of fR()
  x1 <- x[, 1]
  x2 <- x[, 2]
  100 * (x2 - x1 * x1)^2 + (1 - x1)^2
}
x <- seq(-2, 2, length.out = 100)
y <- seq(-0.5, 1.5, length.out = 100)
z <- fx(expand.grid(x, y))
op <- par(mfrow = c(2, 1), mar = 0.1 + c(3, 3, 0, 0))
contour(x, y, matrix(log10(z), length(x)))

nlm.f2 <- nlm(fR, c(-1.2, 1), hessian = TRUE)
points(rbind(nlm.f2$estim[1:2]), col = "red", pch = 20)

## Zoom in :
rect(0.9, 0.9, 1.1, 1.1, border = "orange", lwd = 2)
x <- y <- seq(0.9, 1.1, length.out = 100)
z <- fx(expand.grid(x, y))
contour(x, y, matrix(log10(z), length(x)))
mtext("zoomed in")
box(col = "orange")
points(rbind(nlm.f2$estim[1:2]), col = "red", pch = 20)
par(op)
```

```
with(
  nlm.f2,
  stopifnot(
    all.equal(estimate, c(1, 1), tol = 1e-5),
    minimum < 1e-11, abs(gradient) < 1e-6, code %in% 1:2
  )
)
```

```
fg <- function(x) {
  gr <- function(x1, x2) {
    c(-400 * x1 * (x2 - x1 * x1) - 2 * (1 - x1), 200 * (x2 - x1 * x1))
  }
  x1 <- x[1]
  x2 <- x[2]
  structure(100 * (x2 - x1 * x1)^2 + (1 - x1)^2,
    gradient = gr(x1, x2)
```

统计建模 ©

```
  )
}

nfg <- nlm(fg, c(-1.2, 1), hessian = TRUE)
str(nfg)
```

```
with(
  nfg,
  stopifnot(
    minimum < 1e-17, all.equal(estimate, c(1, 1)),
    abs(gradient) < 1e-7, code %in% 1:2
  )
)
```

```
## or use deriv to find the derivatives

fd <- deriv(~ 100 * (x2 - x1 * x1)^2 + (1 - x1)^2, c("x1", "x2"))
fdd <- function(x1, x2) {}
body(fdd) <- fd

nlfd <- nlm(function(x) fdd(x[1], x[2]), c(-1.2, 1), hessian = TRUE)
str(nlfd)
```

```
with(
  nlfd,
  stopifnot(
    minimum < 1e-17, all.equal(estimate, c(1, 1)),
    abs(gradient) < 1e-7, code %in% 1:2
  )
)
```

```
fgh <- function(x) {
  gr <- function(x1, x2) {
    c(-400 * x1 * (x2 - x1 * x1) - 2 * (1 - x1), 200 * (x2 - x1 * x1))
  }
  h <- function(x1, x2) {
    a11 <- 2 - 400 * x2 + 1200 * x1 * x1
    a21 <- -400 * x1
    matrix(c(a11, a21, a21, 200), 2, 2)
  }
  x1 <- x[1]
  x2 <- x[2]
  structure(100 * (x2 - x1 * x1)^2 + (1 - x1)^2,
    gradient = gr(x1, x2),
    hessian  =  h(x1, x2)
  )
}
```

© 王湘


```
nlfgh <- nlm(fgh, c(-1.2, 1), hessian = TRUE)

str(nlfgh)
```

```
## NB: This did _NOT_ converge for R version <= 3.4.0
with(
  nlfgh,
  stopifnot(
    minimum < 1e-15, # see 1.13e-17 .. slightly worse than above
    all.equal(estimate, c(1, 1), tol = 9e-9), # see 1.236e-9
    abs(gradient) < 7e-7, code %in% 1:2
  )
) # g[1] = 1.3e-7
```

22.1.3 极大似然估计

教材简短一句话，这里面有很多信息值得发散，一个数学家提出了统计学领域极其重要的一个核心思想，他是在研究什么的时候提出了这个想法，为什么后来没有得到重视，虽然这可能有点离题，但是对于读者可能有很多别的启迪。整整 100 年以后，Fisher 又是怎么提出这一思想的呢？他做了什么使得这个思想被广泛接受和应用？

统计决策理论，任何统计推断都应该依赖损失函数，而极大似然估计未曾考虑到，这是它的局限性。Lasso 和贝叶斯先验的关系，和损失函数的关系

是最大似然估计还是极大似然估计？当然是极大似然估计，如果有人告诉你是最大似然估计那一定是假的，这两个概念归根结底是极值和最值得区别

书本定义和性质，在后续章节介绍

介绍线性模型为何引入 REML 减少偏差

极大似然估计是费舍尔提出来的

- 边际似然 Marginal Likelihood
- 条件似然 conditional lkelihood
- 完全似然 complete Likelihood
- 层次似然 Hierarchical likelihood
- 部分似然 partial likelihood
- 剖面似然 Profile Likelihood
- 限制似然 Restricted Likelihood
- 惩罚/边际拟似然 (PQL/MQL) Penalized Quasi-Likelihood/Marginal Quasi-Likelihood
- 分布边际分布条件分布
- 似然边际似然条件似然
- 极大似然估计 Maximum likelihood 简称 ML
- 限制极大似然 Restricted Maximum likelihood, 简称 REML
- 惩罚拟似然 Penalized Quasi-Likelihood, 简称 PQL 和边际拟似然 Marginal Quasi-Likelihood, 简称 MQL，Profile Maximal Likelihood, 简称 PML

拟似然估计 极大似然估计 似然函数

Penalized maximum likelihood estimates are calculated using optimization methods such as the limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS).

BFGS 拟牛顿法和采样器 https://bookdown.org/rdpeng/advstatcomp

22.2 区间估计

22.2.1 正态分布

正态分布 $\mathcal{N}(\mu, \sigma^2)$，$\sigma^2$ 未知，关于参数 μ 的置信水平为 $1-\alpha$ 的区间估计

1. 构造统计量 $t = \frac{\bar{x}-\mu}{s/\sqrt{n}} \sim t(n-1)$
2. 参数 μ 的 $1-\alpha$ 置信区间为

$$\bar{x} \pm t_{1-\alpha/2}(n-1)s/\sqrt{n}$$

其中，$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2$ 是 σ^2 的无偏估计。若取 $\alpha = 0.05$，则置信水平 $1-\alpha = 0.95$。

```
set.seed(2020) # 为了可重复，设置随机数种子
mu_ci <- function(alpha = 0.05, n = 100, mu = 4) {
  x <- rnorm(n = n, mean = mu, sd = 1)
  x_bar <- mean(x)
  d <- qt(p = 1 - alpha / 2, df = n - 1, lower.tail = TRUE) * var(x) / sqrt(n)
  c(mu = mu, lower = x_bar - d, upper = x_bar + d)
}
# 重抽样 100 次，获得 100 个置信区间
dat <- t(replicate(n = 100, mu_ci(alpha = 0.05, n = 100, mu = 4)))
dat <- transform(dat, idx = 1:100, cover = ifelse(mu >= lower & mu <= upper, TRUE, FALSE))
```

真实的参数值 $\mu = 4$，重抽样 100 次，覆盖真值的次数为 97 次，覆盖概率为 0.97

```
# 覆盖概率
mean(dat$cover)
```

```
## [1] 0.97
```

```
library(ggplot2)
ggplot() +
  geom_segment(data = dat, aes(
    x = idx, xend = idx,
    y = lower, yend = upper, color = cover
  )) +
  geom_hline(yintercept = 4) +
  theme_minimal() +
  labs(x = "", y = "")
```

方差 σ^2 已知的情况下，标准正态分布 $N(\mu, \sigma^2), \mu = 0, \sigma^2 = 1$ 的参数 μ 的区间估计和覆盖概率 https://yihui.org/animation/example/conf-int/

© 黄湘云

图 22.1: μ 的置信水平为 0.95 的置信区间

22.2.2 0-1 分布

设 0-1 分布 $B(1,p)$ 的成功概率 $p=0.95$，假定是抛硬币的场景，成功概率对应正面朝上的概率为 0.95。一次实验，重复抛 10 次，有两次正面朝上。现在要根据这次实验结果估计成功概率 p 的值，及其置信区间

```
# 卡方近似
prop.test(x = 2, n = 10, p = 0.95, conf.level = 0.95, correct = TRUE)
```

```
## Warning in prop.test(x = 2, n = 10, p = 0.95, conf.level = 0.95, correct =
## TRUE): Chi-squared approximation may be incorrect

##
##  1-sample proportions test with continuity correction
##
## data:  2 out of 10, null probability 0.95
## X-squared = 103.16, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.95
## 95 percent confidence interval:
##  0.03542694 0.55781858
## sample estimates:
##   p
## 0.2
```

© 基础统计

```r
# 二项精确估计
binom.test(x = 2, n = 10, p = 0.95, conf.level = 0.95)
```

```
##
##  Exact binomial test
##
## data:  2 and 10
## number of successes = 2, number of trials = 10, p-value = 1.605e-09
## alternative hypothesis: true probability of success is not equal to 0.95
## 95 percent confidence interval:
##  0.02521073 0.55609546
## sample estimates:
## probability of success
##                    0.2
```

可知，在置信水平都是 0.95 的情况下，带连续矫正的单样本比例检验方法获得的区间估计是 (0.0354, 0.5578)，区间长度 0.5224。精确二项检验方法获得的区间估计是 (0.0252, 0.5560)，区间长度 0.5308。

从二项分布 $B(30, 0.2)$ 中随机抽取一个样本，为可重复记，设置随机数种子为 2020

```r
set.seed(2020)
rbinom(1, size = 30, prob = 0.2)
```

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

得到样本观测值为 7，

```r
7 - qnorm(1 - 0.95 / 2) * sqrt(0.2 * 0.8 / 30) # 6.995
```

```
## [1] 6.995421
```

```r
7 + qnorm(1 - 0.95 / 2) * sqrt(0.2 * 0.8 / 30) # 7.0045
```

```
## [1] 7.004579
```

样本观测值 7 对应的参数 p 的区间估计，如下

```r
prop.test(x = 7, n = 30, p = 0.2, conf.level = 0.95, correct = TRUE)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  7 out of 30, null probability 0.2
## X-squared = 0.052083, df = 1, p-value = 0.8195
## alternative hypothesis: true p is not equal to 0.2
## 95 percent confidence interval:
##  0.1063502 0.4270023
## sample estimates:
##         p
## 0.2333333
```

随机变量 X 服从二项分布 $B(30, 0.2)$，则概率值 $P(x \leq 7) = 0.7607$

© 王建平


```
pbinom(7, size = 30, prob = 0.2, lower.tail = TRUE)
```

```
## [1] 0.7607906
```

已知概率值为 0.95，即 $P(x \le m) = 0.95$ 且 $X \sim B(30, 0.2)$，现在计算 m 的值，即求下分位点，为 10

```
qbinom(p = 0.95, size = 30, prob = 0.2, lower.tail = TRUE)
```

```
## [1] 10
```

提示

二项分布的特点，主要用于计算期望，概率 $P\{C_1 + 1 \le x \le C_2 - 1\}$

$$\sum_{x=C_1+1}^{C_2-1} x\binom{n}{x}p^x(1-p)^{n-x} = np\sum_{x=C_1+1}^{C_2-1}\binom{n-1}{x-1}p^{x-1}(1-p)^{(n-1)-(x-1)}$$

```
n = 30
c2 = 20
c1 = 10
p = 0.2
n * p * (pbinom(q = c2 - 2, size = n - 1, prob = p) - pbinom(q = c1 - 1, size = n - 1, prob = p))
```

```
## [1] 0.2955803
```

22.2.3 置信区间和信仰区间

计算置信区间的覆盖概率 binom

二项分布的参数估计，包括点估计和区间估计 [Clopper and Pearson, 1934]

给定样本量 n = 10 0-1 分布成功概率 p 分别取 0.1,0.2,...,1 置信度为 95% 观测到 x 取 1,2,3,..,10 时估计 p 的上下限

```
set.seed(2019)
x <- rbinom(n = 1, size = 10, prob = 0.1) # 结果解读
```

抛掷硬币 10 次，观测到 2 次正面朝上，估计正面朝上的概率

观测到正面朝上 2 次此时请以 95% 的信心给出 p 的区间 (p_low, p_up)

绘制曲线 p 关于 x 的曲线

```
set.seed(2019)
p <- seq(from = 0, to = 1, length.out = 11)
# 成功概率 总体参数 p 值
sapply(rep(p, each = 10), rbinom, n = 1, size = 10)
```

```
##   [1]  0  0  0  0  0  0  0  0  0  0  2  1  0  1  0  0  2  0  0  1  3  2  1  1  3
##  [26]  2  0  3  1  2  3  3  5  2  1  0  3  5  3  3  5  1  5  3  1  2  3  4  1  5
##  [51]  5  4  7  6  6  5  7  7  4  4  7  8  9  7  6  7  2  4  8  8  8  8  6  8  6
##  [76]  4  8  9  6  7  9  9  9  9  8  4  9  8  9  7 10  8  7 10  9 10  9  9  8 10
## [101] 10 10 10 10 10 10 10 10 10 10
```

©黄湘云

计算每一次抽样获得的上下限

Clopper-Pearson 方法，即求和搜索，在保持累积概率

$$B(x,n;n,p)=\sum_{r=x}^{n}\binom{n}{r}p^r(1-p)^{n-r}=\alpha/2$$

其中 n 表示试验次数，这里是 10，p 是未知待求，已知 $\alpha=0.05$，而 $1-\alpha$ 表示置信水平，意思是说对于我给出的区间估计，长期来看，我有 95% 的信心认为，真实值 p 会落在此区间内。

对上尾部从 x 到 n 求和，计算 p，对每一个 x 都能计算出一个 p，根据二项分布的对称性，区间 $[0,x]$ 和 $[x,n]$ 的累积概率是相同的，各占 $\alpha/2$

```
# 精确计算二项分布检验的 p
# 调用符号计算
# x = 7
fun <- function(p, r = 8, n = 10) {
  choose(n, n-2)*p^r*(1-p)^(n-r) + choose(n, n-1)*p^(n-1)*(1-p) + choose(n, n)*p^n - 0.025
}
uniroot(fun, lower = 0, upper = 1)
```

```
## $root
## [1] 0.4439038
##
## $f.root
## [1] -2.707352e-07
##
## $iter
## [1] 9
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.103516e-05
```

```
# x = 8
fun <- function(p) {
  45*p^8*(1-p)^2 + 10*p^9*(1-p) + p^10 - 0.025
}
uniroot(fun, lower = 0, upper = 1)
```

```
## $root
## [1] 0.4439038
##
## $f.root
## [1] -2.707352e-07
##
## $iter
```

汪荣鹏©著


```
## [1] 9
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.103516e-05
```

```r
# x = 9
fun <- function(x) {
  9 * x^10 - 10 * x^9 + 0.025
}
# 0.555 计算下限
uniroot(fun, lower = 0, upper = 1)
```

```
## $root
## [1] 0.5549828
##
## $f.root
## [1] 3.773379e-07
##
## $iter
## [1] 10
##
## $init.it
## [1] NA
##
## $estim.prec
## [1] 6.462529e-05
```

```r
# x = 10
fun <- function(x) {
  x^10 - 0.025
}
# 0.691
uniroot(fun, lower = 0, upper = 1)
```

```
## $root
## [1] 0.6914996
##
## $f.root
## [1] -1.194136e-06
##
## $iter
## [1] 9
##
## $init.it
```

© 王湘黔

```
## [1] NA
##
## $estim.prec
## [1] 6.103516e-05
```

累积二项概率

找到最小的 p 使得其等于 9

```r
# 已知概率求上分位点

# 等于
qbinom(0.025, size = 10, prob = 0.565, lower.tail = F)
```

```
## [1] 9
```

找到使得函数为 0 的 p 中最小的那个，找到所有的根，然后取最小的那个

```r
fun <- function(p, r = 9) qbinom(0.025, size = 10, prob = p, lower.tail = F) - r
# 计算每个 x 对应的 p
(p <- sapply(1:10, function(x) uniroot(fun, lower = 0, upper = 1, r = x)$root))
```

```
##  [1] 0.01666667 0.05333333 0.09375000 0.16000000 0.25000000 0.30000000
##  [7] 0.35000000 0.53333333 0.67500000 1.00000000
```

```r
plot(x = 1:10, y = p)
```

© 黄湘云

```
# 二项检验 菱形置信带
set.seed(2019)

dat <- replicate(10^3, expr = {
  x = sample(0:1, size = 10, replace = TRUE, prob = c(0.8, 0.2))
  sum(x)/10
})

# 成功概率 p = 0.2 每个样本量 10
dat <- rbinom(n = 10^3, size = 10, prob = 0.2)/10
table(dat)
```

```
## dat
##   0 0.1 0.2 0.3 0.4 0.5 0.6 0.7
## 119 274 304 180  82  30  10   1
```

© 黄湘云

```
# 分布图 y 轴是密度
# right = TRUE 区间形式 (a,b] 左开右闭
hist(dat, probability = T, breaks = seq(from = -0.1, to = 1, by = 0.1))
```

```
# 0.2^10 左闭右开区间
hist(dat, probability = T, breaks = seq(from = 0, to = 1.1, by = 0.1),
     right = FALSE, xlim = c(0, 1.1))
```

© 黄湘云

```
# 分布
library(ggplot2)
library(magrittr)
# 这个图里面会不会隐含什么信息，分布是怎样的？
# 二项展开有关系吗
dat1 <- as.data.frame(table(dat))


ggplot(data = dat1, aes(x = dat, y = Freq)) +
  geom_col()
```

© 黄湘云

```
ggplot(as.data.frame(dat), aes(x = dat)) +
  geom_histogram(bins = 12)
```

© 黄湘云

22.3 最小角回归

1. Efron, Bradley and Hastie, Trevor and Johnstone, Iain and Tibshirani, Robert. 2004. Least angle regression. The Annals of Statistics. 32(2): 407–499. https://doi.org/10.1214/009053604000000067.

方差缩减技术，修偏技术

22.4 刀切法

1. Efron, B. 1979. Bootstrap Methods: Another Look at the Jackknife. The Annals of Statistics. 7(1):1–26. https://doi.org/10.1214/aos/1176344552

22.5 重抽样

22.6 Delta 方法

© 吉祥图书

第二十三章　假设检验

The Earth is Round ($p < 0.05$)

— Jacob Cohen [Cohen, 1994]

```
x = seq(from = -4, to = 8, length.out = 193)
y1 = dnorm(x, mean = 3, sd = 1)
```

© 数据科学


```
y2 = dnorm(x, mean = 2, sd = 1.5)
library(magrittr)
hline <- function(y = 0, color = "red") {
  list(
    type = "line",
    x0 = 0,
    x1 = 1,
    xref = "paper",
    y0 = y,
    y1 = y,
    line = list(color = color, dash = 'dash', width = .5)
  )
}

vline <- function(x = 0, color = "red") {
  list(
    type = "line",
    x0 = x,
    x1 = x,
    yref = "paper",
    y0 = 0,
    y1 = 1,
    line = list(color = color, dash = 'dash', width = .5)
  )
}

plotly::plot_ly(
  x = x, y = y1,
  type = "scatter", mode = "lines",
  fill = "tozeroy", fillcolor = "rgba(92, 184, 92, 0.2)",
  text = ~ paste0(
    "x: ", x, "<br>",
    "y: ", round(y1, 3), "<br>"
  ),
  hoverinfo = "text",
  name = plotly::TeX("\\mathcal{N}(3,1^2)"),
  line = list(shape = "spline", color = "#5CB85C")
) %>%
  plotly::add_trace(
    x = x, y = y2,
    type = "scatter", mode = "lines",
    fill = "tozeroy", fillcolor = "rgba(91, 192, 222, 0.2)",
    text = ~ paste0(
      "x: ", x, "<br>",
      "y: ", round(y2, 3), "<br>"
```

© 基础统计

```
  ),
  hoverinfo = "text",
  name = plotly::TeX("\\mathcal{N}(2, 1.5^2)"),
  line = list(shape = "spline", color = "#5BC0DE")
) %>%
plotly::add_segments(
  x = 2,
  y = 0.28,
  xend = 3,
  yend = 0.28,
  line = list(color = "black"),
  showlegend = F
) %>%
plotly::add_annotations(
  x = 2.5, y = 0.3,
  showarrow = F, font = list(size = 24),
  text = plotly::TeX("d")
) %>%
plotly::add_annotations(
  x = 0, y = 1 / sqrt(2 * pi),
  font = list(size = 100), showarrow = F,
  text = plotly::TeX("\\frac{1}{\\sqrt{2\\pi}}")
) %>%
plotly::add_annotations(
  x = 0, y = 1 / (1.5 * sqrt(2 * pi)),
  font = list(size = 100), showarrow = F,
  text = plotly::TeX("\\frac{1}{1.5\\sqrt{2\\pi}}")
) %>%
plotly::layout(
  shapes = list(
    hline(y = 1 / sqrt(2 * pi), color = "#F27B0C"),
    hline(y = 1 / (1.5 * sqrt(2 * pi)), color = "#F27B0C"),
    vline(x = 3, color = "#F27B0C"),
    vline(x = 2, color = "#F27B0C")
  ),
  xaxis = list(showgrid = F, title = plotly::TeX("x")),
  yaxis = list(showgrid = F, title = plotly::TeX("f(x)")),
  legend = list(x = 0.8, y = 1, orientation = "v")
) %>%
plotly::config(displayModeBar = FALSE, mathjax = "cdn")
```

R. A. Fisher 将抽样分布、参数估计和假设检验列为统计推断的三个中心内容，可见假设检验的重要地位

呈现常见检验的公式，将手写代码和 R 内置函数计算结果进行比较，每一组原假设和备择假设要说明对应的 R 函数和及其参数设置，尽量理论和代码并重，最后结合实际的数据予以解释说明。

© 王建湘

Jacob Cohen 实际谈的是更加深刻的问题。开篇介绍为什么需要假设检验，做检验和不做检验有什么区别？杨灿老师在讨论帖提出检验的作用和实际应用问题

有了均值和方差，为什么还要位置参数和尺度参数？为了更一般地描述问题，扩展范围。

Summary and Analysis of Extension Program Evaluation in R 介绍了各类假设检验方法

The IQUIT R video series

假设检验，实验 A 和 B 的区分度适用于在线服务的 A/B 测试方法论 http://www.fengjunchen.com/

统计分布的检验

从心理学和可视化的角度谈 Cohen's d

Bootstrap 方法和置换/秩检验（Permutation Test）的入门读物

非平衡的 A/B 试验设计 Optimal unbalanced design for A/B test

Wilcoxon (WMWU) test sensitivity 检验的灵敏性

从抛硬币到 P 值和统计显著性

一分钟学会 A/B 测试

rstatix 包提供了一个简明的管道友好的框架，和 tidyverse 的设计哲学保持一致，支持常见的统计检验，如 T 检验，Wilcoxon 检验，方差分析，Kruskal-Wallis 检验，相关性分析，并将结果整理成干净的数据框形式，以方便可视化。

https://github.com/pieces201020/AB-Test-Sample-Size-Calculator 又一个样本量计算器

23.1 Ansari-Bradley 检验 `ansari.test`

Ansari-Bradley 检验目的是检验两样本的尺度参数是否有显著性差异

尺度参数可以理解为方差 σ^2

位置参数可以理解为均值 μ

```
usage(ansari.test)
ansari.test(x, ...)
usage("ansari.test.default")
## Default S3 method:
ansari.test(x, y, alternative = c("two.sided", "less", "greater"), exact = NULL,
    conf.int = FALSE, conf.level = 0.95, ...)
usage("ansari.test.formula")
## S3 method for class 'formula'
ansari.test(formula, data, subset, na.action, ...)
```

23.2 Bartlett 检验 `bartlett.test`

`ansari.test` 和 `mood.test` 是基于秩的两样本尺度参数显著性差异检验，是非参数检验

©黄文涛

Bartlett 检验：检验各个组的方差是否有显著性差异，即方差齐性检验。

var.test 和 bartlett.test 都属于参数检验，用于检验方差齐性问题，前者考虑正态总体下方差齐性检验，后者没有对总体的分布形式做限定。

```
usage(bartlett.test)
bartlett.test(x, ...)
usage("bartlett.test.default")
## Default S3 method:
bartlett.test(x, g, ...)
usage("bartlett.test.formula")
## S3 method for class 'formula'
bartlett.test(formula, data, subset, na.action, ...)
```

23.3 二项检验 binom.test

比例 p 的检验，做 n 次独立试验，样本 $X_1, \ldots, X_n \sim b(1, p)$，事件发生的总次数 $\sum_{i=1}^{n} X_i$

函数 binom.test 用来检验伯努利试验中成功概率 p 和给定概率 p_0 的关系，属于精确检验。

编程手动实现一个，再调用函数计算，比较结果

```
# 模拟一组样本
x <- sample(x = c(0, 1), size = 100, replace = TRUE, prob = c(0.8, 0.2))
```

二项分布中成功概率的检验

```
binom.test(sum(x), n = 100, p = 0.5)
```

```
##
##  Exact binomial test
##
## data:  sum(x) and 100
## number of successes = 16, number of trials = 100, p-value = 2.606e-12
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.09431029 0.24678760
## sample estimates:
## probability of success
##                   0.16
```

检验成功概率 p 是否等于 0.5，P 值 6.148×10^{-11} 结论是拒绝原假设

```
binom.test(sum(x), n = 100, p = 0.2)
```

```
##
##  Exact binomial test
##
## data:  sum(x) and 100
## number of successes = 16, number of trials = 100, p-value = 0.3814
```

© 基础统计


```
## alternative hypothesis: true probability of success is not equal to 0.2
## 95 percent confidence interval:
##  0.09431029 0.24678760
## sample estimates:
## probability of success
##                   0.16
```

检验成功概率 p 是否等于 0.2，P 值 0.7081 结论是不能拒绝原假设

二项检验 [Clopper and Pearson, 1934]

```
usage(binom.test)
```

```
binom.test(x, n, p = 0.5, alternative = c("two.sided", "less", "greater"),
    conf.level = 0.95)
```

23.4 时间序列独立性检验 Box.test

计算 Box-Pierce 或 Ljung-Box 检验统计量来检查给定时间序列的独立性假设。

```
usage(Box.test)
```

```
Box.test(x, lag = 1, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)
```

23.5 皮尔逊卡方检验 chisq.test

用于计数数据的皮尔逊卡方检验：列联表独立性检验和拟合优度检验

chisq.test χ^2 检验：列联表检验和拟合优度检验

```
usage(chisq.test)
```

```
chisq.test(x, y = NULL, correct = TRUE, p = rep(1/length(x), length(x)),
    rescale.p = FALSE, simulate.p.value = FALSE, B = 2000)
```

23.6 费舍尔精确检验 fisher.test

固定边际的情况下，检验列联表行和列之间的独立性

```
usage(fisher.test)
```

```
fisher.test(x, y = NULL, workspace = 2e+05, hybrid = FALSE,
    hybridPars = c(expect = 5, percent = 80, Emin = 1), control = list(),
    or = 1, alternative = "two.sided", conf.int = TRUE, conf.level = 0.95,
    simulate.p.value = FALSE, B = 2000)
```

© 基础

23.7 方差齐性检验 fligner.test

Fligner-Killeen (中位数) 检验各个组的样本方差是不是一致的，也是方差齐性检验

```
usage(fligner.test)
fligner.test(x, ...)
usage("fligner.test.default")
## Default S3 method:
fligner.test(x, g, ...)
usage("fligner.test.formula")
## S3 method for class 'formula'
fligner.test(formula, data, subset, na.action, ...)
```

23.8 Friedman 秩和检验 friedman.test

Friedman 秩和检验

Performs a Friedman rank sum test with unreplicated blocked data.

```
usage(friedman.test)
friedman.test(y, ...)
usage("friedman.test.default")
## Default S3 method:
friedman.test(y, groups, blocks, ...)
usage("friedman.test.formula")
## S3 method for class 'formula'
friedman.test(formula, data, subset, na.action, ...)
```

23.9 Kruskal-Wallis 秩和检验 kruskal.test

Kruskal-Wallis 秩和检验

```
usage(kruskal.test)
kruskal.test(x, ...)
usage("kruskal.test.default")
## Default S3 method:
kruskal.test(x, g, ...)
usage("kruskal.test.formula")
## S3 method for class 'formula'
kruskal.test(formula, data, subset, na.action, ...)
```

© 基础

23.10 同分布检验 ks.test

Lilliefors 检验[1] 和单样本的 ks 检验的关系

> As to whether you can do a **Lilliefors test** for several groups, that depends entirely on your ability to understand what the underlying question would be (see Adams D 1979).
>
> — Knut M. Wittkowski[2]

Kolmogorov-Smirnov 检验：单样本或两样本的同分布检验

```
usage(ks.test)
```

```
ks.test(x, y, ..., alternative = c("two.sided", "less", "greater"),
    exact = NULL)
```

23.11 Cochran-Mantel-Haenszel 卡方检验 mantelhaen.test

用于计数数据的 Cochran-Mantel-Haenszel 卡方检验

Performs a Cochran-Mantel-Haenszel chi-squared test of the null that two nominal variables are conditionally independent in each stratum, assuming that there is no three-way interaction.

```
usage(mantelhaen.test)
```

```
mantelhaen.test(x, y = NULL, z = NULL,
    alternative = c("two.sided", "less", "greater"), correct = TRUE,
    exact = FALSE, conf.level = 0.95)
```

23.12 Mauchly 球形检验 mauchly.test

检验：Wishart 分布的协方差矩阵是否正比于给定的矩阵

Mauchly's Test of Sphericity

Tests whether a Wishart-distributed covariance matrix (or transformation thereof) is proportional to a given matrix.

```
usage(mauchly.test)
mauchly.test(object, ...)
usage("mauchly.test.mlm")
## S3 method for class 'mlm'
mauchly.test(object, ...)
usage("mauchly.test.SSD")
## S3 method for class 'SSD'
mauchly.test(object, Sigma = diag(nrow = p), T = Thin.row(proj(M) - proj(X)),
    M = diag(nrow = p), X = ~0, idata = data.frame(index = seq_len(p)), ...)
```

[1] https://personal.utdallas.edu/~herve/Abdi-Lillie2007-pretty.pdf

[2] https://stat.ethz.ch/pipermail/r-help/2004-February/045597.html

©黄文海

23.13 McNemar 卡方检验 mcnemar.test

两种统计量的比较参看谢益辉的博文 渐近理想国：McNemar 检验的两种统计量

用于计数数据的 McNemar's 卡方检验

McNemar's χ^2 检验：检验二维列联表行和列的对称性

```
usage(mcnemar.test)
```

```
mcnemar.test(x, y = NULL, correct = TRUE)
```

23.14 Mood 方差检验 mood.test

检验方差

Mood's 两样本检验：检验两样本尺度参数之间的差异性

```
usage(mood.test)
mood.test(x, ...)
usage("mood.test.default")
## Default S3 method:
mood.test(x, y, alternative = c("two.sided", "less", "greater"), ...)
usage("mood.test.formula")
## S3 method for class 'formula'
mood.test(formula, data, subset, na.action, ...)
```

23.15 单因素多重比较 oneway.test

单因素方差分析，各个组的方差不一定相同，检验两个及以上来自正态分布的样本是否有相同的均值?

```
usage(oneway.test)
```

```
oneway.test(formula, data, subset, na.action, var.equal = FALSE)
```

```
## 假定方差不等
oneway.test(extra ~ group, data = sleep)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  extra and group
## F = 3.4626, num df = 1.000, denom df = 17.776, p-value = 0.07939
```

```
## 假定方差相等
oneway.test(extra ~ group, data = sleep, var.equal = TRUE)
```

```
##
##  One-way analysis of means
```

© 基础统计


```
##
## data:  extra and group
## F = 3.4626, num df = 1, denom df = 18, p-value = 0.07919
```

```
## 和线性回归结果一样
anova(lm(extra ~ group, data = sleep))
```

```
## Analysis of Variance Table
##
## Response: extra
##           Df Sum Sq Mean Sq F value  Pr(>F)
## group      1 12.482 12.4820  3.4626 0.07919 .
## Residuals 18 64.886  3.6048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

CO2 数据

```
# coplot(uptake ~ conc | Plant, data = CO2, show.given = FALSE, type = "b")
# levels(CO2$Plant) # Plant 是有序的
library(ggplot2)
library(patchwork)
p1 <- ggplot(data = CO2, aes(x = conc, y = uptake)) +
  geom_point(aes(color = Treatment)) +
  geom_line(aes(color = Treatment)) +
  facet_wrap(~Plant, ncol = 4, dir = "v")
p2 <- ggplot(data = CO2, aes(x = conc, y = uptake)) +
  geom_point(aes(color = Type)) +
  geom_line(aes(color = Type)) +
  facet_wrap(~Plant, ncol = 4, dir = "v")
p1 / p2
```

23.16 配对样本的检验

配对样本和单样本的等价转化

23.16.1 配对比例检验 pairwise.prop.test

配对数据的比例检验

Pairwise comparisons for proportions

Calculate pairwise comparisons between pairs of proportions with correction for multiple testing

```
usage(pairwise.prop.test)
```

```
pairwise.prop.test(x, n, p.adjust.method = p.adjust.methods, ...)
```

© 黄湘云

图 23.1: 草类植物吸收 CO2

© 草稿

23.16.2 配对 t 检验 pairwise.t.test

Calculate pairwise comparisons between group levels with corrections for multiple testing

```
usage(pairwise.t.test)
```

```
pairwise.t.test(x, g, p.adjust.method = p.adjust.methods, pool.sd = !paired,
    paired = FALSE, alternative = c("two.sided", "less", "greater"), ...)
```

谢益辉以配对组 t 检验谈 Cohen's d

```
pairwise.t.test(x = sleep$extra, g = sleep$group, paired = T)
```

```
##
##  Pairwise comparisons using paired t tests
##
## data:  sleep$extra and sleep$group
##
##   1
## 2 0.0028
##
## P value adjustment method: holm
```

成对的 t 检验

23.16.3 配对 Wilcoxon 检验 pairwise.wilcox.test

Pairwise Wilcoxon Rank Sum Tests 配对的 Wilcoxon 秩和检验

Calculate pairwise comparisons between group levels with corrections for multiple testing.

```
usage(pairwise.wilcox.test)
```

```
pairwise.wilcox.test(x, g, p.adjust.method = p.adjust.methods, paired = FALSE,
    ...)
```

23.16.4 配对样本相关性检验 cor.test

配对样本的相关性检验：Pearson's 相关系数

Test for association between paired samples, using one of Pearson's product moment correlation coefficient,

Kendall's τ 检验或者 Spearman's ρ 检验.

```
usage(cor.test)
```

```
cor.test(x, ...)
```

- `Kendall::Kendall` [McLeod, 2011]
- `SuppDists::pKendall` 和 `SuppDists::pSpearman` [Wheeler, 2020]
- `pspearman::spearman.test` [Savicky, 2014]

© 基础统计

23.17 精确泊松检验 poisson.test

泊松分布是 1837 年由法国数学家泊松 (Poisson, 1781-1840) 首次提出

泊松分布的参数 $\lambda(>0)$ 的精确检验

Performs an exact test of a simple null hypothesis about the rate parameter in Poisson distribution, or for the ratio between two rate parameters. 适用于单样本和两样本

```
usage(poisson.test)
```

```
poisson.test(x, T = 1, r = 1, alternative = c("two.sided", "less", "greater"),
    conf.level = 0.95)
```

23.18 单位根检验 PP.test

时间序列平稳性检验

Phillips-Perron 的单位根检验

Computes the Phillips-Perron test for the null hypothesis that x has a unit root against a stationary alternative.

```
usage(PP.test)
```

```
PP.test(x, lshort = TRUE)
```

23.19 比例检验 prop.test

函数 `prop.test` 用来检验两组或多组二项分布的成功概率（比例）是否相等，或等于给定的值。近似检验

```
usage(prop.test)
```

```
prop.test(x, n, p = NULL, alternative = c("two.sided", "less", "greater"),
    conf.level = 0.95, correct = TRUE)
```

设随机变量 X 服从参数为 p 的二项分布 $b(n,p)$，Y 服从参数为 θ 的二项分布 $b(m,\theta)$，m,n 都假定为较大的正整数，检验如下问题

$$H_0: P_A \geq P_B \quad vs. \quad H_1: P_A < P_B$$

根据中心极限定理

$$\frac{\bar{X}-\bar{Y}}{\sqrt{\frac{p(1-p)}{n}+\frac{\theta(1-\theta)}{m}}}$$

近似服从标准正态分布 $N(0,1)$。如果用矩估计 $\bar{X}$ 和 $\bar{Y}$ 分别替代总体参数 p 和 θ，构造检验统计量

$$T=\frac{\bar{X}-\bar{Y}}{\sqrt{\frac{\bar{X}(1-\bar{X})}{n}+\frac{\bar{Y}(1-\bar{Y})}{m}}}$$

© 黄湘云

根据 Slutsky 定理，检验统计量 T 近似服从标准正态分布，当 T 偏大时，拒绝 H_0。该方法的优势在于当 n, m 比较大时，二项分布比较复杂，无法建立统计表，利用标准正态分布表来给出检验所需要的临界值，简便易行！

当 p 和 θ 都比较小，上述方法检验效果不好，原因在于由中心极限定理对 $\bar{X}$ 和 $\bar{Y}$ 的正态分布近似效果不好，或者间接地导致 $\bar{X}-\bar{Y}$ 的方差偏小，进而 T 的分辨都不好，而且当 p, θ 很接近 1 时，上述现象也会产生！

下面介绍新的解决办法

上面的检验问题等价于

$$H_0: \frac{P_A}{P_B} \geq 1 \quad vs. \quad H_1: \frac{P_A}{P_B} < 1$$

引入检验统计量

$$T^\star = \frac{\bar{X}}{\bar{Y}}$$

同样由 Slutsky 定理和中心极限定理可知，$\bar{X}/\bar{Y}$ 近似服从正态分布 $N(1, \frac{1-\theta}{m\theta})$

当 $(T^\star - 1)/\hat{\sigma}$ 偏大时接受 H_0，临界值可通过 $N(0, \hat{\sigma}^2)$ 分布表计算得到，$\hat{\sigma}^2$ 是对 $\frac{1-\theta}{m\theta}$ 的估计，比如取 $\hat{\sigma}^2 = \frac{1-\bar{Y}}{m} \cdot \frac{1}{\bar{Y}}$ 或取 $\hat{\sigma}^2 = \frac{1-\bar{Y}}{m} \cdot \frac{1}{\bar{X}}$

由于渐近方差形如 $\frac{1-\theta}{m\theta}$，因而在 θ 较小，渐近方差较大，克服了之前 $\bar{X}-\bar{Y}$ 的方差较小的问题

p, θ 很接近 1 时，我们取检验统计量

$$T^{\star\star} = \frac{1-\bar{Y}}{1-\bar{X}}$$

结论和 $T^\star$ 类似，当 $T^{\star\star}$ 偏大时，拒绝 H_0。

两个二项总体成功概率的比较 [宋泽熙, 2011]

23.19.1 两个独立二项总体等价性检验

关于比例的检验问题

$$H_0: P_A = P_B \quad vs. \quad H_1: P_A > P_B \tag{23.1}$$

$$H_0: P_A = P_B \quad vs. \quad H_1: P_A < P_B \tag{23.2}$$

H_0 成立的情况下，暗示着两个样本来自同一总体。在这种假设设置下，拒绝原假设是不是意味着接受备择假设？如何判断样本点会落在哪个拒绝域内呢？

2009 年东南大学韦博成教授将两个独立二项总体的等价性检验应用于《红楼梦》前 80 回与后 40 回某些文风差异的统计分析 [韦博成, 2009]

©李漫漫

23.19.2 不同页面的点击率问题

CTR：点击率 Click Ratio

矩阵 x 第一行表示页面 A 的点击情况，即 1000 次展示有 500 次点击，第二行表示页面 B 的点击情况，即 100 次展示有 80 次点击。通过统计检验的方式比较页面 A 和 B 的点击率哪个更好？

	S	F
A	500	500
B	80	20

```
(x <- matrix(c(500, 80, 500, 20), nrow = 2, ncol = 2, byrow = FALSE))
```

```
##      [,1] [,2]
## [1,]  500  500
## [2,]   80   20
```

```
# 等价于 prop.test(x, alternative = "two.sided", correct = TRUE)
prop.test(x) # 默认参数设置情形是双边检验
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  x
## X-squared = 31.632, df = 1, p-value = 1.863e-08
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.3898012 -0.2101988
## sample estimates:
## prop 1 prop 2
##    0.5    0.8
```

默认的假设检验问题

$$H_0: P_A = P_B \quad vs. \quad H_1: P_A \neq P_B$$

输出结果中 alternative hypothesis 表示备择假设，参数 `alternative` 指定备择假设的形式

备择假设 $P_A < P_B$ 对应

```
prop.test(x, alternative = "less")
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  x
## X-squared = 31.632, df = 1, p-value = 9.315e-09
## alternative hypothesis: less
## 95 percent confidence interval:
```

© 黄湘云


```
##  -1.0000000 -0.2237522
## sample estimates:
## prop 1 prop 2
##    0.5    0.8
```

P 值 9.315×10^{-09} 结论是拒绝原假设，并且接受备择假设，即 $P_A < P_B$，在原假设成立的情况下，样本落入拒绝域的概率很小，小于 0.05，即在一次实验中，样本不可能落入拒绝域，应当接受原假设，因为将备择假设设为

备择假设 $P_A > P_B$

```
prop.test(x, alternative = "greater")
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  x
## X-squared = 31.632, df = 1, p-value = 1
## alternative hypothesis: greater
## 95 percent confidence interval:
##  -0.3762478  1.0000000
## sample estimates:
## prop 1 prop 2
##    0.5    0.8
```

P 值为 1 不能拒绝原假设，在原假设成立的情况下，样本落入拒绝域的概率是 1

备择假设和原假设在这里是对立的关系

页面 A 观测到的点击率为 50% 页面 B 观测到的点击率为 80%，设置检验问题

$$H_0 : P_A = P_B \quad vs. \quad H_1 : P_A \leq P_B$$

页面点击率 A 等于 B，则备择假设页面点击率 A 不大于 B

默认启用 Yates' 连续性校正 (continuity correction, 简称 CC)

23.19.3 比例齐性检验

原假设四个组里面病人中吸烟的比例是相同的，备择假设是四个组里面至少有一个组的吸烟比例是不同的

```
## Data from Fleiss (1981), p. 139.
## H0: The null hypothesis is that the four populations from which
##     the patients were drawn have the same true proportion of smokers.
## A:  The alternative is that this proportion is different in at
##     least one of the populations.

smokers <- c(83, 90, 129, 70)
```

© 基础统计

```
patients <- c(86, 93, 136, 82)
prop.test(smokers, patients)
```

```
##
##  4-sample test for equality of proportions without continuity
##  correction
##
## data:  smokers out of patients
## X-squared = 12.6, df = 3, p-value = 0.005585
## alternative hypothesis: two.sided
## sample estimates:
##    prop 1    prop 2    prop 3    prop 4
## 0.9651163 0.9677419 0.9485294 0.8536585
```

Wilson 检验统计量 [Wilson, 1927] 考虑单样本比例 p 的区间估计问题，

Probable Inference (Usual): 可能的推断，或然推断，概率推断

在某个总体中抽取 n 个样本，观测到某个比率/频率 p_0，相应的标准差 $\sigma_0 = (p_0 q_0/n)^{1/2}$，常见的概率推断表述是说：比率 p 的真值落在区间 $[p_0 - \lambda\sigma_0, p_0 + \lambda\sigma_0]$ 外的概率小于等于 P_λ，并且随着 λ 增大，P_λ 减小。

如果使用 Tchebysheff 切比雪夫准则，我们知道 P_λ 本身小于 $1/\lambda^2$，但是如果使用概率表 P_λ 是概率密度曲线与坐标 $\pm\lambda\sigma_0$ 之外的部分围成的面积。尽管切比雪夫准则在估计 P_λ 的时候过于保守，但是概率表给出了一个本质的估计。

严格来说，上面给出的概率推断的表述是简略的。真实概率 p 落在指定范围之外的机会要么是 0 要么是 1，就是说 p 要么在那个范围要么不在那个范围。观测的比率 p_0 有更大或更小的机会落在真实比率 p 的某个区间。观测者运气不好，观测到一个相对罕见的事件发生了，基于已有的推断理论，他会获得一个相当宽的标记。

Probable Inference (Improved):

一个更好的方式来阐述推理过程：

有某个比率 p 它的标准差是 $(pq/n)^{1/2} = \sigma$，一个观测糟糕如 p_0 发生的可能性，即 p_0 落在区间 $[p - \lambda\sigma, p + \lambda\sigma]$ 是小于等于 P_λ。

这个表述强调了特殊观测相对于一般典型情况更容易犯的错误。

两样本比例 p 的检验问题。

思路需要推导，考虑如下检验问题

$$H_0 : P_A \geq P_B \quad vs. \quad H_1 : P_A < P_B$$

比例检验，未知 p 的情况下，且样本量有限，是 t 分布多种二项检验的办法 [Newcombe, 1998]

© 黄湘云

提示

切比雪夫不等式 Chebyshev, 1821-1894
设随机变量 X 的数学期望和方差都存在，则对任意常数 $\epsilon > 0$，有

$$P(|X - EX| \geq \epsilon) \leq \frac{Var(X)}{\epsilon^2} \tag{23.3}$$

$$P(|X - EX| \leq \epsilon) \geq 1 - \frac{Var(X)}{\epsilon^2} \tag{23.4}$$

23.20 比例趋势检验 prop.trend.test

Performs χ^2 test for trend in proportions, i.e., a test asymptotically optimal for local alternatives where the log odds vary in proportion with score. By default, score is chosen as the group numbers.

```
usage(prop.trend.test)
```

```
prop.trend.test(x, n, score = seq_along(x))
```

23.21 Quade 检验 quade.test

Quade Test

Performs a Quade test with unreplicated blocked data.

```
usage(quade.test)
quade.test(y, ...)
usage("quade.test.default")
## Default S3 method:
quade.test(y, groups, blocks, ...)
usage("quade.test.formula")
## S3 method for class 'formula'
quade.test(formula, data, subset, na.action, ...)
```

23.22 正态性检验 shapiro.test

> Usually (but not always) doing tests of normality reflect a lack of understanding of the power of rank tests, and an assumption of high power for the tests (qq plots don't always help with that because of their subjectivity). When possible it's good to choose a robust method. Also, doing pre-testing for normality can affect the type I error of the overall analysis.
>
> — Frank Harrell [3]

检验：拒绝原假设和接受原假设的风险，数据本身和理论的正态分布的距离，抛开 P 值

[3] https://stat.ethz.ch/pipermail/r-help/2005-April/070508.html

Shapiro 和 Wilk's 提出的 W 检验

Performs the Shapiro-Wilk test of normality.

```
usage(shapiro.test)
```

```
shapiro.test(x)
```

23.23 正态性检验 Epps-Pully 检验

> The issue really comes down to the fact that the questions: "exactly normal?", and "normal enough?" are 2 very different questions (with the difference becoming greater with increased sample size) and while the first is the easier to answer, the second is generally the more useful one.
>
> — Greg Snow [4]

EP 检验对多种备择假设有较高的效率，利用样本的特征函数和正态分布的特征函数的差的模的平方产生的一个加权积分得到 EP 检验统计量 [Epps and Pulley, 1983]

提示

样本量 $n \geq 200$ EP 检验统计量 T_{EP} 非常接近 $n = \infty$ 时 T_{EP} 的分位数。

设 $x_1, \ldots, x_n$ 是来自正态总体 $N(\mu, \sigma^2)$ 的样本，EP 检验统计量定义为

$$T_{EP} = 1 + \frac{n}{\sqrt{3}} + \frac{2}{n}\sum_{i=2}^{n}\sum_{j=1}^{i-1}\exp\left\{-\frac{(x_j - x_i)^2}{2s_\star^2}\right\} - \sqrt{2}\sum_{i=1}^{n}\exp\left\{-\frac{(x_i - \bar{x})^2}{4s_\star^2}\right\}$$

其中 $\bar{x}, s_\star^2$ 就是样本均值和（除以 n 的）样本方差

提示

几个正态性检验的功效比较 https://arxiv.org/ftp/arxiv/papers/1605/1605.06293.pdf 和 PoweR 包 [Lafaye de Micheaux and Tran, 2016]

23.24 学生 t 检验 `t.test`

t 分布的推导、t 分布的形式两样本的均值检验到 Behrens-Fisher 问题到大规模推荐系统中的 A/B 检验

23.24.1 正态总体两样本的均值之差的检验

常见检验问题

[4] https://stat.ethz.ch/pipermail/r-help/2009-May/390164.html

© 王湘君

$$\text{I} \quad H_0: \mu_1 - \mu_2 \le 0 \quad vs. \quad H_1: \mu_1 - \mu_2 > 0 \tag{23.5}$$

$$\text{II} \quad H_0: \mu_1 - \mu_2 \ge 0 \quad vs. \quad H_1: \mu_1 - \mu_2 < 0 \tag{23.6}$$

$$\text{III} \quad H_0: \mu_1 - \mu_2 = 0 \quad vs. \quad H_1: \mu_1 - \mu_2 \ne 0 \tag{23.7}$$

23.24.1.1 方差 σ_1^2, σ_2^2 已知

检验统计量服从标准正态分布

```
set.seed(2019)
x1 <- rnorm(100, mean = 10, sd = 2.5)
y1 <- rnorm(80, mean = 6, sd = 4.5)
u0 <- (mean(x1) - mean(y1)) / sqrt(2.5^2 / 100 + 4.5^2 / 80)
```

$$u = \frac{(\bar{X} - \bar{Y}) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{m} + \frac{\sigma_2^2}{n}}}$$

$u \sim N(0,1)$，检验统计量 u 对应的样本值 u_0，检验的拒绝域和 P 值如下

$$W_1 = \{u \ge u_{1-\alpha}\}, \quad p_1 = 1 - \Phi(u_0)$$

对检验问题 I，给定显著性水平 $\alpha = 0.05$，得出拒绝域 $\{u \ge 1.645\}$，计算样本观察值得到的检验统计量的值 $u_0 = 7.946$，而该值落在拒绝域，所以拒绝原假设，即拒绝 $\mu_1 - \mu_2 \le 0$，则接受 $\mu_1 - \mu_2 > 0$。

```
# 计算拒绝域
qnorm(1 - 0.05)
```

```
## [1] 1.644854
```

```
# 计算 P 值
1 - pnorm(u0)
```

```
## [1] 9.992007e-16
```

23.24.1.2 方差 $\sigma_1^2 = \sigma_2^2 = \sigma^2$ 未知

检验统计量服从自由度为 $m + n - 2$ 的 t 分布

```
set.seed(2019)
x1 <- rnorm(100, mean = 10, sd = 4.5)
y1 <- rnorm(80, mean = 6, sd = 4.5)
s_w <- sqrt(1 / (100 + 80 - 2) * ((100 - 1) * var(x1) + (80 - 1) * var(y1)))
t0 <- (mean(x1) - mean(y1)) / (s_w * sqrt(1 / 100 + 1 / 80))
```

样本观察值 $t_0 = 6.6816 > t_{0.95}(100 + 80 - 2) = 1.653$ 落在拒绝域内，对于检验问题 I 我们要拒绝原假设

内部资料©

```r
# 临界值: 0.95 分位点对应的分位数
qt(1 - 0.05, df = 100 + 80 - 2)
```

```
## [1] 1.653459
```

```r
# p 值
1 - pt(t0, df = 100 + 80 - 2, lower.tail = TRUE)
```

```
## [1] 1.461666e-10
```

利用 R 内置的 `t.test()` 函数计算

```r
t.test(x = x1, y = y1, alternative = "greater", var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  x1 and y1
## t = 6.6816, df = 178, p-value = 1.462e-10
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  3.249227      Inf
## sample estimates:
## mean of x mean of y
##  9.669997  5.352296
```

与线性回归比较

```r
dat <- data.frame(
  value = c(x1, y1),
  group = c(rep("x1", length(x1)), rep("y1", length(y1)))
)
fit <- lm(value ~ 1 + I(group == "y1"), data = dat)
# fit <- lm(value ~ 0 + I(group == "y1"), data = dat) # 无截距项
summary(fit)
```

```
##
## Call:
## lm(formula = value ~ 1 + I(group == "y1"), data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2282  -3.0198  -0.2959   3.0161  12.1921
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)            9.6700     0.4308  22.446  < 2e-16 ***
## I(group == "y1")TRUE  -4.3177     0.6462  -6.682 2.92e-10 ***
## ---
```

© 王湘君


```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.308 on 178 degrees of freedom
## Multiple R-squared:  0.2005, Adjusted R-squared:  0.196
## F-statistic: 44.64 on 1 and 178 DF,  p-value: 2.923e-10
```

注意

lm 回归和 t 检验的差别，回归系数第二行，t 统计量为 -6.682，P 值为 2.92e-10，前者是因为截距项，后者是因为双边检验（模型系数显著性检验是和 0 比较），所以有 2 倍的关系。直观解释详见 翻译：常见统计检验的本质都是线性模型（或：如何教统计学）

两样本方差不齐、样本量严重不等，在大样本和小样本情况下的比较，t 检验方差不齐有多重要

23.24.1.3 方差 σ_1^2/σ_2^2 已知

方差比 $c = \sigma_1^2/\sigma_2^2$ 已知

23.24.1.4 方差 σ_1^2/σ_2^2 未知

英国统计学家 William Sealy Gosset (1876-1937) 于 1908 年在杂志《Biometrics》上以笔名 Student 发表论文《The probable error of a mean》["Student", 1908]，论文中展示了独立同正态分布的样本 $x_1, \ldots, x_n \overset{i.i.d}{\sim} \mathcal{N}(\mu, \sigma^2)$ 的样本方差 s^2 和样本标准差 s 的抽样分布，根据均值和标准差不相关的性质导出 t 分布，宣告 t 分布的诞生，因其在小样本领域的突出贡献，W. S. Gosset 进入世纪名人录 [Heyde et al., 2001]

$$\frac{(n-1)s^2}{\sigma^2} \sim \chi^2(n-1)$$

$$\frac{\bar{x}-\mu}{s/\sqrt{n}} \sim t(n-1)$$

$$E(s^2) = \sigma^2, \quad Var(s^2) = \frac{2\sigma^4}{n-1}$$

1. 两样本的样本量很大，总体方差未知，检验两样本均值的显著性检验，极限分布是正态，u 检验
2. 两个样本的样本量不是很大，总体方差也未知，检验两样本均值的显著性检验，即著名的 Behrens-Fisher 问题，Welsh 在 1938 年提出近似服从自由度为 ℓ 的 t 分布。

Egon Pearson 接过他父亲 Karl Pearson 的职位，担任伦敦大学学院的高尔顿统计教授

许宝$\boxed{\text{F}}$在 Jerzy Neyman 和 Egon Pearson 主编的杂志《Statistical Research Memoirs》发表第一篇关于 Behrens-Fisher 问题的论文

这里提及许宝$\boxed{\text{F}}$（Pao-Lu Hsu）的贡献 [HSU, 1938]，

陈家鼎和郑忠国一起整理了许宝$\boxed{\text{F}}$的生平事迹和学术成就，见《许宝$\boxed{\text{F}}$先生的生平和学术成就》。

1998 年关于 Behrens-Fisher 问题的综述 [Kim and Cohen, 1998]

钟开莱（Kai-Lai Chung）将许宝$\boxed{\text{F}}$的论文集整理出版 [HSU, 1983]

`t.test()` 提供单样本和两样本的检验

© 基础统计

```
usage(t.test)
## S3 method for class 'test'
t(x, ...)
usage("t.test.default")
## Default S3 method:
t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"), mu = 0,
    paired = FALSE, var.equal = FALSE, conf.level = 0.95, ...)
usage("t.test.formula")
## S3 method for class 'formula'
t.test(formula, data, subset, na.action, ...)
```

学生睡眠数据 sleep 见图 23.2

图 23.2: 学生睡眠数据 sleep

两个样本的 Welch's t 检验，总体方差未知，样本量也不大，两样本均值差的显著性检验

© 王建国


```r
## 等价于 with(sleep, t.test(extra[group == 1], extra[group == 2]))
t.test(extra ~ group, data = sleep)
```

```
##
##  Welch Two Sample t-test
##
## data:  extra by group
## t = -1.8608, df = 17.776, p-value = 0.07939
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  -3.3654832  0.2054832
## sample estimates:
## mean in group 1 mean in group 2
##            0.75            2.33
```

实际上睡眠数据是配对的，我们可以做配对数据的检验

```r
## 数据变形操作，长格式变为宽格式
sleep2 <- reshape(sleep,
  direction = "wide",
  idvar = "ID", timevar = "group"
)
# R 4.0.0
t.test(Pair(extra.1, extra.2) ~ 1, data = sleep2)
```

```
##
##  Paired t-test
##
## data:  Pair(extra.1, extra.2)
## t = -4.0621, df = 9, p-value = 0.002833
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.4598858 -0.7001142
## sample estimates:
## mean of the differences
##                   -1.58
```

注意

函数 `t.test()` 和 `wilcox.test()` 的公式接口要求 R 版本在 4.0.0 及以上

23.24.2 办公软件里的 T 检验

以 MacOS 上的 Numbers 表格软件为例，如图23.3所示，首先打开 Numbers 软件，新建工作表，输入两组数值，然后点击空白处，再从顶部导航栏找到「插入」菜单，「公式」选项，点击扩展选项「新建公式」，在弹出的会话条里输入 TTEST，依次选择第一组，第二组值，检验类型和样本类型，最后点击确认，即可得到两样本 T 检验的 P 值结果。

图 23.3: MacOS 的办公软件 Numbers 做两样本 T 检验

微软 Excel 办公软件也提供 T 检验计算器，和 MacOS 系统上的 Numbers 办公软件类似，它提供 `T.TEST` 函数，计算结果也一样，此处从略。R 软件自带 `t.test()` 函数，也是用于做 T 检验，如下：

```
t.test(x = c(3, 4, 5, 8, 9, 1, 2, 4, 5), y = c(6, 19, 3, 2, 14, 4, 5, 17, 1))
```

```
##
##  Welch Two Sample t-test
##
## data:  c(3, 4, 5, 8, 9, 1, 2, 4, 5) and c(6, 19, 3, 2, 14, 4, 5, 17, 1)
## t = -1.3622, df = 10.255, p-value = 0.2023
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.767183  2.100516
## sample estimates:
## mean of x mean of y
##  4.555556  7.888889
```

23.25 方差比检验 `var.test`

TeachingDemos 的 `sigma.test()` 方差检验，适用于正态总体，它对非正态性很敏感。

F 检验：来自正态总体的两个样本的方差比较

```
usage(var.test)
var.test(x, ...)
usage("var.test.default")
## Default S3 method:
var.test(x, y, ratio = 1, alternative = c("two.sided", "less", "greater"),
    conf.level = 0.95, ...)
usage("var.test.formula")
## S3 method for class 'formula'
var.test(formula, data, subset, na.action, ...)
```

© 草稿

23.26 Wilcoxon 秩和检验 wilcox.test

单样本 Wilcoxon 秩和检验，两样本 Wilcoxon 符号秩检验，也叫 Mann-Whitney 检验

Wilcoxon Rank Sum and Signed Rank Tests

Performs one- and two-sample Wilcoxon tests on vectors of data; the latter is also known as 'Mann-Whitney' test.

```
usage(wilcox.test)
wilcox.test(x, ...)
usage("wilcox.test.default")
## Default S3 method:
wilcox.test(x, y = NULL, alternative = c("two.sided", "less", "greater"),
    mu = 0, paired = FALSE, exact = NULL, correct = TRUE, conf.int = FALSE,
    conf.level = 0.95, tol.root = 1e-04, digits.rank = Inf, ...)
usage("wilcox.test.formula")
## S3 method for class 'formula'
wilcox.test(formula, data, subset, na.action, ...)
```

- `coin::wilcox_test` for exact, asymptotic and Monte Carlo conditional p-values, including in the presence of ties.

coin 包 [Hothorn et al., 2008] 提供大量基于秩的检验

23.26.1 ROC 曲线和 wilcox.test 检验的关系

https://github.com/xrobin/pROC/wiki/FAQ---Frequently-asked-questions#can-i-test-if-a-single-roc-curve-is-significantly-different-from-05

ROC 曲线越往左上角拱越好，AUC 是 ROC 曲线下的面积，所以 AUC 指标越接近 1 越好。

对每个标签的预测概率指定服从均匀分布，相当于随机猜测，所以最后 ROC 会接近对角线，而且样本量越大越接近，AUC 会越来越接近 0.5

> 再往深一点就是研究一下 R 内置的排序算法，因为计算 AUC 最核心的步骤是排序。`order` 函数默认的排序方法是 `auto` 即当数据量较小的时候，自动选择 radix 排序，当数据量比较大的时候，自动选择 shell 排序[5]

```
# 模拟一些数据
set.seed(2019) # 设置随机数种子
N <- 10^5 # 样本量
sim_dat <- cbind.data.frame(
  pred = runif(N),
  label = rbinom(N, size = 1, prob = 0.95)
)

# 计算 auc 的函数
```

[5] radix 排序翻译过来叫桶排序或基数排序，详细描述见 `?sort`

内部© 草稿

```r
# dat is a data.frame as input return AUC value
comp_auc <- function(dat, show_roc = TRUE) {
  # order label by predicted probability
  dat <- dat[order(dat$pred, dat$label, decreasing = TRUE), ]

  # total samples
  n_total <- length(dat$label)

  # number of positive label 1
  n_pos <- sum(dat$label)

  # number of negative label 0
  n_neg <- n_total - n_pos

  # calculate TPR and FPR
  tpr <- cumsum(dat$label) / n_pos
  fpr <- (1:n_total - cumsum(dat$label)) / n_neg

  # calculate auc
  auc <- 0
  for (i in 1:(n_total - 1)) {
    auc <- auc + (fpr[i + 1] - fpr[i]) * tpr[i]
  }
  # show ROC curve or not?
  if (show_roc) {
    plot(fpr, tpr, type = "l")
  }
  auc
}

comp_auc(dat = sim_dat, show_roc = FALSE)
```

```
## [1] 0.5015558
```

模拟一个逻辑回归模型测试自编 AUC 计算程序和 R 包 pROC 计算结果

```r
set.seed(2018)
N <- 10^4 # 样本量
x <- rnorm(N)
beta_0 <- 0.5
beta_1 <- 0.3
eta <- beta_0 + beta_1 * x
# 模拟数据集
dat <- data.frame(x = x, y = rbinom(N, 1, prob = exp(eta) / (1 + exp(eta))))

# 数据集分隔
```

© 基础统计


```
is_train <- sample(1:nrow(dat), N * 0.7)
train <- dat[is_train, ]
test <- dat[-is_train, ]
# 模型拟合
fit <- glm(y ~ x, data = train, family = binomial(link = "logit"))
# 预测
y_pred <- predict(fit, newdata = test, type = "response")

dat2 <- data.frame(pred = y_pred, label = test$y)
# 计算 auc
comp_auc(dat = dat2, show_roc = FALSE)
```

```
## [1] 0.5850287
```

对比 R 包 pROC 的计算结果是一致的

```
pROC::auc(test$y, y_pred)
```

计算一下运行时间

```
# 100 万样本
system.time(comp_auc(dat = dat2, show_roc = FALSE))
```

```
##    user  system elapsed
##   0.003   0.000   0.003
```

更多关于 auc 计算的讨论见统计之都论坛帖 https://d.cosx.org/d/419436，我感觉这个问题最后会归结到排序问题。

```
# https://stat.ethz.ch/pipermail/r-help/2005-April/069217.html
trap.rule <- function(x, y) sum(diff(x) * (y[-1] + y[-length(y)])) / 2
```

23.27 3 + 1 统计检验

Wald 检验，似然比检验/ Wilks 检验，得分检验/Rao 检验，梯度检验

> Unfortunately, this is one of those situations where as far as I can tell all of the real statisticians are out there playing with large data sets where the small-sample corrections are not so important and leaving the rest of us to figure it out for ourselves ...
>
> — Ben Bolker [6]

[6] https://stat.ethz.ch/pipermail/r-sig-mixed-models/2011q4/017392.html

© 王湘黎

表 23.2: 伯克利大学各个院系的录取人数

Admit	Gender	DeptA	DeptB	DeptC	DeptD	DeptE	DeptF
Admitted	Male	512	353	120	138	53	22
Rejected	Male	313	207	205	279	138	351
Admitted	Female	89	17	202	131	94	24
Rejected	Female	19	8	391	244	299	317

23.28 经典案例

23.28.1 1973 年加州大学伯克利分校的学生招生

录取人数按院系和性别分类统计，研究目标是各个院系在录取学生的时候是否有性别歧视？统计数据见表 23.2

```
as.data.frame(UCBAdmissions) %>%
  reshape(.,
    v.names = "Freq", idvar = c("Admit", "Gender"),
    timevar = "Dept", direction = "wide", sep = ""
  ) %>%
  knitr::kable(.,
    caption = "伯克利大学各个院系的录取人数",
    row.names = FALSE, col.names = gsub("(Freq)", "Dept", names(.)),
    align = "c"
  )
```

```
# plot(UCBAdmissions, col = "lightblue", border = "white")
library(ggmosaic)
ggplot(data = as.data.frame(UCBAdmissions)) +
  geom_mosaic(aes(weight = Freq, x = product(Gender, Admit), fill = Dept)) +
  coord_flip() +
  theme_minimal() +
  labs(x = "Admit", y = "Gender")
```

```
## Warning: `unite_()` was deprecated in tidyr 1.2.0.
## Please use `unite()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

23.28.2 1976~1977 年美国佛罗里达州的凶杀案件中被告肤色和死刑判决的关系

被告	被害人	判死	不判死
白人	白人	19	132
	黑人	0	9
黑人	白人	11	32

©黄湘云

被告	被害人	判死	不判死
	黑人	6	97

23.28.3 统计专业学生的头发和眼睛的颜色

HairEyeColor 是一个 table 类型的数据对象，和数组的关系 array

```
class(HairEyeColor)
```

```
## [1] "table"
```

```
str(HairEyeColor)
```

```
##  'table' num [1:4, 1:4, 1:2] 32 53 10 3 11 50 10 30 10 25 ...
##  - attr(*, "dimnames")=List of 3
##   ..$ Hair: chr [1:4] "Black" "Brown" "Red" "Blond"
##   ..$ Eye : chr [1:4] "Brown" "Blue" "Hazel" "Green"
##   ..$ Sex : chr [1:2] "Male" "Female"
```

```
apply(HairEyeColor, c(1, 2), sum)
```

```
##        Eye
## Hair    Brown Blue Hazel Green
##   Black    68   20    15     5
##   Brown   119   84    54    29
##   Red      26   17    14    14
##   Blond     7   94    10    16
```

```
# plot(HairEyeColor, col = "lightblue", border = "white")
library(ggmosaic)
ggplot(data = as.data.frame(HairEyeColor)) +
  geom_mosaic(aes(weight = Freq, x = product(Hair, Eye), fill = Sex)) +
  theme_minimal() +
  labs(x = "Hair", y = "Eye")
```

23.29 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
```

©黄湘云

图 23.4: UCBAdmissions 马赛克图

图 23.5: 头发、眼睛颜色和性别的比例

© 王沛勇


```
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ggmosaic_0.3.3  patchwork_1.1.1 nomnoml_0.2.5   ggplot2_3.3.5
## [5] magrittr_2.0.3  formatR_1.11
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.2   xfun_0.30          purrr_0.3.4        colorspace_2.0-3
##  [5] vctrs_0.4.0        generics_0.1.2     viridisLite_0.4.0  htmltools_0.5.2
##  [9] yaml_2.3.5         plotly_4.10.0      utf8_1.2.2         rlang_1.0.2
## [13] pillar_1.7.0       glue_1.6.2         withr_2.5.0        DBI_1.1.2
## [17] plyr_1.8.7         lifecycle_1.0.1    stringr_1.4.0      munsell_0.5.0
## [21] gtable_0.3.0       htmlwidgets_1.5.4  evaluate_0.15      labeling_0.4.2
## [25] knitr_1.38         callr_3.7.0        fastmap_1.1.0      ps_1.6.0
## [29] curl_4.3.2         fansi_1.0.3        Rcpp_1.0.8.3       scales_1.1.1
## [33] webshot_0.5.2      jsonlite_1.8.0     sysfonts_0.8.8     farver_2.1.0
## [37] digest_0.6.29      stringi_1.7.6      bookdown_0.25      processx_3.5.3
## [41] dplyr_1.0.8        ggrepel_0.9.1      grid_4.1.3         cli_3.2.0
## [45] tools_4.1.3        productplots_0.1.1 lazyeval_0.2.2     tibble_3.1.6
## [49] tidyr_1.2.0        crayon_1.5.1       pkgconfig_2.0.3    ellipsis_0.3.2
## [53] data.table_1.14.2  httr_1.4.2         assertthat_0.2.1   rmarkdown_2.13
## [57] R6_2.5.1           compiler_4.1.3
```

© 基础统计

第二十四章　功效分析

CRAN 上有很多功效计算和分析的 R 包，我们针对不同的混合效应模型和统计检验，提供对应的 R 实现。

MKpower 包提供 Welch 和 Hsu（许宝🅕）t 检验、Wilcoxon 秩和检验、符号秩检验的功效分析和样本量计算，经验功效和第一类错误的计算方法是蒙特卡罗模拟。Superpower 基于模拟的方法分析三因素方差分析实验设计的功效，开发者写了本书介绍，详见 https://aaroncaldwell.us/SuperpowerBook/，也开发了两个 Shiny 应用。powerlmm 可用于计算两、三个水平的纵向多水平/线性混合效应模型的功效。pwrAB Welch 两样本 t 检验的功效分析，常用于 A/B 测试。Metin Bulus 开发 PowerUpR 计算响应变量是连续型的多水平随机对照实验统计功效，最小可检测的效应大小，最小样本量要求。simr 通过模拟方法分析广义线性混合效应模型的功效。WebPower 提供相关性、比例、t 检验、单因素方差分析、两因素方差分析、线性回归、逻辑回归、泊松回归、纵向数据分析、结构方程模型和多水平模型等的功效分析，详见网站 https://webpower.psychstat.org/，包含书籍和功效分析的工具。PowerAnalysisIL 功效分析的 shiny 应用 http://daniellakens.blogspot.com/2015/01/always-use-welchs-t-test-instead-of.html。

此外，还有 lmerTest [Kuznetsova et al., 2017] 和 lmtest [Zeileis and Hothorn, 2002]。试验设计 [茆诗松 et al., 2004] 可以视为一种组织形式，包括各类检验，R 语言实战 [Kabacoff, 2015] 作者 Robert I. Kabacoff 创建了网站 Quick-R，实战这本书第 10 章功效分析主要基于 pwr 包来介绍，Jacob Cohen 的著作《Statistical Power Analysis for the Behavioral Sciences》第二版 [Cohen, 1988]

https://powerandsamplesize.com/ 功效和样本量计算器

```
library(pwr)
library(Matrix)
library(lme4)
```

pbkrtest 提供 parametric bootstrap test、Kenward-Roger-type F-test、Satterthwaite-type F-test 用于线性混合效应模型，parametric bootstrap test 用于广义线性混合效应模型

24.1　方差分析检验的功效

`power.anova.test()` 计算平衡的单因素方差分析检验的功效

```
usage(power.anova.test)
```

```
power.anova.test(groups = NULL, n = NULL, between.var = NULL, within.var = NULL,
    sig.level = 0.05, power = NULL)
```

```
power.anova.test(
  groups = 4,        #  4 个组
  between.var = 1,   # 组间方差为 1
```

© 基础统计


```
  within.var = 3,   # 组内方差为 3
  power = 0.95      # 1 - 犯第二类错误的概率
)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##          groups = 4
##               n = 18.18245
##     between.var = 1
##      within.var = 3
##       sig.level = 0.05
##           power = 0.95
##
## NOTE: n is number in each group
```

24.2 比例检验的功效

power.prop.test() 计算两样本比例检验的功效

```
usage(power.prop.test)
```

```
power.prop.test(n = NULL, p1 = NULL, p2 = NULL, sig.level = 0.05, power = NULL,
    alternative = c("two.sided", "one.sided"), strict = FALSE,
    tol = .Machine$double.eps^0.25)
```

功效可以用来计算实验所需要的样本量，检验统计量的功效越大/高，检验方法越好，实验所需要的样本量越少

```
# p1 >= p2 的检验 单边和双边检验
power.prop.test(
  p1 = .65, p2 = 0.6, sig.level = .05,
  power = 0.90, alternative = "one.sided"
)
```

```
##
##      Two-sample comparison of proportions power calculation
##
##               n = 1603.846
##              p1 = 0.65
##              p2 = 0.6
##       sig.level = 0.05
##           power = 0.9
##     alternative = one.sided
##
## NOTE: n is number in *each* group
```

©黄海洋

```r
power.prop.test(
  p1 = .65, p2 = 0.6, sig.level = .05,
  power = 0.90, alternative = "two.sided"
)
```

```
##
##      Two-sample comparison of proportions power calculation
##
##               n = 1968.064
##              p1 = 0.65
##              p2 = 0.6
##       sig.level = 0.05
##           power = 0.9
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

pwr 包 `pwr.2p.test()` 函数提供了类似 `power.prop.test()` 函数的功能

```r
library(pwr)
# 明确 p1 > p2 的检验
# 单边检验拆分更加明细，分为大于和小于
pwr.2p.test(
  h = ES.h(p1 = 0.65, p2 = 0.6),
  sig.level = 0.05, power = 0.9, alternative = "greater"
)
```

```
##
##    Difference of proportion power calculation for binomial distribution (arcsine transformation)
##
##               h = 0.1033347
##               n = 1604.007
##       sig.level = 0.05
##           power = 0.9
##     alternative = greater
##
## NOTE: same sample sizes
```

已知两样本的样本量不等，检验 H0: $p_1 = p_2$ H1: $p_1 \neq p_2$ 的功效

```r
library(pwr)
pwr.2p2n.test(
  h = 0.30, n1 = 80, n2 = 245,
  sig.level = 0.05, alternative = "greater"
)
```

```
##
##    difference of proportion power calculation for binomial distribution (arcsine transformation)
##
```

©黄文海


```
##                h = 0.3
##               n1 = 80
##               n2 = 245
##        sig.level = 0.05
##            power = 0.7532924
##      alternative = greater
##
## NOTE: different sample sizes
```

h 表示两个样本的差异，计算得到的功效是 0.75

24.3 t 检验的功效

power.t.test() 计算单样本或两样本的 t 检验的功效，或者根据功效计算参数，如样本量

Cohen's d 单样本/配对 t 检验的功效分析

```r
n = 30  # 样本量（只是一个例子）
x = seq(0, 12, 0.01)
library(ggplot2)
dat <- data.frame(xx = x/sqrt(n), yy = 2 * (1 - pt(x, n - 1)))
ggplot(data = dat, aes(x = xx, y = yy)) +
  geom_line() +
  geom_vline(xintercept = c(0.01, 0.2, 0.5, 0.8, 1.2, 2), linetype = 2) +
  theme_minimal() +
  labs(x = "d = t / sqrt(n)", y = "2 * (1 - pt(x, n - 1))")
```

```r
usage(power.t.test)
```

```
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05, power = NULL,
    type = c("two.sample", "one.sample", "paired"),
    alternative = c("two.sided", "one.sided"), strict = FALSE,
    tol = .Machine$double.eps^0.25)
```

```r
power.t.test(
  n = 100, delta = 2.2,
  sd = 1, sig.level = 0.05,
  type = "two.sample",
  alternative = "two.sided"
)
```

```
##
##      Two-sample t test power calculation
##
##               n = 100
##           delta = 2.2
##              sd = 1
##       sig.level = 0.05
```

© 黄湘云

图 24.1: t 检验的功效

```
##           power = 1
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

表 24.1: 函数 `power.t.test()` 的参数表

参数	含义
`n`	每个组的样本量
`delta`	两个组的均值之差
`sd`	标准差，默认值 1
`sig.level`	显著性水平，默认是 0.05 （犯第 I 类错误的概率）
`power`	检验的功效（1 - 犯第 II 类错误的概率）
`type`	t 检验的类型 `"two.sample"` 两样本、`"one.sample"` 单样本或 `"paired"` 配对样本
`alternative`	单边或双边检验，取值为 `"two.sided"` 或 `"one.sided"`

参数 `n`，`delta`，`power`，`sd` 和 `sig.level` 必须有一个值为 `NULL`，为 `NULL` 的参数是由其它参数决定的。

Jacob Cohen 提出的 Cohen's d 和 Cohen's f 详见书籍 [Cohen, 1988]，他的代表性文章，地球是圆的 [Cohen, 1994]

© 王建平


```r
# 前面 t 检验和方差分析检验的等价功效计算
library(pwr)
pwr.t.test(
  d = 2.2 / 6.4,
  n = 100,
  sig.level = 0.05,
  type = "two.sample",
  alternative = "two.sided"
)
```

```
##
##      Two-sample t test power calculation
##
##               n = 100
##               d = 0.34375
##       sig.level = 0.05
##           power = 0.6768572
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```r
# f 是如何和上面的组间/组内方差等价指定的
pwr.anova.test(
  k = 4, # 组数
  f = 0.5,
  power = 0.95 # 检验的效
)
```

```
##
##      Balanced one-way analysis of variance power calculation
##
##               k = 4
##               n = 18.18244
##               f = 0.5
##       sig.level = 0.05
##           power = 0.95
##
## NOTE: n is number in each group
```

```r
with(
  aggregate(
    data = PlantGrowth, weight ~ group,
    FUN = function(x) c(dist_mean = mean(x), dist_sd = sd(x))
  ),
  cbind.data.frame(weight, group)
)
```

汪荣贵 ©

注意

R 3.5.0 以后，函数 aggregate 的参数 drop 默认设置为 TRUE 表示扔掉未用来分组的变量，聚合返回的是一个矩阵类型的数据对象。

ggsignif 添加显著性注释

```
library(ggplot2)
library(ggsignif)

ggplot(data = PlantGrowth, aes(x = group, y = weight)) +
  geom_boxplot() +
  geom_signif(comparisons = list(c("ctrl", "trt1"), c("trt1", "trt2")),
              map_signif_level = function(p) sprintf("p = %.2g", p),
              textsize = 6, test = "t.test") +
  theme_minimal()
```

无条件 2×2 列联表

fisher.test https://en.wikipedia.org/wiki/Fisher's_exact_test

Exact https://en.wikipedia.org/wiki/Barnard's_test exact.test power.exact.test

exact2x2

24.4 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
```

© 王湘君


```
## other attached packages:
## [1] ggplot2_3.3.5  lme4_1.1-28    Matrix_1.4-1   pwr_1.3-0      formatR_1.11
## [6] magrittr_2.0.3
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.2 xfun_0.30        purrr_0.3.4      splines_4.1.3
##  [5] lattice_0.20-45  colorspace_2.0-3 vctrs_0.4.0      generics_0.1.2
##  [9] htmltools_0.5.2  yaml_2.3.5       utf8_1.2.2       rlang_1.0.2
## [13] nloptr_2.0.0     pillar_1.7.0     glue_1.6.2       withr_2.5.0
## [17] DBI_1.1.2        lifecycle_1.0.1  stringr_1.4.0    munsell_0.5.0
## [21] gtable_0.3.0     evaluate_0.15    labeling_0.4.2   knitr_1.38
## [25] fastmap_1.1.0    curl_4.3.2       fansi_1.0.3      Rcpp_1.0.8.3
## [29] scales_1.1.1     sysfonts_0.8.8   farver_2.1.0     digest_0.6.29
## [33] stringi_1.7.6    bookdown_0.25    dplyr_1.0.8      grid_4.1.3
## [37] cli_3.2.0        tools_4.1.3      tibble_3.1.6     crayon_1.5.1
## [41] pkgconfig_2.0.3  MASS_7.3-56      ellipsis_0.3.2   assertthat_0.2.1
## [45] minqa_1.2.4      rmarkdown_2.13   R6_2.5.1         boot_1.3-28
## [49] nlme_3.1-157     compiler_4.1.3
```

© 王湘君

第二十五章　试验设计

```
library(magrittr)
library(ggplot2)
```

注意

我想不少人初次见到本章题目首先疑惑的可能是到底是试验还是实验？这里做一下说明，实验的意思是带有验证性的目的，已经有结果了，做实验验证某个规律，常常用在物理、化学的课堂里，学生做实验验证自由落体运动、做实验测量重力加速度等等。试验的意思是人为设定一系列操作步骤去探索未知，不确定结果如何，试一试。

试验设计（Design of Experiment，简称 DOE）是一个应用性很强的学科领域，R. A. Fisher 曾在英国洛桑试验站做实验验证孟德尔的豌豆实验结果。

Vikneswaran 提供了一份书籍 Berger and Maurer [2002] 的补充材料 — An R companion to "Experimental Design"，目前 Paul Berger 的这本书已经迭代到第二版 [Berger et al., 2018]，2015 年 Paul Berger 出版了新书《Improving the User Experience through Practical Data Analytics: Gain Meaningful Insight and Increase Your Bottom Line》[Fritz and Berger, 2015] 颇具应用性，结合产品用户体验来谈试验设计。

Bill Venables 开发的 conf.design 是试验设计领域的核心 R 包，CRAN 官网上试验设计视图 https://cran.r-project.org/view=ExperimentalDesign 可以让我们对试验设计这个领域有一个粗略的了解。

推荐读者使用贴合 R 语言的试验设计入门书《Design and Analysis of Experiments with R》[Lawson, 2014]，作者提供相应的 R 包 daewr 打包了该书的数据和代码。另外，推荐的读物是《Statistics for Experimenters: Design, Innovation, and Discovery》[Box et al., 2005] 和《Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing》[Kohavi et al., 2020]。

另一个和试验设计紧密相关的话题是敏感性分析，推荐 Devin Incerti 的敏感性分析系列博客 https://devinincerti.com/blog.html，R 包 sensitivity 提供 140+ 页的手册，功能非常强，模型的全局敏感性分析，SWATplusR SWAT 分析法和 R 语言结合。

25.1　学生睡眠质量

```
ggplot(data = sleep, aes(x = group, y = extra, color = group)) +
  geom_boxplot() +
  geom_jitter() +
  theme_minimal()
```

© 黄湘云

25.2 驱虫喷雾的效果

InsectSprays 数据集 [Beall, 1942] 来源于农业实验，记录了不同杀虫剂的效果，即杀虫剂过后，单位实验区域内虫子的数量，如图25.1所示，横轴表示杀虫剂种类，纵轴表示虫子数量。

```
ggplot(data = InsectSprays, aes(x = spray, y = count, color = spray)) +
  geom_boxplot() +
  geom_jitter() +
  theme_minimal()
```

先创建一个 aov 对象，把它命名为 mod1，见下方

```
mod1 <- aov(count ~ spray, data = InsectSprays)
```

第一个参数告诉 R count 是响应变量，spray 是协变量，第二个参数告诉 R 去对象 InsectSprays 中寻找这些变量。下面把分析结果以一种漂亮的格式打印出来

```
summary(mod1)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## spray        5   2669   533.8    34.7 <2e-16 ***
## Residuals   66   1015    15.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

© 黄湘云

图 25.1: 不同杀虫剂的效果

表格中的条目是很容易理解的，比如最右边的列表示 P 值。如果我们想做固定显著性水平下的检验，比如 $\alpha = 0.075$ 时的 F 统计量的值，

```
qf(0.075, 5, 66, lower.tail = F)
```

```
## [1] 2.110783
```

上面的命令是说 $F(5, 66)$ 分布的 0.075 分位点，最后一个参数很关键，因为默认情况下 R 计算下分位点，详情见 `?qf`。

方差分析做了三个假设

1. 残差 ϵ_{ij} 是相互独立的随机变量；
2. 残差 ϵ_{ij} 服从正态分布；
3. 残差 ϵ_{ij} 均值为 0，方差是固定的常数。

假设 1 和 3 通过图来检验，假设 2 通过 QQ 图来检验。值得一提的是 mod1 对象除了打印出来，还有很多方法

```
names(mod1)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "contrasts"     "xlevels"       "call"          "terms"
## [13] "model"
```

比如获取残差，考虑到篇幅，这里仅显示前 10 个

© 黄湘云


```
head(mod1$residuals, 10)
```

```
##    1    2    3    4    5    6    7    8    9   10
## -4.5 -7.5  5.5 -0.5 -0.5 -2.5 -4.5  8.5  2.5  5.5
```

```
par(mar = c(4, 4, 2, 2), mfrow = c(2,2))
plot(mod1)
```

```
plot(mod1$fitted.values, mod1$residuals, main = "Residuals vs. Fitted", pch = 20)
abline(h = 0, lty = 2)
```

© 黄湘云

```
plot(mod1$model$spray, mod1$residuals, main = "Residuals vs. Levels" )
```

© 王湘明


```
plot(1:72, mod1$residuals, main = "Residuals vs. Time Order")
abline(h = 0, lty = 2)
```

```
qqnorm(mod1$residuals, pch = 20)
qqline(mod1$residuals)
```

© 黄湘云

如果上面的假设显著失效，我们要采用非参数检验

```
mod2 <- kruskal.test(count ~ spray, data = InsectSprays)
mod2
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  count by spray
## Kruskal-Wallis chi-squared = 54.691, df = 5, p-value = 1.511e-10
```

计算给定水平下的置信区间，构造置信水平为 95% 的区间

$$\bar{X} \pm t_{1-\alpha/2}(s/\sqrt{n})$$

以 A 号杀虫剂为例，

```
xbar = mean(InsectSprays[InsectSprays$spray == "A", "count"])
t_crit <- qt(0.025, mod1$df.residual, lower.tail = F)
s <- sqrt(sum((mod1$residuals)^2) / mod1$df.residual)
n <- sum(InsectSprays$spray == "A")
# 最后置信区间的上下限
c(xbar - t_crit * (s/ sqrt(n)), xbar + t_crit * (s/ sqrt(n)))
```

```
## [1] 12.23958 16.76042
```

比较 A 号和 C 号杀虫剂的效果，计算两个均值差的置信区间

©黄湘云

$$\bar{X}_1 - \bar{X}_2 \pm t_{1-\alpha/2}(s/\sqrt{1/n_1 + 1/n_2})$$

```
n1 <- sum(InsectSprays$spray == "A")
n2 <- sum(InsectSprays$spray == "C")

x1bar = mean(InsectSprays[InsectSprays$spray == "A", "count"])
x2bar = mean(InsectSprays[InsectSprays$spray == "C", "count"])
```

代入公式即可计算得到置信区间

```
(x1bar - x2bar) - t_crit * s * sqrt( 1/ n1 + 1/n2)
```

```
## [1] 9.219948
```

```
(x1bar - x2bar) + t_crit * s * sqrt( 1/ n1 + 1/n2)
```

```
## [1] 15.61339
```

Fisher's 最小显著性检验 (Fisher's Least Significant Difference Test) 即

```
t_crit * s * sqrt( 1/ n1 + 1/n2)
```

```
## [1] 3.196719
```

Tukey's Honestly Significant Difference Test 主要测量成对实验的误差比率，假定每个水平下的实验次数是相等的，只需将上面的 aov 对象传递给函数 `TukeyHSD()`

```
mod3 <- TukeyHSD(mod1, ordered = TRUE)
mod3
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = count ~ spray, data = InsectSprays)
##
## $spray
##           diff       lwr       upr     p adj
## E-C  1.4166667 -3.282742  6.116075 0.9488669
## D-C  2.8333333 -1.866075  7.532742 0.4920707
## A-C 12.4166667  7.717258 17.116075 0.0000000
## B-C 13.2500000  8.550591 17.949409 0.0000000
## F-C 14.5833333  9.883925 19.282742 0.0000000
## D-E  1.4166667 -3.282742  6.116075 0.9488669
## A-E 11.0000000  6.300591 15.699409 0.0000000
## B-E 11.8333333  7.133925 16.532742 0.0000000
## F-E 13.1666667  8.467258 17.866075 0.0000000
## A-D  9.5833333  4.883925 14.282742 0.0000014
## B-D 10.4166667  5.717258 15.116075 0.0000002
## F-D 11.7500000  7.050591 16.449409 0.0000000
## B-A  0.8333333 -3.866075  5.532742 0.9951810
```

```
## F-A  2.1666667 -2.532742  6.866075 0.7542147
## F-B  1.3333333 -3.366075  6.032742 0.9603075
```

其中，diff 表示均值之差，lwr 和 upr 表示置信区间的上下限，p_{adj} 是对应的。检查一下，看看哪些置信区间包含 0，包含 0 的表示不显著，从第三行来看，A 和 C 之间差别显著。之前计算过 A、C 均值，均值之差即

```
(x1bar - x2bar)
```

```
## [1] 12.41667
```

在误差比率 $\alpha = 0.05$ 的情况下，如果你想手动计算 HSD 值

```
q_crit <- qtukey(p = 0.05, nmeans = length(mod1$xlevels[[1]]), df = mod1$df.residual, lower.tail = F)
# mod1$df.residual 是 6
hsd <- q_crit * s / sqrt(6)
hsd
```

```
## [1] 6.645967
```

将模型结果 mod3 用图画出来，见下图

```
plot(mod3)
```

图 25.2: 成对显著性水平

关于多重比较请见 Frank Bretz, Torsten Hothorn, Peter Westfall 的书《Multiple Comparisons Using R》及配套 R 包 multcomp，该 R 包现由 Torsten Hothorn 维护，他还维护了一个由数据集构成的 R 包 TH.data，我们后续章节也会用到。

25.3 重复数不等的多重比较

Tukey 的检验方法要求各个组的重复数相等，而方差分析的重复数不等时，我们需要用如下方法

© 王湘君

$$\frac{(\bar{y}_{i\cdot}-\bar{y}_{j\cdot})-(\mu_i-\mu_j)}{\sqrt{\frac{1}{m_i}+\frac{1}{m_j}}\hat{\sigma}}\sim t(f_e)$$

$$c_{ij}=\sqrt{(r-1)F_{1-\alpha}(r-1,f_e)(\frac{1}{m_i}+\frac{1}{m_j})\hat{\sigma^2}}$$

$\hat{\sigma^2}=S_e/f_e$ 是 σ^2 无偏估计。

$$y_{ij}=\mu+a_i+\epsilon_{ij},\quad i=1,2,\ldots,r,\quad j=1,2,\ldots,m_i.\quad \sum_{i=1}^{r}m_ia_i=0,$$

其中，ϵ_{ij} 相互独立，服从 $\mathcal{N}(0,\sigma^2)$.

$f_e=n-r, S_e=\sum_{i=1}^{r}\sum_{j=1}^{m_i}(y_{ij}-\bar{y}_{i\cdot})^2=S_T-S_A$

25.4 不同地区的草类植物吸收二氧化碳的情况

通过观察不同地区的草类植物吸收二氧化碳的情况，研究植物的耐寒性

```
ggplot(data = CO2, aes(x = conc, y = uptake, color = Type, shape = Treatment)) +
  geom_point() +
  geom_line() +
  facet_wrap(~Plant, ncol = 3) +
  theme_minimal() +
  labs(x = "conc (mL/L)", y = "uptake (umol/m^2 sec)")
```

25.5 果园喷雾剂的效力

评估喷雾杀虫剂的在果园的效果

```
data("OrchardSprays")
```

25.6 验证孟德尔的豌豆实验结果

R. A. Fisher 在农业站做实验验证孟德尔的豌豆实验结果

```
data("npk")
```

豌豆产量和氮 (nitrogen, N) 磷酸盐 (phosphate, P) 钾盐 (potassium, K) 的关系

© 黄湘云

图 25.3: 草类植物吸收二氧化碳的量

© 黄湘云

第六部分

统计模型

© 黄湘云

介绍

统计模型

© 王建国

第二十六章　线性模型

> There's probably some examples, but there are some examples of people using `solve(t(X) %*% W %*% X) %*% W %*% Y` to compute regression coefficients, too.
>
> — Thomas Lumley [1]

26.1　方差分析

> I was profoundly disappointed when I saw that S-PLUS 4.5 now provides "Type III" sums of squares as a routine option for the summary method for aov objects. I note that it is not yet available for multistratum models, although this has all the hallmarks of an oversight (that is, a bug) rather than common sense seeing the light of day. When the decision was being taken of whether to include this feature, "because the FDA requires it" a few of my colleagues and I were consulted and our reply was unhesitatingly a clear and unequivocal "No", but it seems the FDA and SAS speak louder and we were clearly outvoted.
>
> — Bill Venables [2]

方差分析、A/B Test 和多重比较多用于互联网数据 lme 的特例

26.2　单因素方差分析

chickwts 不同的喂食方式对体重的影响

```
boxplot(weight ~ feed, data = chickwts, col = "lightgray",
    varwidth = TRUE, notch = TRUE, main = "chickwt data",
    ylab = "Weight at six weeks (gm)")
```

```
## Warning in (function (z, notch = FALSE, width = NULL, varwidth = FALSE, : some
## notches went outside hinges ('box'): maybe set notch=FALSE
```

[1] https://stat.ethz.ch/pipermail/r-help/2006-March/101596.html

[2] 来源于 Exegeses on Linear Models

© 黄湘云

```
anova(fm1 <- lm(weight ~ feed, data = chickwts))
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df Sum Sq Mean Sq F value    Pr(>F)
## feed       5 231129   46226  15.365 5.936e-10 ***
## Residuals 65 195556    3009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
opar <- par(mfrow = c(2, 2), oma = c(0, 0, 1.1, 0),
            mar = c(4.1, 4.1, 2.1, 1.1))
plot(fm1)
```

© 黄湘云

```
par(opar)
```

sleep

```
## Student's paired t-test 成对样本的 t 检验
with(sleep,
     t.test(extra[group == 1],
            extra[group == 2], paired = TRUE))
```

```
##
##  Paired t-test
##
## data:  extra[group == 1] and extra[group == 2]
## t = -4.0621, df = 9, p-value = 0.002833
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.4598858 -0.7001142
## sample estimates:
## mean of the differences
##                   -1.58
```

```
## The sleep *prolongations*
sleep1 <- with(sleep, extra[group == 2] - extra[group == 1])
summary(sleep1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    1.05    1.30    1.58    1.70    4.60
```

© 统计学基础

```
stripchart(sleep1, method = "stack", xlab = "hours",
           main = "Sleep prolongation (n = 10)")
boxplot(sleep1, horizontal = TRUE, add = TRUE,
        at = .6, pars = list(boxwex = 0.5, staplewex = 0.25))
```

另一个关于测量光速的例子，带分类变量的

```
michelson <- transform(morley,
                       Expt = factor(Expt), Run = factor(Run))
xtabs(~ Expt + Run, data = michelson)  # 5 x 20 balanced (two-way)
```

```
##     Run
## Expt 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
##    1 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
##    2 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
##    3 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
##    4 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
##    5 1 1 1 1 1 1 1 1 1  1  1  1  1  1  1  1  1  1  1  1
```

```
plot(Speed ~ Expt, data = michelson,
     main = "Speed of Light Data", xlab = "Experiment No.")
```

© 黄湘云

```
fm <- aov(Speed ~ Run + Expt, data = michelson)
summary(fm)
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## Run         19 113344    5965   1.105 0.36321
## Expt         4  94514   23629   4.378 0.00307 **
## Residuals   76 410166    5397
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
fm0 <- update(fm, . ~ . - Run)
anova(fm0, fm)
```

```
## Analysis of Variance Table
##
## Model 1: Speed ~ Expt
## Model 2: Speed ~ Run + Expt
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     95 523510
## 2     76 410166 19    113344 1.1053 0.3632
```

ToothGrowth 维生素 C 对牙齿增长的关系

```
coplot(len ~ dose | supp, data = ToothGrowth, panel = panel.smooth,
       xlab = "ToothGrowth data: length vs dose, given type of supplement")
```

© 黄湘云

ToothGrowth data: length vs dose, given type of supplement

26.3 双因素方差分析

`?lm` mlm

26.4 多因素方差分析

MANOVA.RM 和 ffmanova 包处理多因素方差分析

26.5 核学习

基于核的机器学习算法 kernlab

David Meyer 基于 libsvm 开发了 e1071 包，基于核方法实现了非线性回归分类算法

> 线性模型、逻辑回归模型、多项逻辑回归模型、神经网络、朴素贝叶斯、分类回归树等模型和算法借助 Shiny 整合在一起 https://radiant-rstats.github.io/docs/ 和 http://radiant-rstats.github.io/radiant.model/

26.6 通用机器学习

©黄湘云

表 26.1: R 包之间的不一致性，计算预测分类的概率的语法

函数	R 包	代码
lda	MASS	predict(obj)
glm	stats	predict(obj, type = "response")
gbm	gbm	predict(obj, type = "response", n.trees)
mda	mda	predict(obj, type = "posterior")
rpart	rpart	predict(obj, type = "prob")
Weka	RWeka	predict(obj, type = "probability")
logitboost	LogitBoost	predict(obj, type = "raw", nIter)
pamr.train	pamr	pamr.predict(obj, type = "posterior")

26.7 理论基础

$$Y = X\beta + \epsilon \tag{26.1}$$

$$X^\top Y = X^\top X\beta \tag{26.2}$$

$$\hat{\beta} = (X^\top X)^{-1}X^\top Y \tag{26.3}$$

$$\hat{Y} = X(X^\top X)^{-1}X^\top Y \tag{26.4}$$

$$\hat{\sigma^2} = \frac{\|Y - \hat{Y}\|_2}{n - rk(X)} \tag{26.5}$$

$$= \frac{\|(I - X(X^\top X)^{-1}X^\top)Y\|_2}{n - rk(X)} \tag{26.6}$$

$$= \frac{Y^\top(I - X(X^\top X)^{-1}X^\top)Y}{n - rk(X)} \tag{26.7}$$

26.8 多重多元线性回归

参考 John Fox 和 Sanford Weisberg 的著作 [Fox and Weisberg, 2019] 附录[3]

多个响应变量和协变量[4]

多重多元线性回归 multiply linear regression `lm` R 版本 3.6 以上 PR#17407

```
fit_mtcars <- lm(cbind(mpg, qsec) ~ 1, data = mtcars, offset = cbind(wt, wt * 2))
summary(fit_mtcars)
```

```
## Response mpg :
##
## Call:
## lm(formula = mpg ~ 1, data = mtcars, offset = cbind(wt, wt *
##     2))
```

[3] https://socialsciences.mcmaster.ca/jfox/Books/Companion/appendices.html

[4] https://data.library.virginia.edu/getting-started-with-multivariate-multiple-regression/

© 基础统计

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.897  -4.947  -1.316   2.984  15.192
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   16.873      1.219   13.85  8.1e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.893 on 31 degrees of freedom
##
##
## Response qsec :
##
## Call:
## lm(formula = qsec ~ 1, data = mtcars, offset = cbind(wt, wt *
##     2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6842 -2.0793 -0.1693  2.2693  5.1857
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.4142     0.5076   22.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.871 on 31 degrees of freedom
```

26.9 回归诊断

包括线性模型和广义线性模型

Regression Deletion Diagnostics `?influence.measures`

```
library(ggplot2)
library(patchwork)
data("anscombe")

form <- sprintf('y%d ~ x%d', 1:4, 1:4)
fit <- lapply(form, lm, data = anscombe)
```

©黄湘云

```
plot_lm <- function(i) {
  annotate_texts <- c("", "nonlinearity", "outlier", "influential point")
  p <- ggplot(data = anscombe, aes_string(x = paste0("x", i), y = paste0("y", i))) +
    geom_point() +
    geom_abline(intercept = coef(fit[[i]])[1], slope = coef(fit[[i]])[2], color = "red") +
    theme_minimal() +
    labs(
      x = substitute(bold(x[a]), list(a = i)), y = substitute(bold(y[b]), list(b = i)),
      title = bquote(bold(R)^2 == .(round(summary(fit[[i]])$r.squared, 3)))
    )
  p + annotate("text", x = 12, y = 11, label = annotate_texts[i])

}

Reduce("+", lapply(1:4, plot_lm))
```

26.10 1977 年美国人口普查

```
state_data <- data.frame(state.x77, row.names = state.abb)
fit_state <- lm(Life.Exp ~ ., data = state_data)
summary(fit_state)
```

```
##
## Call:
## lm(formula = Life.Exp ~ ., data = state_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48895 -0.51232 -0.02747  0.57002  1.49447
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.094e+01  1.748e+00  40.586  < 2e-16 ***
## Population   5.180e-05  2.919e-05   1.775   0.0832 .
## Income      -2.180e-05  2.444e-04  -0.089   0.9293
## Illiteracy   3.382e-02  3.663e-01   0.092   0.9269
## Murder      -3.011e-01  4.662e-02  -6.459 8.68e-08 ***
## HS.Grad      4.893e-02  2.332e-02   2.098   0.0420 *
## Frost       -5.735e-03  3.143e-03  -1.825   0.0752 .
## Area        -7.383e-08  1.668e-06  -0.044   0.9649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7448 on 42 degrees of freedom
```

©黄湘云

图 26.1: 线性模型可能在欺骗你

©黄湘云

```
## Multiple R-squared:  0.7362, Adjusted R-squared:  0.6922
## F-statistic: 16.74 on 7 and 42 DF,  p-value: 2.534e-10
```

```
# step(fit_state)
```

26.11 石油岩石样品的测量

```
data(rock)
```

多元线性回归

26.12 1888 年瑞士生育率分析

1888 年，瑞士开始进入一个人口转变的阶段，从发展中国家的高出生率开始下滑。数据集 swiss 记录了 1888 年瑞士 47 个说法语的省份的生育率和社会经济指标数据，下面是数据集的部分

```
##              Fertility Agriculture Examination Education Catholic
## Courtelary        80.2        17.0          15        12     9.96
## Delemont          83.1        45.1           6         9    84.84
## Franches-Mnt      92.5        39.7           5         5    93.40
## Moutier           85.8        36.5          12         7    33.77
## Neuveville        76.9        43.5          17        15     5.16
....
```

Fertility（生育率，采用常见的标准生育率统计口径）、Agriculture（男性从事农业生产的比例）、Examination（应征者在军队考试中获得最高等级的比例）、Education（应征者有小学以上教育水平的比例）、Catholic（信仰天主教的比例）、Infant.Mortality（婴儿死亡率，仅考虑出生一年内死亡），各个指标都统一标准化为百分比的形式。其中，Examination 和 Education 是 1887 年、1888 年和 1889 年的平均值。瑞士 182 个地区 1888 年及其它年份的数据可从网站获得。

```
fit_swiss <- lm(Fertility ~ . - 1, data = swiss)
```

```
summary(fit_swiss)
```

```
##
## Call:
## lm(formula = Fertility ~ . - 1, data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.8358  -6.3606  -0.5603   6.0585  23.3203
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## Agriculture       0.11100    0.07424   1.495  0.14233
## Examination       0.44406    0.31435   1.413  0.16514
```

©黄湘云

图 26.2: 1888 年瑞士生育率和社会经济指标的关系

© 王兆军


```
## Education        -0.70674    0.25009  -2.826  0.00719 **
## Catholic          0.11707    0.04860   2.409  0.02046 *
## Infant.Mortality  2.98366    0.31683   9.417 6.53e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.893 on 42 degrees of freedom
## Multiple R-squared:  0.9828, Adjusted R-squared:  0.9807
## F-statistic: 478.8 on 5 and 42 DF,  p-value: < 2.2e-16
```

```r
anova(fit_swiss)
```

```
## Analysis of Variance Table
##
## Response: Fertility
##                  Df Sum Sq Mean Sq   F value    Pr(>F)
## Agriculture       1 204039  204039 2084.6865 < 2.2e-16 ***
## Examination       1  16781   16781  171.4556 < 2.2e-16 ***
## Education         1     24      24    0.2454    0.6229
## Catholic          1   4782    4782   48.8556 1.504e-08 ***
## Infant.Mortality  1   8680    8680   88.6858 6.528e-12 ***
## Residuals        42   4111      98
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Cook 距离 `?plot.lm`

```r
par(mar = c(4, 4, 2, 2))
plot(fit_swiss, which = 4, sub.caption = "")
```

© 黄湘云

```
par(mar = c(4, 4, 2, 2))
plot(fit_swiss, which = 5, sub.caption = "")
```

© 王湘明


```
X <- as.matrix(swiss[, setdiff(names(swiss), "Fertility")])
Y <- as.matrix(swiss[, "Fertility"])
# beta 的估计
(beta_hat <- solve(a = crossprod(X, X), b = crossprod(X, Y)))
```

```
##                        [,1]
## Agriculture       0.1110005
## Examination       0.4440591
## Education        -0.7067362
## Catholic          0.1170662
## Infant.Mortality  2.9836617
```

```
# Y 的预测 MSE 残差平方和
sigma2_hat <- (t(Y) %*% (diag(rep(1, dim(X)[1])) - X %*% solve(crossprod(X)) %*% t(X)) %*% Y)/(dim(X
# RMSE
sqrt(sigma2_hat)
```

```
##          [,1]
## [1,] 9.893187
```

26.13 Intercountry Life-Cycle Savings Data 1960-1970

```
data("LifeCycleSavings")
```

26.14 Longley's Economic Regression Data 1947-1962

```
data("longley")
```

26.15 甲醛的测定

```
ggplot(data = Formaldehyde, aes(x = carb, y = optden)) +
  geom_point() +
  theme_minimal()
```

© 黄湘云

表 26.2: 迈克尔逊光速数据

Expt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	850	740	900	1070	930	850	950	980	980	880	1000	980	930	650	760	810	1000	1000
2	960	940	960	940	880	800	850	880	900	840	830	790	810	880	880	830	800	790
3	880	880	880	860	720	720	620	860	970	950	880	910	850	870	840	840	850	840
4	890	810	810	820	800	770	760	740	750	760	910	920	890	860	880	720	840	850
5	890	840	780	810	760	810	790	810	820	850	870	870	810	740	810	940	950	800

26.16 迈克尔逊光速数据分析

1879 年迈克尔逊光速测量数据，记录了五次实验，每次试验测量 20 次光速，得到表格 26.2

```
reshape(
  data = morley, v.names = "Speed", idvar = "Expt",
  timevar = "Run", direction = "wide", sep = ""
) %>%
  knitr::kable(.,
    caption = "迈克尔逊光速数据",
    row.names = FALSE, col.names = gsub("(Speed)", "", names(.)),
    align = "c"
  )
```

© 黄湘云

数据集 morley 中光速 Speed 已经编码过了，原始观测速度减去了 299000 (km/sec)，为了展示方便

```
ggplot(data = morley, aes(x = Expt, y = Speed, group = Expt)) +
  geom_boxplot() +
  geom_jitter() +
  theme_minimal() +
  labs(x = "Expt", y = "Speed (km/sec)")
```

图 26.3: 1879 年迈克尔逊光速实验数据

26.17 不同喂食方式对小鸡体重的影响 I

```
ggplot(data = chickwts, aes(x = feed, y = weight, color = feed)) +
  geom_boxplot() +
  geom_jitter() +
  theme_minimal()
```

26.18 不同喂食方式对小鸡体重的影响 II

```
ggplot(data = ChickWeight, aes(x = Time, y = weight, group = Chick, color = Diet)) +
  geom_point() +
  geom_line() +
```

© 黄湘云

图 26.4: 不同喂食方式对小鸡的影响

```
facet_wrap(~Diet) +
theme_minimal()
```

© 黄湘云

添加趋势线

```
ggplot(data = ChickWeight,
       aes(x = Time, y = weight, group = Diet, colour = Diet)) +
  facet_wrap(~Diet) +
  geom_jitter() +
  stat_summary(fun = "mean", geom = "line", colour = "black") +
  theme_minimal() +
  labs(
    title = "Chick Weight over Time by Diet",
    x = "Time (days)",
    y = "Weight (grams)"
  )
```

© 黄湘云

26.19 酶的酶联免疫吸附测定

```
ggplot(data = DNase, aes(x= conc,y= density, color = Run)) +
  geom_point() +
  theme_minimal()
```

© 黄湘云

26.20 婴儿的体重随年龄的变化情况

BirthWeight 数据集记录了婴儿的体重随年龄的变化情况，年龄以周为单位计，体重以克为单位计

© 黄湘云

性别和年龄两个变量，分别是离散型的分类变量和连续型的变量

©黄湘云

```r
# 带截距项和不带截距项
summary(l1 <- lm(birthw ~ sex + age), correlation = TRUE)
```

```
##
## Call:
## lm(formula = birthw ~ sex + age)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -257.49 -125.28  -58.44  169.00  303.98
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1610.28     786.08  -2.049   0.0532 .
## sexF         -163.04      72.81  -2.239   0.0361 *
## age           120.89      20.46   5.908 7.28e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 177.1 on 21 degrees of freedom
## Multiple R-squared:   0.64,  Adjusted R-squared:  0.6057
## F-statistic: 18.67 on 2 and 21 DF,  p-value: 2.194e-05
##
## Correlation of Coefficients:
##      (Intercept) sexF
## sexF  0.07
## age  -1.00       -0.12
```

```r
anova(l1)
```

```
## Analysis of Variance Table
##
## Response: birthw
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## sex        1   76163   76163  2.4279    0.1341
## age        1 1094940 1094940 34.9040 7.284e-06 ***
## Residuals 21  658771   31370
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# 与带交互项的模型比较
summary(li <- lm(birthw ~ sex + sex:age), correlation = TRUE)
```

```
##
## Call:
## lm(formula = birthw ~ sex + sex:age)
##
```

©南京大学出版社

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -246.69 -138.11  -39.13  176.57  274.28
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1268.67    1114.64  -1.138 0.268492
## sexF         -872.99    1611.33  -0.542 0.593952
## sexM:age      111.98      29.05   3.855 0.000986 ***
## sexF:age      130.40      30.00   4.347 0.000313 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 180.6 on 20 degrees of freedom
## Multiple R-squared:  0.6435, Adjusted R-squared:   0.59
## F-statistic: 12.03 on 3 and 20 DF,  p-value: 0.000101
##
## Correlation of Coefficients:
##          (Intercept) sexF  sexM:age
## sexF     -0.69
## sexM:age -1.00        0.69
## sexF:age  0.00       -0.72  0.00
```

```r
anova(li, l1)
```

```
## Analysis of Variance Table
##
## Model 1: birthw ~ sex + sex:age
## Model 2: birthw ~ sex + age
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     20 652425
## 2     21 658771 -1   -6346.2 0.1945 0.6639
```

```r
# 类似，只是使用 glm 命令来拟合而已
summary(zi <- glm(birthw ~ sex + age, family = gaussian()))
```

```
##
## Call:
## glm(formula = birthw ~ sex + age, family = gaussian())
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -257.49  -125.28   -58.44   169.00   303.98
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1610.28     786.08  -2.049   0.0532 .
```

©黄湘云

```
## sexF         -163.04      72.81  -2.239   0.0361 *
## age           120.89      20.46   5.908 7.28e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 31370.04)
##
##     Null deviance: 1829873  on 23  degrees of freedom
## Residual deviance:  658771  on 21  degrees of freedom
## AIC: 321.39
##
## Number of Fisher Scoring iterations: 2
```

```
anova(zi)
```

```
## Analysis of Deviance Table
##
## Model: gaussian, link: identity
##
## Response: birthw
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev
## NULL                    23    1829873
## sex   1    76163        22    1753711
## age   1  1094940        21     658771
```

```
# summary(z.o4 <- update(zi, subset = -4))
summary(zz <- update(zi, birthw ~ sex + age + sex:age))
```

```
##
## Call:
## glm(formula = birthw ~ sex + age + sex:age, family = gaussian())
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -246.69  -138.11   -39.13   176.57   274.28
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1268.67    1114.64  -1.138 0.268492
## sexF         -872.99    1611.33  -0.542 0.593952
## age           111.98      29.05   3.855 0.000986 ***
## sexF:age       18.42      41.76   0.441 0.663893
## ---
```

© 基础统计


```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 32621.23)
##
##     Null deviance: 1829873  on 23  degrees of freedom
## Residual deviance:  652425  on 20  degrees of freedom
## AIC: 323.16
##
## Number of Fisher Scoring iterations: 2
```

```
anova(zi, zz)
```

```
## Analysis of Deviance Table
##
## Model 1: birthw ~ sex + age
## Model 2: birthw ~ sex + age + sex:age
##   Resid. Df Resid. Dev Df Deviance
## 1        21     658771
## 2        20     652425  1   6346.2
```

26.21 火炬松树的生长情况

表 26.3 记录了 14 颗火炬树种子的生长情况

```
reshape(Loblolly, idvar = "Seed", timevar = "age",
        v.names = "height", direction = "wide", sep = "") %>%
  knitr::kable(.,
    caption = "火炬松树的高度（英尺）随时间（年）的变化",
    row.names = FALSE, col.names = gsub("(height)", "", names(.)),
    align = "c"
  )
```

图 26.5 火炬树种子基本决定了树的长势，不同种子预示最后的高度，并且在生长期也是很稳定地生长

```
p <- ggplot(data = Loblolly, aes(x = age, y = height, color = Seed)) +
  geom_point() +
  geom_line() +
  theme_minimal() +
  labs(x = "age (yr)", y = "height (ft)")
p
```

```
library(gganimate)
p + transition_reveal(age)
```

© 黄湘云

表 26.3: 火炬松树的高度（英尺）随时间（年）的变化

Seed	3	5	10	15	20	25
301	4.51	10.89	28.72	41.74	52.70	60.92
303	4.55	10.92	29.07	42.83	53.88	63.39
305	4.79	11.37	30.21	44.40	55.82	64.10
307	3.91	9.48	25.66	39.07	50.78	59.07
309	4.81	11.20	28.66	41.66	53.31	63.05
311	3.88	9.40	25.99	39.55	51.46	59.64
315	4.32	10.43	27.16	40.85	51.33	60.07
319	4.57	10.57	27.90	41.13	52.43	60.69
321	3.77	9.03	25.45	38.98	49.76	60.28
323	4.33	10.79	28.97	42.44	53.17	61.62
325	4.38	10.48	27.93	40.20	50.06	58.49
327	4.12	9.92	26.54	37.82	48.43	56.81
329	3.93	9.34	26.08	37.79	48.31	56.43
331	3.46	9.05	25.85	39.15	49.12	59.49

图 26.5: 不同火炬树的生长情况

© 黄湘云

26.22 酶促反应的反应速率

Puromycin 酶促反应的反应速度，模型拟合 `?SSmicmen`

```
ggplot(data = Puromycin, aes(x = conc, y = rate, color = state)) +
  geom_point() +
  geom_line() +
  theme_minimal() +
  labs(
    x = "Substrate concentration (ppm)",
    y = "Reaction velocity (counts/min/min)",
    title = "Puromycin data and fitted Michaelis-Menten curves"
  )
```

© 黄湘云

26.23 茶碱的药代动力学

```
ggplot(data = Theoph, aes(x = Time, y = conc, color = Subject)) +
  geom_point() +
  geom_line() +
  facet_wrap(Wt ~ Dose, ncol = 3, labeller = "label_both") +
  theme_minimal() +
  labs(
    x = "Time since drug administration (hr)",
    y = "Theophylline concentration (mg/L)",
    title = "Observed concentrations and fitted model"
  )
```

© 基础统计

```
Theoph %>%
  transform(., wt_dose = paste(Wt, Dose, sep = "~")) %>%
  ggplot(., aes(x = Time, y = conc, color = wt_dose)) +
  geom_point() +
  geom_line() +
  theme_minimal() +
  labs(
    x = "Time since drug administration (hr)",
    y = "Theophylline concentration (mg/L)",
    title = "Observed concentrations and fitted model"
  )
```

© 黄湘云

```
ggplot(data = Theoph, aes(x = Time, y = conc, color = Subject)) +
  geom_point() +
  geom_line() +
  theme_minimal() +
  labs(
    x = "Time since drug administration (hr)",
    y = "Theophylline concentration (mg/L)",
    title = "Observed concentrations and fitted model"
  )
```

©黄湘云

26.24 本章总结

模型永远没完，总是需要自己去构造符合自己需求的模型及其实现，只有自己能够实现，才能在海洋中遨游

> This is a bit like asking how should I tweak my sailboat so I can explore the ocean floor.
>
> — Roger Koenker [5]

26.25 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
```

[5] https://stat.ethz.ch/pipermail/r-help/2013-May/354311.html

© 黄湘云


```
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] patchwork_1.1.1 gganimate_1.0.7 ggplot2_3.3.5   magrittr_2.0.3
##
## loaded via a namespace (and not attached):
##  [1] progress_1.2.2    tidyselect_1.1.2  xfun_0.30         purrr_0.3.4
##  [5] lattice_0.20-45   splines_4.1.3     colorspace_2.0-3  vctrs_0.4.0
##  [9] generics_0.1.2    htmltools_0.5.2   viridisLite_0.4.0 yaml_2.3.5
## [13] mgcv_1.8-40       utf8_1.2.2        rlang_1.0.2       pillar_1.7.0
## [17] glue_1.6.2        withr_2.5.0       DBI_1.1.2         tweenr_1.0.2
## [21] plyr_1.8.7        lifecycle_1.0.1   stringr_1.4.0     munsell_0.5.0
## [25] gtable_0.3.0      codetools_0.2-18  evaluate_0.15     labeling_0.4.2
## [29] knitr_1.38        fastmap_1.1.0     curl_4.3.2        fansi_1.0.3
## [33] gifski_1.4.3-1    Rcpp_1.0.8.3      scales_1.1.1      sysfonts_0.8.8
## [37] farver_2.1.0      hms_1.1.1         digest_0.6.29     stringi_1.7.6
## [41] bookdown_0.25     dplyr_1.0.8       grid_4.1.3        cli_3.2.0
## [45] tools_4.1.3       tibble_3.1.6      crayon_1.5.1      pkgconfig_2.0.3
## [49] Matrix_1.4-1      ellipsis_0.3.2    prettyunits_1.1.1 assertthat_0.2.1
## [53] rmarkdown_2.13    R6_2.5.1          nlme_3.1-157      compiler_4.1.3
```

© 王建国

第二十七章　广义线性模型

> It's not meant for sampling weights. It's meant for precision weights. How best to include sampling weights in mixed models is a research problem at the moment, but you can rely on getting the wrong answer if you just use the `weights =` argument.
>
> — Thomas Lumley [1]

一般广义线性模型理论参考文献 An Introduction to Generalized Linear Models [Dobson and Barnett, 2018] 和 Generalized Linear Models [McCullagh and Nelder, 1989]，逻辑回归模型主要参考 Applied Logistic Regression [Hosmer and Lemeshow, 2000] 和 Discrete Choice Methods with Simulation [Train, 2009]。

简单线性模型（Linear Models，简称 LM），`stats::lm()` 函数可以拟合线性模型，而一般线性模型（General Linear Models，简称 GLM）允许线性模型方差非齐性、存在相关关系，甚至可以扩展到线性混合效应模型，将线性回归模型，方差分析模型，协方差分析模型统一地看待，一般要采用广义最小二乘（Generalized Least Squares，简称 GLS）拟合，`nlme::gls()` 函数实现广义最小二乘拟合线性模型，类似地，`nlme::gnls()` 函数实现广义最小二乘拟合非线性模型。`glm2::glm2()` 补充 `glm()`，提供更加稳定的拟合方法，适应于 `glm()` 不收敛的情况，而 `fastglm::fastglm()` 主要是加快 `glm()` 求解效率，收敛效果也比 `glm()` 和 `glm2()` 好。

glmnet 包是处理广义线性模型的事实标准。其官网见 https://glmnet.stanford.edu/，而 glmnetUtils 补充公式接口，适用于弹性网络回归，交叉验证筛选 α 参数等。glmpath 包实现 path-following 算法用于带 L1 正则项的广义线性模型和 Cox 比例风险模型。Boom 和 **BoomSpikeSlab** 包实现 MCMC 算法用于 Spike 和 Slab 回归，而 **spikeslab** 包进一步实现预测和变量选择 [Ishwaran and Rao, 2005]。Cyclops 包实现 Cyclic coordinate descent 算法用于逻辑回归、泊松回归和生存分析，适用于大规模正则回归 large scale regularized regressions，达到百万级别的观测和特征变量，交叉验证自动选择超参数，独立变量稀疏表示，用剖面似然估计某个变量的置信区间。plsRglm 包实现偏最小二乘回归方法用于广义线性模型。**biglm**、speedglm 和 bigReg 用于处理大数据集的回归，求解限制内存的 GLM biglmm。cglm 估计带聚类数据的条件 GLM 的回归系数和发散参数。MGLM 拟合多个响应变量的广义线性回归模型（多重 GLM）。robmixglm 响应变量扩展到混合分布的情形，实现稳健 GLM 回归估计。ClusterBootstrap 实现自主法估计带聚类数据的 GLM。lcpm 和 oglmx 处理有序输出的回归。gmnl、mlogit [Train, 2009] 和 **mnlogit** [Hasan et al., 2016] 处理多项逻辑回归。pscl 包（Political Science Computational Laboratory）可以处理贝叶斯 IRT 模型，zero-inflated 零膨胀模型，广义线性模型的拟合优度度量。

27.1　介绍

模型结构，模型种类，参数估计办法，相当于综述

[1] https://stat.ethz.ch/pipermail/r-help/2012-January/301501.html

© 基础

响应变量分别服从二项分布、多项分布、对数正态分布、泊松分布、伽马分布

27.2 理论基础

分两个段落分别介绍指数族和 GLM

$$f(y;\theta,\phi) = \exp[(a(y)b(\theta) + c(\theta))/f(\phi) + d(y,\phi)]$$

泊松分布 (with $\lambda \to \theta$, $x \to y$) ($\phi = 1$):

$$\begin{aligned} f(y,\theta) &= \exp(-\theta)\theta^y/(y!) \\ &= \exp\left(\underbrace{y}_{a(y)}\underbrace{\log\theta}_{b(\theta)} + \underbrace{(-\theta)}_{c(\theta)} + \underbrace{(-\log(y!))}_{d(y)}\right) \end{aligned} \tag{27.1}$$

27.2.1 岭回归

Geometry and properties of generalized ridge regression in high dimensions http://web.ccs.miami.edu/~hishwaran/papers/IR.conmath2014.pdf

这篇文章借助三维几何图形展示高维情形下的广义岭回归

27.2.2 Lasso

glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models https://glmnet.stanford.edu

27.2.3 最优子集回归

bestglm: Best Subset GLM and Regression Utilities

27.2.4 偏最小二乘回归

pls 包 [Mevik and Wehrens, 2007] 实现了偏最小二乘回归（partial least squares regression，PLS）和主成分回归（principal component regression，PCR），详见主页 https://mevik.net/work/software/pls.html 帮助文档的质量较高，是比较完整全面的。

- several algorithms: the traditional orthogonal scores (NIPALS) PLS algorithm, kernel PLS, wide kernel PLS, Simpls and PCR through svd
- supports multi-response models (aka PLS2)
- flexible cross-validation
- Jackknife variance estimates of regression coefficients
- extensive and flexible plots: scores, loadings, predictions, coefficients, (R)MSEP, R^2, correlation loadings

©南京大学

- formula interface, modelled after lm(), with methods for predict, print, summary, plot, update, etc.
- extraction functions for coefficients, scores and loadings
- MSEP, RMSEP and R^2 estimates
- multiplicative scatter correction (MSC)

27.3 吸烟喝酒和食道癌的关系

存在有序分类数据

酒精的作用 effects of alcohol, tobacco and interaction, age-adjusted 数据集描述见 `help(esoph)`

```
head(esoph)
```

```
##   agegp     alcgp    tobgp ncases ncontrols
## 1 25-34 0-39g/day 0-9g/day      0        40
## 2 25-34 0-39g/day    10-19      0        10
## 3 25-34 0-39g/day    20-29      0         6
## 4 25-34 0-39g/day      30+      0         5
## 5 25-34     40-79 0-9g/day      0        27
## 6 25-34     40-79    10-19      0         7
```

```
str(esoph)
```

```
## 'data.frame':    88 obs. of  5 variables:
##  $ agegp    : Ord.factor w/ 6 levels "25-34"<"35-44"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ alcgp    : Ord.factor w/ 4 levels "0-39g/day"<"40-79"<..: 1 1 1 1 2 2 2 2 3 3 ...
##  $ tobgp    : Ord.factor w/ 4 levels "0-9g/day"<"10-19"<..: 1 2 3 4 1 2 3 4 1 2 ...
##  $ ncases   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ ncontrols: num  40 10 6 5 27 7 4 7 2 1 ...
```

```
p1 <- ggplot(data = esoph, aes(x = agegp, y = ncases / (ncases + ncontrols), color = agegp)) +
  geom_boxplot(show.legend = FALSE) +
  geom_jitter(show.legend = FALSE) +
  theme_minimal()

p2 <- ggplot(data = esoph, aes(x = alcgp, y = ncases / ncontrols, color = alcgp)) +
  geom_boxplot(show.legend = FALSE) +
  geom_jitter(show.legend = FALSE) +
  theme_minimal()

p3 <- ggplot(data = esoph, aes(x = tobgp, y = ncases / ncontrols, color = tobgp)) +
  geom_boxplot(show.legend = FALSE) +
  geom_jitter(show.legend = FALSE) +
  theme_minimal()

bottom_row <- plot_grid(p2, p3, labels = c('B', 'C'), label_size = 12)
```

©黄湘云


```
## Warning: Removed 12 rows containing non-finite values (stat_boxplot).

## Warning: Removed 12 rows containing missing values (geom_point).

## Warning: Removed 12 rows containing non-finite values (stat_boxplot).

## Warning: Removed 12 rows containing missing values (geom_point).
```

```
plot_grid(p1, bottom_row, labels = c('A', ''), label_size = 12, ncol = 1)
```

图 27.1: 吸烟喝酒和食道癌的关系

```
fit_esoph_glm <- glm(cbind(ncases, ncontrols) ~ agegp + tobgp * alcgp,
  data = esoph, family = binomial(link = "logit")
)
```

```
library(Rcpp)
fit_esoph_brm <- brms::brm(ncases | trials(ncases + ncontrols) ~ agegp + tobgp * alcgp,
  data = esoph, family = binomial(link = "logit"), refresh = 0
)
```

27.4 自然流产和人工流产后的不育

```
help(infert)
```

```
head(infert)
```

```
##   education age parity induced case spontaneous stratum pooled.stratum
```

©李漫 仅供学习

```
## 1    0-5yrs  26      6       1    1           2       1              3
## 2    0-5yrs  42      1       1    1           0       2              1
## 3    0-5yrs  39      6       2    1           0       3              4
## 4    0-5yrs  34      4       2    1           0       4              2
## 5   6-11yrs  35      3       1    1           1       5             32
## 6   6-11yrs  36      4       2    1           1       6             36
```

```
str(infert)
```

```
## 'data.frame':    248 obs. of  8 variables:
##  $ education     : Factor w/ 3 levels "0-5yrs","6-11yrs",..: 1 1 1 1 2 2 2 2 2 2 ...
##  $ age           : num  26 42 39 34 35 36 23 32 21 28 ...
##  $ parity        : num  6 1 6 4 3 4 1 2 1 2 ...
##  $ induced       : num  1 1 2 2 1 2 0 0 0 0 ...
##  $ case          : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ spontaneous   : num  2 0 0 0 1 1 0 0 1 0 ...
##  $ stratum       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ pooled.stratum: num  3 1 4 2 32 36 6 22 5 19 ...
```

存在无序分类变量

```
infert_glm_1 <- glm(case ~ spontaneous + induced,
  data = infert, family = binomial()
)
summary(infert_glm_1)
```

```
##
## Call:
## glm(formula = case ~ spontaneous + induced, family = binomial(),
##     data = infert)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6678  -0.8360  -0.5772   0.9030   1.9362
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.7079     0.2677  -6.380 1.78e-10 ***
## spontaneous   1.1972     0.2116   5.657 1.54e-08 ***
## induced       0.4181     0.2056   2.033    0.042 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 316.17  on 247  degrees of freedom
## Residual deviance: 279.61  on 245  degrees of freedom
## AIC: 285.61
```

©黄湘云


```
##
## Number of Fisher Scoring iterations: 4
```

考虑其他潜在的因素

```
infert_glm_2 <- glm(case ~ age + parity + education + spontaneous + induced,
  data = infert, family = binomial()
)
summary(infert_glm_2)
```

```
##
## Call:
## glm(formula = case ~ age + parity + education + spontaneous +
##     induced, family = binomial(), data = infert)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7603  -0.8162  -0.4956   0.8349   2.6536
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.14924    1.41220  -0.814   0.4158
## age               0.03958    0.03120   1.269   0.2046
## parity           -0.82828    0.19649  -4.215 2.49e-05 ***
## education6-11yrs -1.04424    0.79255  -1.318   0.1876
## education12+ yrs -1.40321    0.83416  -1.682   0.0925 .
## spontaneous       2.04591    0.31016   6.596 4.21e-11 ***
## induced           1.28876    0.30146   4.275 1.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 316.17  on 247  degrees of freedom
## Residual deviance: 257.80  on 241  degrees of freedom
## AIC: 271.8
##
## Number of Fisher Scoring iterations: 4
```

实际上应该使用条件逻辑回归，调用 **survival** 包

```
library(survival)
infert_glm_3 <- clogit(case ~ spontaneous + induced + strata(stratum),
  data = infert
)
summary(infert_glm_3)
```

```
## Call:
```

©黄湘云

```
## coxph(formula = Surv(rep(1, 248L), case) ~ spontaneous + induced +
##     strata(stratum), data = infert, method = "exact")
##
##   n= 248, number of events= 83
##
##               coef exp(coef) se(coef)     z Pr(>|z|)
## spontaneous 1.9859    7.2854   0.3524 5.635 1.75e-08 ***
## induced     1.4090    4.0919   0.3607 3.906 9.38e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##             exp(coef) exp(-coef) lower .95 upper .95
## spontaneous     7.285     0.1373     3.651    14.536
## induced         4.092     0.2444     2.018     8.298
##
## Concordance= 0.776  (se = 0.044 )
## Likelihood ratio test= 53.15  on 2 df,   p=3e-12
## Wald test            = 31.84  on 2 df,   p=1e-07
## Score (logrank) test = 48.44  on 2 df,   p=3e-11
```

27.5 细菌数据集

流感嗜血杆菌的细菌与中耳炎患儿

```r
data(bacteria, package = "MASS")
```

```r
# 惩罚拟似然
fit_glmmpql <- MASS::glmmPQL(y ~ trt + I(week > 2),
  random = ~ 1 | ID, verbose = FALSE,
  family = binomial, data = bacteria
)
summary(fit_glmmpql)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: bacteria
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | ID
##         (Intercept)  Residual
## StdDev:    1.410637 0.7800511
##
## Variance function:
##  Structure: fixed weights
```

©清华大学出版社

```
##  Formula: ~invwt
## Fixed effects:  y ~ trt + I(week > 2)
##                     Value Std.Error  DF   t-value p-value
## (Intercept)      3.412014 0.5185033 169  6.580506  0.0000
## trtdrug         -1.247355 0.6440635  47 -1.936696  0.0588
## trtdrug+        -0.754327 0.6453978  47 -1.168779  0.2484
## I(week > 2)TRUE -1.607257 0.3583379 169 -4.485311  0.0000
##  Correlation:
##                 (Intr) trtdrg trtdr+
## trtdrug         -0.598
## trtdrug+        -0.571  0.460
## I(week > 2)TRUE -0.537  0.047 -0.001
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med         Q3        Max
## -5.1985361  0.1572336  0.3513075  0.4949482  1.7448845
##
## Number of Observations: 220
## Number of Groups: 50
```

```r
# 拉普拉斯近似
fit_glmer <- lme4::glmer(y ~ trt + I(week > 2) + (1 | ID),
  family = binomial, data = bacteria
)
summary(fit_glmer)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: y ~ trt + I(week > 2) + (1 | ID)
##    Data: bacteria
##
##      AIC      BIC   logLik deviance df.resid
##    202.3    219.2    -96.1    192.3      215
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5615  0.1359  0.3022  0.4217  1.1276
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  ID     (Intercept) 1.543    1.242
## Number of obs: 220, groups:  ID, 50
##
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)       3.5479     0.6958   5.099 3.41e-07 ***
## trtdrug          -1.3667     0.6770  -2.019 0.043516 *
## trtdrug+         -0.7826     0.6831  -1.146 0.251926
## I(week > 2)TRUE  -1.5985     0.4759  -3.359 0.000783 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) trtdrg trtdr+
## trtdrug     -0.593
## trtdrug+    -0.537  0.487
## I(wk>2)TRUE -0.656  0.126  0.064
```

27.6 研究婴儿出生体重低的相关危险因素

在线性回归的基础上，响应变量是离散的类别，且无序 [Hasan et al., 2016]

birthwt 数据是 1986 年在马萨诸塞州斯普林菲尔德的 Baystate 医疗中心收集的，用于研究婴儿出生体重低的相关危险因素

```r
# 加载数据
# library(MASS)
data(birthwt, package = "MASS")
# 查看 birthwt 数据集 `help(birthwt)`
head(birthwt)
```

```
##    low age lwt race smoke ptl ht ui ftv  bwt
## 85   0  19 182    2     0   0  0  1   0 2523
## 86   0  33 155    3     0   0  0  0   3 2551
## 87   0  20 105    1     1   0  0  0   1 2557
## 88   0  21 108    1     1   0  0  1   2 2594
## 89   0  18 107    1     1   0  0  1   0 2600
## 91   0  21 124    3     0   0  0  0   0 2622
```

```r
str(birthwt)
```

```
## 'data.frame':    189 obs. of  10 variables:
##  $ low  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ age  : int  19 33 20 21 18 21 22 17 29 26 ...
##  $ lwt  : int  182 155 105 108 107 124 118 103 123 113 ...
##  $ race : int  2 3 1 1 1 3 1 3 1 1 ...
##  $ smoke: int  0 0 1 1 1 0 0 0 1 1 ...
##  $ ptl  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ht   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ui   : int  1 0 0 1 1 0 0 0 0 0 ...
##  $ ftv  : int  0 3 1 2 0 0 1 1 1 0 ...
##  $ bwt  : int  2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...
```

© 基湘石

low 表示婴儿出生体重小于 2.5kg，age 表示母亲的年龄（年），lwt 母亲最后一次月经期间的体重 (磅)，race 母亲的种族 (1 = 白人，2 = 黑人，3 = 其他)。，smoke 怀孕期间的吸烟状况，ptl 以前早产的次数，ht 高血压病史，ui 子宫过敏，ftv 妊娠头三个月的医生就诊次数，bwt 出生体重（克）

```
with(birthwt, tapply(lwt, ui, var))
```

```
##        0        1
## 940.8472 783.7196
```

```
t.test(lwt ~ ui, data = birthwt, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  lwt by ui
## t = 2.1138, df = 187, p-value = 0.03586
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##   0.8753389 25.3544748
## sample estimates:
## mean in group 0 mean in group 1
##        131.7578        118.6429
```

```
t.test(lwt ~ ui, data = birthwt)
```

```
##
##  Welch Two Sample t-test
##
## data:  lwt by ui
## t = 2.2547, df = 39.163, p-value = 0.02982
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##   1.351128 24.878685
## sample estimates:
## mean in group 0 mean in group 1
##        131.7578        118.6429
```

```
# birthwt$ui <- as.factor(birthwt$ui)
# library(lattice)
# bwplot(lwt ~ ui, data = birthwt, pch = "|")

boxplot(lwt ~ ui, data = birthwt)
```

©黄湘云

```
# 重新编码，数据预处理，方便代入模型
bwt <- with(birthwt, {
  race <- factor(race, labels = c("white", "black", "other"))
  ptd <- factor(ptl > 0)
  ftv <- factor(ftv)
  levels(ftv)[-(1:2)] <- "2+" # 除了前两个水平外，其余的都编码为 2+
  data.frame(
    low = factor(low), age, lwt, race, smoke = (smoke > 0),
    ptd, ht = (ht > 0), ui = (ui > 0), ftv
  )
})
```

```
# 查看编码后的数据
head(bwt)
```

```
##   low age lwt  race smoke   ptd    ht    ui ftv
## 1   0  19 182 black FALSE FALSE FALSE  TRUE   0
## 2   0  33 155 other FALSE FALSE FALSE FALSE  2+
## 3   0  20 105 white  TRUE FALSE FALSE FALSE   1
## 4   0  21 108 white  TRUE FALSE FALSE  TRUE  2+
## 5   0  18 107 white  TRUE FALSE FALSE  TRUE   0
## 6   0  21 124 other FALSE FALSE FALSE FALSE   0
```

```
str(bwt)
```

```
## 'data.frame':    189 obs. of  9 variables:
```

©黄湘云

```
##  $ low  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ age  : int  19 33 20 21 18 21 22 17 29 26 ...
##  $ lwt  : int  182 155 105 108 107 124 118 103 123 113 ...
##  $ race : Factor w/ 3 levels "white","black",..: 2 3 1 1 1 3 1 3 1 1 ...
##  $ smoke: logi  FALSE FALSE TRUE TRUE TRUE FALSE ...
##  $ ptd  : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ht   : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ ui   : logi  TRUE FALSE FALSE TRUE TRUE FALSE ...
##  $ ftv  : Factor w/ 3 levels "0","1","2+": 1 3 2 3 1 1 2 2 2 1 ...
```

广义线性模型拟合，二项逻辑回归，响应变量为婴儿出生的体重，以 2.5kg 为界，它被编码成二分类变量 0 或 1

```
options(contrasts = c("contr.treatment", "contr.poly"))
glm(formula = low ~ ., family = binomial, data = bwt)
```

```
##
## Call:  glm(formula = low ~ ., family = binomial, data = bwt)
##
## Coefficients:
## (Intercept)          age          lwt    raceblack    raceother    smokeTRUE
##     0.82302     -0.03723     -0.01565      1.19241      0.74068      0.75553
##     ptdTRUE       htTRUE       uiTRUE         ftv1        ftv2+
##     1.34376      1.91317      0.68020     -0.43638      0.17901
##
## Degrees of Freedom: 188 Total (i.e. Null);  178 Residual
## Null Deviance:       234.7
## Residual Deviance: 195.5     AIC: 217.5
```

多项逻辑回归

```
library(nnet)
(bwt.mu <- multinom(formula = low ~ ., data = bwt))
```

```
## # weights:  12 (11 variable)
## initial  value 131.004817
## iter  10 value 98.029803
## final  value 97.737759
## converged
```

```
## Call:
## multinom(formula = low ~ ., data = bwt)
##
## Coefficients:
## (Intercept)         age         lwt   raceblack   raceother   smokeTRUE
##  0.82320102 -0.03723828 -0.01565359  1.19240391  0.74065606  0.75550487
##     ptdTRUE      htTRUE      uiTRUE         ftv1        ftv2+
##  1.34375901  1.91320116  0.68020207 -0.43638470  0.17900392
##
```

```
## Residual Deviance: 195.4755
## AIC: 217.4755
```

```
summary(bwt.mu)
```

```
## Call:
## multinom(formula = low ~ ., data = bwt)
##
## Coefficients:
##                  Values  Std. Err.
## (Intercept)  0.82320102 1.24476766
## age         -0.03723828 0.03870437
## lwt         -0.01565359 0.00708079
## raceblack    1.19240391 0.53598076
## raceother    0.74065606 0.46176615
## smokeTRUE    0.75550487 0.42503626
## ptdTRUE      1.34375901 0.48063449
## htTRUE       1.91320116 0.72076133
## uiTRUE       0.68020207 0.46434974
## ftv1        -0.43638470 0.47941107
## ftv2+        0.17900392 0.45639129
##
## Residual Deviance: 195.4755
## AIC: 217.4755
```

计算 Z 分数和 P 值

```
z <- summary(bwt.mu)$coefficients / summary(bwt.mu)$standard.errors
z
```

```
## (Intercept)         age         lwt   raceblack   raceother   smokeTRUE
##   0.6613291  -0.9621210  -2.2107121   2.2247140   1.6039635   1.7775069
##     ptdTRUE      htTRUE      uiTRUE        ftv1       ftv2+
##   2.7958023   2.6544170   1.4648486  -0.9102516   0.3922159
```

```
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
## (Intercept)         age         lwt   raceblack   raceother   smokeTRUE
## 0.508401310 0.335988847 0.027055777 0.026100443 0.108722092 0.075484881
##     ptdTRUE      htTRUE      uiTRUE        ftv1       ftv2+
## 0.005177106 0.007944557 0.142962228 0.362689827 0.694898695
```

模型解释

27.7 哥本哈根住房状况调查

响应变量是离散类别，且存在强弱，等级，大小之分

调用函数 MASS::polr()

数据集 housing 哥本哈根住房状况调查中的次数分布表，Sat 住户对目前居住环境的满意程度，是一个有序的因子变量，Infl 住户对物业管理的感知影响程度，Type 租赁住宿类型，如塔楼、中庭、公寓、露台，Cont 联系居民可与其他居民联系 (低、高)，Freq 每个类中的居民人数，调查的人数

```
data("housing", package = "MASS")
# 查看数据 help(housing)
head(housing)
```

```
##      Sat   Infl  Type Cont Freq
## 1    Low    Low Tower  Low   21
## 2 Medium    Low Tower  Low   21
## 3   High    Low Tower  Low   28
## 4    Low Medium Tower  Low   34
## 5 Medium Medium Tower  Low   22
## 6   High Medium Tower  Low   36
```

```
str(housing)
```

```
## 'data.frame':    72 obs. of  5 variables:
##  $ Sat : Ord.factor w/ 3 levels "Low"<"Medium"<..: 1 2 3 1 2 3 1 2 3 1 ...
##  $ Infl: Factor w/ 3 levels "Low","Medium",..: 1 1 1 2 2 2 3 3 3 1 ...
##  $ Type: Factor w/ 4 levels "Tower","Apartment",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ Cont: Factor w/ 2 levels "Low","High": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Freq: int  21 21 28 34 22 36 10 11 36 61 ...
```

居民对居住环境满意度 Sat 三个等级的有序回归

```
options(contrasts = c("contr.treatment", "contr.poly"))
house.plr <- MASS::polr(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
house.plr
```

```
## Call:
## MASS::polr(formula = Sat ~ Infl + Type + Cont, data = housing,
##     weights = Freq)
##
## Coefficients:
##    InflMedium      InflHigh TypeApartment    TypeAtrium   TypeTerrace
##     0.5663937     1.2888191    -0.5723501    -0.3661866    -1.0910149
##      ContHigh
##     0.3602841
##
## Intercepts:
##  Low|Medium Medium|High
##  -0.4961353   0.6907083
##
## Residual Deviance: 3479.149
## AIC: 3495.149
```

再计算一下 P 值，置信区间

```
ctable <- coef(summary(house.plr))
p <- pnorm(abs(ctable[, "t value"]), lower.tail = FALSE) * 2
ctable <- cbind(ctable, "p value" = p)
# confidence intervals 计算置信区间
ci <- confint(house.plr)
exp(coef(house.plr))
```

```
##    InflMedium      InflHigh TypeApartment    TypeAtrium   TypeTerrace
##     1.7619017     3.6284990     0.5641979     0.6933734     0.3358754
##      ContHigh
##     1.4337368
```

```
## OR and CI
exp(cbind(OR = coef(house.plr), ci))
```

```
##                      OR     2.5 %    97.5 %
## InflMedium    1.7619017 1.4356845 2.1639915
## InflHigh      3.6284990 2.8319659 4.6626461
## TypeApartment 0.5641979 0.4462124 0.7121941
## TypeAtrium    0.6933734 0.5114084 0.9398410
## TypeTerrace   0.3358754 0.2492277 0.4514276
## ContHigh      1.4337368 1.1892931 1.7296674
```

模型解释

参考文档 `help(housing)` 包含泊松回归、多项回归、比例风险模型，以及 https://www.analyticsvidhya.com/blog/2016/02/multinomial-ordinal-logistic-regression/

好好看文档 `help(housing)` 和对应的参考书籍，把原理弄清楚

有序因子变量是如何实现编码的

27.8 癫痫病发作次数

纵向数据 [Thall and Vail, 1990]，考虑了过度发散 overdispersion 异方差 heteroscedasticity 观测不独立

数据集 epil 记录癫痫发作的次数及病人的特征，下面是数据建模分析过程

```
data(epil, package = "MASS")
fit_glm_epil <- glm(y ~ lbase * trt + lage + V4,
  family = poisson,
  data = epil
)
summary(fit_glm_epil)

fit_glmm_epil<- MASS::glmmPQL(y ~ lbase * trt + lage + V4,
  random = ~ 1 | subject,
  family = poisson, data = epil
```

© 草稿


```
)
summary(fit_glmm_epil)

fit_glmm_lme4 <- lme4::glmer(y ~ lbase * trt + lage + V4 + (1 | subject),
  family = poisson, data = epil
)
summary(fit_glmm_lme4)


fit_glmm_glmmtmb <- glmmTMB::glmmTMB(y ~ lbase * trt + lage + V4 + (1 | subject),
  data = epil, family = poisson, REML = TRUE
) # REML 估计
summary(fit_glmm_glmmtmb)


# https://github.com/drizopoulos/GLMMadaptive
fit_glmm_glmmadaptive <- GLMMadaptive::mixed_model(
  fixed = y ~ lbase * trt + lage + V4,
  random = ~ 1 | subject, data = epil,
  family = poisson()
)
summary(fit_glmm_glmmadaptive)
```

27.9 对数线性模型

当响应变量 Y 服从对数正态分布的时候，广义线性模型具化为对数线性模型，**gllm** 包 [Espeland and Hui, 1987]

27.10 泊松回归模型

加载数据

```
data(beall.webworms, package = "agridat")
```

查看数据

```
head(beall.webworms)
```

```
##   row col y block trt spray lead
## 1   1   1 1    B1  T1     N    N
## 2   2   1 0    B1  T1     N    N
## 3   3   1 1    B1  T1     N    N
## 4   4   1 3    B1  T1     N    N
## 5   5   1 6    B1  T1     N    N
## 6   6   1 0    B2  T1     N    N
```

© 北京师范大学出版社

描述响应变量的分布

```
hist(beall.webworms$y, main = "Histogram of Worm Count", xlab = "Number of Worms")
```

```
boxplot(y ~ trt, data = beall.webworms)
```

© 黄湘云

抖动图

```
plot(y ~ trt, data = beall.webworms, xlab = "Types of Worms", ylab = "Worm Count")
```

©黄湘云

```
ggplot(beall.webworms, aes(trt, y)) +
  geom_boxplot(colour = "red") +
  geom_jitter() +
  labs(x = "Types of Worms", y = "Worm Count") +
  theme_minimal()
```

```
pois.mod <- glm(y ~ trt, data = beall.webworms, family = "poisson")
summary(pois.mod)
```

```
##
## Call:
## glm(formula = y ~ trt, family = "poisson", data = beall.webworms)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6733  -1.0046  -0.9081   0.6141   4.2771
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.33647    0.04688   7.177 7.12e-13 ***
## trtT2       -1.02043    0.09108 -11.204  < 2e-16 ***
## trtT3       -0.49628    0.07621  -6.512 7.41e-11 ***
## trtT4       -1.22246    0.09829 -12.438  < 2e-16 ***
## ---
```

仅供学习 ©


```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1955.9  on 1299  degrees of freedom
## Residual deviance: 1720.4  on 1296  degrees of freedom
## AIC: 3125.5
##
## Number of Fisher Scoring iterations: 6
```

模型系数 T2 的解释，这里 GLM 使用了对数联系函数 (log link function)，因此 -1.02 是对数变换后的值，T2 的系数实际是 0.3605949，实际意义是相对于 T1，T2 类型的蠕虫数量是 T1 的 0.3605949 倍

The first valuable information is related to the residuals of the model, which should be symmetrical as for any normal linear model. From this output we can see that minimum and maximum, as well as the first and third quartiles, are similar, so this assumption is confirmed. Then we can see that the variable trt (i.e. treatment factor) is highly significant for the model, with very low p-values. The statistical test in this case is not a t-test, as in the output of the function lm, but a Wald Test (Wald Test). This test computes the probability that the coefficient is 0, if the p is significant it means the chances the coefficient is zero are very low so the variable should be included in the model since it has an effect on y.

Another important information is the deviance, particularly the residual deviance. As a general rule, this value should be lower or in line than the residuals degrees of freedom for the model to be good. In this case the fact that the residual deviance is high (even though not dramatically) may suggests the explanatory power of the model is low. We will see below how to obtain p-value for the significance of the model.

```
par(mfrow = c(2, 2))
plot(pois.mod)
```

© 黄湘云

```
predict(pois.mod, newdata = data.frame(trt = c("T1", "T2")))
```

```
##          1          2
##  0.3364722 -0.6839588
```

模型的 P 值

```
1 - pchisq(deviance(pois.mod), df.residual(pois.mod))
```

```
## [1] 1.709743e-14
```

模型选择

```
pois.mod2 <- glm(y ~ block + spray * lead, data = beall.webworms, family = "poisson")
```

两模型的 AIC 比较

```
AIC(pois.mod, pois.mod2)
```

```
##           df      AIC
## pois.mod   4 3125.478
## pois.mod2 16 3027.438
```

假设响应变量 Y 服从泊松分布，意味着随机变量 Y 的期望和方差相等

```
mean(beall.webworms$y)
```

```
## [1] 0.7923077
```

```
var(beall.webworms$y)
```

```
## [1] 1.290164
```

©黄湘云

实际上方差比均值大，这种情况称之为过度发散 (overdispersed)，分布应该修正为拟（似然）泊松分布

```
pois.mod3 <- glm(y ~ trt, data = beall.webworms, family = c("quasipoisson"))
summary(pois.mod3)
```

```
##
## Call:
## glm(formula = y ~ trt, family = c("quasipoisson"), data = beall.webworms)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6733  -1.0046  -0.9081   0.6141   4.2771
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.33647    0.05457   6.166 9.32e-10 ***
## trtT2       -1.02043    0.10601  -9.626  < 2e-16 ***
## trtT3       -0.49628    0.08870  -5.595 2.69e-08 ***
## trtT4       -1.22246    0.11440 -10.686  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.35472)
##
##     Null deviance: 1955.9  on 1299  degrees of freedom
## Residual deviance: 1720.4  on 1296  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

计算得知发散参数 (dispersion parameter) 是 1.35472，可见数据 Y 并不是发散得离谱，泊松分布可能仍然是对这个数据的合理假设

AER 包是书籍 Applied Econometrics with R 的配套材料 [Kleiber and Zeileis, 2008]，可用于直接检验发散参数是否大于 1

```
# AER::dispersiontest(pois.mod, alternative="greater")
```

如果数据真的过度离散，就应该使用负二项分布作为响应变量的拟合分布，拟合它就采用 MASS 包 [Venables and Ripley, 2002] 提供的 `glm.nb` 函数

```
NB.mod1 <- MASS::glm.nb(y ~ trt, data = beall.webworms)
summary(NB.mod1)
```

```
##
## Call:
## MASS::glm.nb(formula = y ~ trt, data = beall.webworms, init.theta = 2.004130573,
##     link = log)
##
```

© 统计学

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4572  -0.9488  -0.8660   0.5340   2.7698
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.33647    0.06110   5.507 3.65e-08 ***
## trtT2       -1.02043    0.10661  -9.572  < 2e-16 ***
## trtT3       -0.49628    0.09423  -5.267 1.39e-07 ***
## trtT4       -1.22246    0.11283 -10.834  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(2.0041) family taken to be 1)
##
##     Null deviance: 1442.7  on 1299  degrees of freedom
## Residual deviance: 1275.3  on 1296  degrees of freedom
## AIC: 3053
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  2.004
##           Std. Err.:  0.325
##
##  2 x log-likelihood:  -3042.969
```

两个模型的方差分析

```
anova(pois.mod, pois.mod2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ trt
## Model 2: y ~ block + spray * lead
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      1296     1720.4
## 2      1284     1598.4 12   122.04 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

从方差分析比较的结果来看，P 值告诉我们，两模型是显著不同的，由上面对两模型的 AIC 计算结果来看，模型 pois.mod2 比模型 pois.mod 要好，模型的 AIC 值越小，表明拟合得越准确。

© 黄湘云

第七部分

数据建模

© 吉林大学

介绍

数据建模

© 王建国

第二十八章　文本分析

Supervised Machine Learning for Text Analysis in R 和 Tidy Text Mining with R

PDFR 和 pdftools 从 PDF 文档抽取文本，tesseract 从扫描件中抽取文本

quanteda

fastTextR https://github.com/facebookresearch/fastText

© 王湘君

第二十九章　生存分析

> The fact that some people murder doesn't mean we should copy them. And murdering data, though not as serious, should also be avoided.
>
> — Frank E. Harrell [1]

R 软件内置了 **survival** 包 它是实现生存分析的核心 R 包。文档见 https://cran.r-project.org/package=survival 相关书籍见 Terry M. Therneau and Patricia M. Grambsch [2000]

survminer 竟然严重依赖 **ggpubr** 包，ggpubr 包曾被 ggtree 的作者余光创严重吐槽过。**ggfortify** 包 大大扩展了 **ggplot2** 包的 `autoplot()` 函数，使得它适应各种模型对象的自动绘图。

29.1　急性粒细胞白血病生存数据

```
library(survival)
leukemia.surv <- survfit(Surv(time, status) ~ x, data = aml)
library(ggfortify)
autoplot(leukemia.surv, data = aml) +
  theme_minimal()
```

[1] https://stat.ethz.ch/pipermail/r-help/2005-July/075649.html

© 黄湘云

图 29.1: 急性粒细胞白血病生存数据

© 黄湘云

第三十章　时序分析

```
library(formatR)
`%>%` <- magrittr::`%>%`
library(ggplot2)
library(ggfortify) # **ggfortify** 包提供的 `autoplot()` 函数可以根据数据对象的不同绘制不同的图形。
library(dygraphs)
# library(robustbase) # Robust Statistics
# library(timeDate) # 日期处理
# library(timeSeries) # 序列处理
# library(fPortfolio) # 投资组合
# library(prophet) # 时间序列预测
# https://github.com/business-science/timetk
# library(timetk) # 处理时间序列数据的工具箱
```

首先介绍时序数据对象及操作，处理时序数据的工具，包括时序图、相关图、平稳性检验，相关检验，之后才是时序建模。timeDate timeSeries 是处理日期和时间序列的 R 包，有专门的官网 https://www.rmetrics.org/，扩展到时间序列、组合优化、金融市场、投资管理等一系列书籍，非常值得一看。此外，北大李东风老师的金融时间序列分析讲义是这方面非常好的中文参考材料。David R. Brillinger 在 1975 年出版的书《Time Series: Data Analysis and Theory》[Brillinger, 2001] 是经典著作，我们可以从时间序列分析的综述上开始入手，比如从 ARIMA 过渡到异方差和非高斯分布 https://mason.gmu.edu/~jgentle/talks/CompFin_Tutorial.pdf，https://www.stat.berkeley.edu/~brill/Papers/encysbs.pdf 和 ARCH or GARCH 的综述 http://public.econ.duke.edu/~boller/Papers/glossary_arch.pdf，宾州州立大学开设的 Applied Time Series Analysis 课程 https://newonlinecourses.science.psu.edu/stat510/，以及《Time Series Analysis and Its Applications With R Examples》已经出到第四版了，和 R 语言结合，理论和应用结合 https://www.stat.pitt.edu/stoffer/tsa4/。从时间序列中寻找规律，这样才是真的数据建模，从数据到模型，而不是相反 Finding Patterns in Time Series，识别金融时间序列的模式和统计规律。现在工业界做时序分析和预测的工具，如 facebook 出品的 prophet，微软收集了一些时间序列预测的最佳实战案例 https://github.com/microsoft/forecasting

forecastML 自回归模型结合机器学习方法。

CausalImpact 借助贝叶斯分析方法推断时间序列中的因果关系，比如广告促销带来的点击效果。

robustbase [Maronna et al., 2006] 提供稳健统计方法。

prophet 基于可加模型的时间序列预测

AnomalyDetection 时间序列数据中的异常值检测

©黄湘云

30.1 时序数据

以数据集 AirPassengers 为例说明一下 R 内置的存储时间序列数据的数据结构 — ts 数据对象。函数 `class()` 、`mode()` 和 `str()` 分别可以查看其数据类型、存储类型和数据结构。

```
# 数据类型
class(AirPassengers)
```

```
## [1] "ts"
```

```
# 存储类型
mode(AirPassengers)
```

```
## [1] "numeric"
```

```
# 数据结构
str(AirPassengers)
```

```
##  Time-Series [1:144] from 1949 to 1961: 112 118 132 129 121 135 148 148 136 119 ...
```

查看该数据集开始和结束的时间点

```
c(start(AirPassengers), end(AirPassengers))
```

```
## [1] 1949    1 1960   12
```

数据集 AirPassengers 在以上时间区间的划分

```
time(AirPassengers)
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 1949 1949.000 1949.083 1949.167 1949.250 1949.333 1949.417 1949.500 1949.583
## 1950 1950.000 1950.083 1950.167 1950.250 1950.333 1950.417 1950.500 1950.583
## 1951 1951.000 1951.083 1951.167 1951.250 1951.333 1951.417 1951.500 1951.583
## 1952 1952.000 1952.083 1952.167 1952.250 1952.333 1952.417 1952.500 1952.583
## 1953 1953.000 1953.083 1953.167 1953.250 1953.333 1953.417 1953.500 1953.583
## 1954 1954.000 1954.083 1954.167 1954.250 1954.333 1954.417 1954.500 1954.583
## 1955 1955.000 1955.083 1955.167 1955.250 1955.333 1955.417 1955.500 1955.583
## 1956 1956.000 1956.083 1956.167 1956.250 1956.333 1956.417 1956.500 1956.583
## 1957 1957.000 1957.083 1957.167 1957.250 1957.333 1957.417 1957.500 1957.583
## 1958 1958.000 1958.083 1958.167 1958.250 1958.333 1958.417 1958.500 1958.583
## 1959 1959.000 1959.083 1959.167 1959.250 1959.333 1959.417 1959.500 1959.583
## 1960 1960.000 1960.083 1960.167 1960.250 1960.333 1960.417 1960.500 1960.583
##           Sep      Oct      Nov      Dec
## 1949 1949.667 1949.750 1949.833 1949.917
## 1950 1950.667 1950.750 1950.833 1950.917
## 1951 1951.667 1951.750 1951.833 1951.917
## 1952 1952.667 1952.750 1952.833 1952.917
## 1953 1953.667 1953.750 1953.833 1953.917
## 1954 1954.667 1954.750 1954.833 1954.917
## 1955 1955.667 1955.750 1955.833 1955.917
```

©黄湘云

```
## 1956 1956.667 1956.750 1956.833 1956.917
## 1957 1957.667 1957.750 1957.833 1957.917
## 1958 1958.667 1958.750 1958.833 1958.917
## 1959 1959.667 1959.750 1959.833 1959.917
## 1960 1960.667 1960.750 1960.833 1960.917
```

期初和期末的周期

```
tsp(AirPassengers)
```

```
## [1] 1949.000 1960.917   12.000
```

函数 diff() 实现差分算子，默认参数 lag = 1 ，differences = 1 表示延迟期数为 1 的一阶差分。

```
# 差分前
AirPassengers
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```

```
# 差分后
diff(AirPassengers)
```

```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1949         6   14   -3   -8   14   13    0  -12  -17  -15   14
## 1950   -3   11   15   -6  -10   24   21    0  -12  -25  -19   26
## 1951    5    5   28  -15    9    6   21    0  -15  -22  -16   20
## 1952    5    9   13  -12    2   35   12   12  -33  -18  -19   22
## 1953    2    0   40   -1   -6   14   21    8  -35  -26  -31   21
## 1954    3  -16   47   -8    7   30   38   -9  -34  -30  -26   26
## 1955   13   -9   34    2    1   45   49  -17  -35  -38  -37   41
## 1956    6   -7   40   -4    5   56   39   -8  -50  -49  -35   35
## 1957    9  -14   55   -8    7   67   43    2  -63  -57  -42   31
## 1958    4  -22   44  -14   15   72   56   14 -101  -45  -49   27
## 1959   23  -18   64  -10   24   52   76   11  -96  -56  -45   43
## 1960   12  -26   28   42   11   63   87  -16  -98  -47  -71   42
```

```
# 延迟一期的二阶差分
diff(AirPassengers, lag = 1, differences = 2)
```

© 黄湘云


```
##       Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
## 1949                8  -17   -5   22   -1  -13  -12   -5    2   29
## 1950  -17   14    4  -21   -4   34   -3  -21  -12  -13    6   45
## 1951  -21    0   23  -43   24   -3   15  -21  -15   -7    6   36
## 1952  -15    4    4  -25   14   33  -23    0  -45   15   -1   41
## 1953  -20   -2   40  -41   -5   20    7  -13  -43    9   -5   52
## 1954  -18  -19   63  -55   15   23    8  -47  -25    4    4   52
## 1955  -13  -22   43  -32   -1   44    4  -66  -18   -3    1   78
## 1956  -35  -13   47  -44    9   51  -17  -47  -42    1   14   70
## 1957  -26  -23   69  -63   15   60  -24  -41  -65    6   15   73
## 1958  -27  -26   66  -58   29   57  -16  -42 -115   56   -4   76
## 1959   -4  -41   82  -74   34   28   24  -65 -107   40   11   88
## 1960  -31  -38   54   14  -31   52   24 -103  -82   51  -24  113
```

30.2 时序图

美国纽黑文自 1912 年至 1971 年的年平均气温变化见图 30.1。

```
plot(nhtemp, main = "美国纽黑文的年平均气温", family = "Noto Serif CJK SC")
```

图 30.1: 美国纽黑文的年平均气温，单位：华氏温度

```
# 构造多个 ts 序列
tmp <- ts(
```

© 黄湘云


```
  data = data.frame(
    pay_one = rnorm(20),
    pay_two = rnorm(20),
    pay_three = rnorm(20)
  ),
  start = c(1961, 1), frequency = 12
)

plot(tmp, main = "pay cnt")
```

```
plot(tmp, plot.type = "single", col = 1:3, ylab = "pay cnt")
```

© 黄湘云

30.3 基本概念

```
# 从某个完整的一天开始统计数据
# 分钟级 ts 数据
time_min <- format(seq.POSIXt(
  from = as.POSIXct("2020-01-01 00:00"),
  to = as.POSIXct("2020-01-01 23:59"), by = "1 min"
),
format = "%H:%M"
)
tmp = ts(data = rnorm(1440 * 3), start = c(2020, 12),
         frequency = 24*60*365.25, names = "访问量")
plot(tmp)
```

© 黄湘云

frequency: the number of observations per unit of time.

frequency 里面乘以 365.25 而不是 365 是因为每隔 4 年出现一次 366 天，多出的这一天分摊到每一年。frequency 表示单位时间内发生的次数，ts 对象的时间基准为 1 年，所以，frequency = 4 表示一年出现四次，依此类推。关于季节性周期的说明 https://robjhyndman.com/hyndsight/seasonal-periods/。

序列长度一样，但是周期不一样，这里的单位时间指的是 1 年

```
# 季数据
op = par(mfrow = c(2,2), mar = c(4,4,4,1))
tmp = ts(rnorm(20), start = c(1961, 1), frequency = 4)
plot(tmp, main = "Quarterly", type = "b")
# 月数据
tmp = ts(rnorm(20), start = c(1961, 1), frequency = 12) # 自然时间周期是一年，每月采样
plot(tmp, main = "Monthly", type = "b")
# 日数据
tmp = ts(rnorm(20), start = c(1961, 1), frequency = 365.25)
plot(tmp, main = "Daily", type = "b")
# 分钟数据
tmp = ts(rnorm(20), start = c(1961, 1), frequency = 24*60*365.25)
plot(tmp, main = "Minutely", type = "b")
```

© 黄湘云

```
par(op)
```

默认情况下，自然时间周期是一年，每月采样。那可不可以设置自然时间周期是一周，每天采样呢？当然可以，只是 Base R 暂不支持，其实表达数据粒度的能力没有变化，以年或周为基准，都可以表达上面的季、月、日、分钟数据。

deltat 和 frequency 只需提供一个参数值即可 deltat = 1/12 和 frequency = 12 表示同样的含义。

R 4.0.0 开始，frequency 不必是整数，还可以是小数，frequency = .2 表示每 5 个时间单位抽样一次，根据周期 T 和频率 f 的关系 T = 1/f

```
tmp = ts(rnorm(20), start = c(1961, 1), frequency = .2)
plot(tmp, type = "b")
```

© 黄湘云

ts 和 seq 构造时间向量的关系是什么?

```
seq(from = 1961, to = 2056, by = 5)
```

```
##  [1] 1961 1966 1971 1976 1981 1986 1991 1996 2001 2006 2011 2016 2021 2026 2031
## [16] 2036 2041 2046 2051 2056
```

即每隔 5 年抽样一次，采一个数据点

```
ts(rnorm(20), start = c(1961, 1), frequency = 365.25/7)
```

```
## Time Series:
## Start = 1961
## End = 1961.36413415469
## Frequency = 52.1785714285714
##  [1]  0.43860860 -1.34605535  0.27697324  0.26180691 -0.29466516  0.29652676
##  [7]  1.02064070 -0.61169443 -2.12240067 -0.55065855  1.79364046 -0.89443046
## [13]  0.19754451 -0.20348458 -0.03185203  0.82964623 -0.23954589 -0.07591583
## [19] -0.47212131  0.01754432
```

周数据，一周采一个点，采了 20 个点

30.4 时序检验

参数的计算公式，实现的 R 代码

© 基础

- Applies linear filtering to a univariate time series or to each series separately of a multivariate time series. 过滤

一元时间序列的线性过滤，或者对多元时间序列的单个序列分别做线性过滤

$$y[i] = x[i] + f[1] * y[i-1] + \ldots + f[p] * y[i-p]$$

$$y[i] = f[1] * x[i+o] + \ldots + f[p] * x[i+o-(p-1)]$$

其中 o 代表 offset

介绍 FTT 算法细节

不同的方法对时间序列平滑的影响 FTT 快速傅里叶变换算法

```
usage(stats::filter)
```

```
## filter(x, filter, method = c("convolution", "recursive"), sides = 2L,
##     circular = FALSE, init = NULL)
```

- `filter()` 时间序列线性过滤
- `fft()` 快速离散傅里叶变换

30.5 指数平滑

30.6 Holt-Winters

可加 Holt-Winters [Winters, 1960, Holt, 2004] 预测函数，周期长度为 p

$$\hat{Y}[t+h] = a[t] + h * b[t] + s[t-p+1+(h-1) \mod p$$

其中 $a[t], b[t], s[t]$ 由以下决定

$$a[t] = \alpha(Y[t] - s[t-p]) + (1-\alpha)(a[t-1] + b[t-1]) \tag{30.1}$$

$$b[t] = \beta(a[t] - a[t-1]) + (1-\beta)b[t-1] \tag{30.2}$$

$$s[t] = \gamma(Y[t] - a[t]) + (1-\gamma)s[t-p] \tag{30.3}$$

可乘 Holt-Winters

$$\hat{Y}[t+h] = (a[t] + h * b[t]) * s[t-p+1+(h-1) \mod p]$$

其中 $a[t], b[t], s[t]$ 由如下决定

© 王沛勇

$$a[t] = \alpha(Y[t]/s[t-p]) + (1-\alpha)(a[t-1] + b[t-1]) \tag{30.4}$$

$$b[t] = \beta(a[t] - a[t-1]) + (1-\beta)b[t-1] \tag{30.5}$$

$$s[t] = \gamma(Y[t]/a[t]) + (1-\gamma)s[t-p] \tag{30.6}$$

`HoltWinters()` 用 Shiny App / 动画的形式展示 α, β, γ 三个参数对模型预测的影响，参数的确定通过最小化预测均方误差

```
## Seasonal Holt-Winters
(m <- HoltWinters(co2))
plot(m)
plot(fitted(m))

p <- predict(m, 50, prediction.interval = TRUE)
plot(m, p)

(m <- HoltWinters(AirPassengers, seasonal = "mult"))
plot(m)

## 指数平滑 Exponential Smoothing
m2 <- HoltWinters(x, gamma = FALSE, beta = FALSE)
lines(fitted(m2)[,1], col = 3)
```

30.7 1749-2013 年太阳黑子数据

再从官网拿到最近的数据

```
plot(sunspot.month, xlab = "Year", ylab = "Monthly sunspot numbers",
     main = "Monthly mean relative sunspot numbers from 1749 to 2013")

autoplot(sunspot.month,
  main = "Monthly mean relative sunspot numbers from 1749 to 2013",
  xlab = "Year", ylab = "Monthly sunspot numbers"
)
```

```
autoplot(sunspots)
```

© 黄湘石

图 30.2: 时序图：太阳黑子月均数量

© 黄湘云

```
autoplot(sunspot.year, xlab = "Year", ylab = "Yearly Sunspot Data, 1700-1988") +
  theme_minimal()
```

```
library(dygraphs)
hw <- HoltWinters(sunspot.month)
predicted <- predict(hw, n.ahead = 72, prediction.interval = TRUE)

dygraph(predicted, main = "Predicted sunspot numbers") %>%
  dyAxis("x", drawGrid = FALSE) %>%
  dySeries(c("lwr", "fit", "upr"), label = "sunspot") %>%
  dyOptions(colors = hcl.colors(3))
```

```
par(family = "Noto Serif CJK SC")
plot(sunspot.month, col = "black")
lines(sunspots, col = "red")
legend("topright", legend = c("1749 至今", "1749-1983"), col = c("black", "red"), lty = 1)
```

30.8 1991-1998 年欧洲主要股票市场日闭市价格指数

```
matplot(time(EuStockMarkets), EuStockMarkets,
  main = "",
  xlab = "Date", ylab = "closing prices",
  pch = 17, type = "l", col = 1:4
```

© 黄湘石

图 30.3: 太阳黑子数量年平均时序图

图 30.4: 月均太阳黑子数

© 王沛雄

```
)
legend("topleft", colnames(EuStockMarkets), pch = 17, lty = 1, col = 1:4)
```

图 30.5: 1991-1998 年间欧洲主要股票市场日闭市价格指数图德国 DAX (Ibis), Switzerland SMI, 法国 CAC 和英国 FTSE

```
# 考虑收集加入最新的数据 1991~1998年的数据
plot(EuStockMarkets, plot.type = "single", col = hcl.colors(4))
legend("topleft", colnames(EuStockMarkets),
  col = hcl.colors(4), text.col = hcl.colors(4), lty = 1,
  box.col = NA, inset = 0.05
)
```

© 黄湘云

30.9 自回归模型

`ar()`

30.10 移动平均模型

`arima()`

30.11 自回归移动平均模型

`arima()` ARIMA

30.12 自回归条件异方差模型

自回归条件异方差模型 ARCH

© 王建湘

30.13 广义自回归条件异方差模型

广义自回归条件异方差模型（Generalized Autoregressive Conditional Heteroskedasticity，简称 GARCH）

30.14 其它特征的时间序列

```
plot(JohnsonJohnson)
plot(AirPassengers)
plot(nottem)
plot(lynx)
```

(a) 1960-1980 年强生公司每股季度收益

(b) 1949-1960 年月均航班乘客数量

(c) 1920-1939 年诺丁汉月均气温

(d) 1821-1934 年加拿大山猫陷阱数量

图 30.6: 时间序列：非平稳、周期性、非线性

30.15 港股走势

美团、阿里巴巴在香港上市

© 王惠湘


```
# 美团
meituan <- quantmod::getSymbols("3690.HK", auto.assign = FALSE, src = "yahoo", from = '2019-06-30')
# 阿里
ali <- quantmod::getSymbols("9988.HK", auto.assign = FALSE, src = "yahoo", from = '2019-06-30')
# 京东
sw <- quantmod::getSymbols("9618.HK", auto.assign = FALSE, src = "yahoo", from = '2019-06-30')
# 腾讯
tx <- quantmod::getSymbols("0700.HK", auto.assign = FALSE, src = "yahoo", from = '2019-06-30')
```

```
# 如何共 x 轴，右对齐
plot(as.ts(meituan[, "3690.HK.Close"]), col = "orange", ylab = "股价")
lines(as.ts(ali[, "9988.HK.Close"]), col = "springgreen4")
lines(as.ts(sw[, "9618.HK.Close"]), col = "purple4")
lines(as.ts(tx[, "0700.HK.Close"]), col = "lightsteelblue4")
legend("topright", col = c("Orange", "springgreen4", "purple4", "lightsteelblue4"),
       lty = 1, legend = c("美团", "阿里", "京东", "腾讯") )
```

30.16 美股走势

拼多多、京东、阿里巴巴、51Talk 在美股上市

```
# 拼多多
pdd <- quantmod::getSymbols("PDD", auto.assign = FALSE, src = "yahoo")
# 京东
jd <- quantmod::getSymbols("JD", auto.assign = FALSE, src = "yahoo")
# 阿里巴巴
baba <- quantmod::getSymbols("BABA", auto.assign = FALSE, src = "yahoo")
# 51Talk
coe <- quantmod::getSymbols("COE", auto.assign = FALSE, src = "yahoo", from = '2016-06-30')
```

30.17 51Talk 股价走势

Joshua M. Ulrich 开发维护的 quantmod 包可以下载国内外股票市场的数据

51talk 于 2016 年 6 月 10 日在美国纽交所上市，股票代码 COE，2020 年 1 月 22 日，武汉封城，受新冠肺炎病毒影响，政府停课不停学的号召，线下教育纷纷转线上，线上教育的春天来临，股价开始回升到发行价的水平，在公司将资源转变为能力后，预期公司股价继续翻倍，回到理性的水平。

```
coe <- quantmod::getSymbols("COE", auto.assign = FALSE, src = "yahoo", from = '2016-06-30')
```

读者可以从雅虎财经获取数据源 https://finance.yahoo.com/

```
plot(coe[, "COE.Close"],
  subset = "2016-06-30/2021-06-30",
```

©黄湘云

```
  col = "Orange", main = "COE Stock Close Price"
)
```

图 30.7: 51Talk 公司上市以来的股价走势

COE 股价变化趋势见下图，包含开盘价 Open、最低价 Low、最高价 High、闭市价 Close 和调整价 Adjust 和交易额 Volume

```
autoplot(coe)
```

30.18 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
```

©黄湘云

图 30.8: CEO 股价变化趋势

©黄湘云

```
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dygraphs_1.1.1.6 ggfortify_0.4.14 ggplot2_3.3.5    formatR_1.11
##
## loaded via a namespace (and not attached):
##  [1] zoo_1.8-9         tidyselect_1.1.2  xfun_0.30         purrr_0.3.4
##  [5] lattice_0.20-45   colorspace_2.0-3  vctrs_0.4.0       generics_0.1.2
##  [9] htmltools_0.5.2   yaml_2.3.5        utf8_1.2.2        rlang_1.0.2
## [13] pillar_1.7.0      glue_1.6.2        withr_2.5.0       DBI_1.1.2
## [17] TTR_0.24.3        lifecycle_1.0.1   quantmod_0.4.18   stringr_1.4.0
## [21] munsell_0.5.0     gtable_0.3.0      htmlwidgets_1.5.4 evaluate_0.15
## [25] labeling_0.4.2    knitr_1.38        fastmap_1.1.0     curl_4.3.2
## [29] fansi_1.0.3       xts_0.12.1        scales_1.1.1      showtext_0.9-5
## [33] sysfonts_0.8.8    farver_2.1.0      gridExtra_2.3     digest_0.6.29
## [37] stringi_1.7.6     bookdown_0.25     dplyr_1.0.8       showtextdb_3.0
## [41] grid_4.1.3        cli_3.2.0         tools_4.1.3       magrittr_2.0.3
## [45] tibble_3.1.6      crayon_1.5.1      tidyr_1.2.0       pkgconfig_2.0.3
## [49] ellipsis_0.3.2    assertthat_0.2.1  rmarkdown_2.13    R6_2.5.1
## [53] compiler_4.1.3
```

© 黄湘云

第八部分

时空数据

© 基辛湖石

介绍

数据建模

© 黄湘云

第三十一章　空间数据分析

Robert Hijmans 开发的 terra 用以替代 raster，提供栅格数据和向量数据处理，基于回归和机器学习方法的空间差值和预测，能够处理相当大的数据集，包括卫星遥感数据，新的 R 包更加简洁、速度更快、功能更强。Edzer Pebesma 创建的 r-spatial 开源组织提供了一系列非常流行的空间分析相关的 R 包，如 sp、sf、stars、mapedit 和 mapview。Edzer Pebesma 长期致力于地理信息和空间统计的软件开发，可以说目前已打造了一个生态。

Timothée Giraud 创建的 riatelab 组织开发系列 R 包工具，可以绘制各种类型和风格的地图，专题地图工具已经从 cartography 过渡到 mapsf，它更加友好、轻量和稳健。类似的 R 包还有 choroplethr，只是上次更新在 2015 年。

空间数据可视化常常离不开基础地图数据，不同的 R 包依赖的地图服务有所不同，比如 RgoogleMaps、ggmap 和 googleway 主要依赖谷歌的地图数据。而 mapdeck 基于 deck.gl 和 Mapbox 支持移动和网页应用，GPU 渲染等。leaflet 则基于开源的Leaflet库提供交互式空间数据可视化的能力。

芝加哥大学空间数据科学中心 开发的 R 包 rgeoda 基于开源的 C++ 库GeoDa，提供一系列空间数据分析能力，包括探索性空间数据分析、空间聚类检测和聚类分析。

Edzer Pebesma 和 Roger Bivand 合著的 Spatial Data Science with applications in R，Christopher K. Wikle, Andrew Zammit-Mangion 和 Noel Cressie 合著的 Spatio-Temporal Statistics with R。推荐学习 Edzer Pebesma 在几届国际 R 语言大会上的材料，2021 年的R Spatial，2020 年的Analyzing and visualising spatial and spatiotemporal data cubes Part I，2019 年的Spatial workshop part I 和 Spatial workshop part II，2017 年的Spatial Data in R: New Directions 2016 年的Handling and Analyzing Spatial, Spatiotemporal and Movement Data。

```
library(sf)
# North Carolina 城镇
nc <- st_read(system.file("shape/nc.shp", package = "sf"))
library(mapview)
mapview(nc, zcol = c("SID74", "SID79"), alpha.regions = 1.0, legend = TRUE)
```

rgdal 包可以实现坐标变换

```
# https://github.com/geodacenter/rgeoda/
library(rgeoda)
library(sf)

guerry <- st_read(system.file("extdata", "Guerry.shp", package = "rgeoda"))

crm_prp <- guerry["Crm_prp"]
```

内部资料 ©

```r
queen_w <- queen_weights(guerry)

lisa <- local_moran(queen_w, crm_prp)

lisa_colors <- lisa_colors(lisa)
lisa_labels <- lisa_labels(lisa)
lisa_clusters <- lisa_clusters(lisa)

plot(st_geometry(guerry),
     col = sapply(lisa_clusters, function(x) {
       lisa_colors[[x + 1]]
     }),
     border = "#333333", lwd = 0.2
)
title(main = "Local Moran Map of Crm_prs")
legend("bottomleft",
  legend = lisa_labels,
  fill = lisa_colors,
  border = "#eeeeee"
)
```

© 王建国

第三十二章　空间数据可视化

Robert J. Hijmans [1] 开发了 raster 包用于网格空间数据的读、写、操作、分析和建模，同时维护了空间数据分析的网站 https://www.rspatial.org。Edzer Pebesma [2] 和 Roger Bivand 等创建了 sp 包定义了空间数据类型和方法，提供了大量的空间数据操作方法，同时维护了空间数据对象 sp 的绘图网站 https://edzer.github.io/sp/，他们也一起合作写了新书 Spatial Data Science，提供了在线 网页版 书籍及其 源代码。Edzer Pebesma 后来开发了 sf 包重新定义了空间数据对象和操作方法，并维护了空间数据分析、建模和可视化网站 https://www.r-spatial.org/

课程案例学习

1. 2018-Introduction to Geospatial Raster and Vector Data with R 空间数据分析课程
2. Peter Ellis 新西兰大选和普查数据 More cartograms of New Zealand census data: district and city level
3. 2017-Mapping oil production by country in R 石油产量在全球的分布
4. 2017-How to highlight countries on a map 高亮地图上的国家
5. 2017-Mapping With Sf: Part 3
6. Data Visualization Shiny Apps 数据可视化核密度估计 In this app I identify crime hotspots using a bivariate density estimation strategy
7. Association of Statisticians of American Religious Bodies (ASARB) viridis USA map
8. 出租车行车轨迹数据
9. Geospatial processing with Clickhouse-CARTO Blog

32.1　空间数据

空间数据存储在数据库中，比如 PostGIS，它是对象关系数据库 PostgreSQL 在空间数据库方面的扩展。

32.1.1　raster

```
library(sp)
library(raster)
```

raster 包定义了获取和操作空间 raster 类型数据集的类和方法，rasterVis 补充加强了 raster 包在数据可视化和交互方面的功能。可视化是基于 lattice 的

raster 包的开发已经被作者 Robert J. Hijmans 迁移到 Github 上啦，官方文档 https://www.rspatial.org/

[1] Department of Environmental Science and Policy at the University of California, Davis. Ecology, Geography, and Agriculture

[2] Institute for Geoinformatics of the University of Münster.

©黄湘云

表 32.1: plot 方法

plot,ANY,ANY-method	plot,Extent,missing-method
plot,Raster,ANY-method	plot,Raster,Raster-method
plot,SpatExtent,missing-method	plot,Spatial,missing-method
plot,SpatialGrid,missing-method	plot,SpatialGridDataFrame,missing-method
plot,SpatialLines,missing-method	plot,SpatialMultiPoints,missing-method
plot,SpatialPixels,missing-method	plot,SpatialPixelsDataFrame,missing-method
plot,SpatialPoints,missing-method	plot,SpatialPolygons,missing-method
plot,SpatRaster,character-method	plot,SpatRaster,missing-method
plot,SpatRaster,numeric-method	plot,SpatRaster,SpatRaster-method
plot,SpatVector,character-method	plot,SpatVector,missing-method
plot,SpatVector,numeric-method	plot,SpatVectorProxy,missing-method
plot.acf	plot.data.frame
plot.decomposed.ts	plot.default
plot.dendrogram	plot.density
plot.ecdf	plot.factor
plot.formula	plot.function
plot.hclust	plot.histogram
plot.HoltWinters	plot.isoreg
plot.lm	plot.medpolish
plot.mlm	plot.ppr
plot.prcomp	plot.princomp
plot.profile.nls	plot.raster
plot.shingle	plot.spec
plot.stepfun	plot.stl
plot.table	plot.trellis
plot.ts	plot.tskernel
plot.TukeyHSD	plot,ANY,ANY-method

`methods(plot)` 星号 * 标记的是 S3 方法。

```
## Warning in matrix(data = methods(plot), ncol = 2, byrow = T): data length [53]
## is not a sub-multiple or multiple of the number of rows [27]
```

查看函数的定义

```
getAnywhere(plot.raster)
```

```
## A single object matching 'plot.raster' was found
## It was found in the following places
##   registered S3 method for plot from namespace graphics
##   namespace:graphics
## with value
##
## function (x, y, xlim = c(0, ncol(x)), ylim = c(0, nrow(x)), xaxs = "i",
##     yaxs = "i", asp = 1, add = FALSE, ...)
```

© 王建平


```
## {
##     if (!add) {
##         plot.new()
##         plot.window(xlim = xlim, ylim = ylim, asp = asp, xaxs = xaxs,
##             yaxs = yaxs)
##     }
##     rasterImage(x, 0, 0, ncol(x), nrow(x), ...)
## }
## <bytecode: 0x55a5d9c04490>
## <environment: namespace:graphics>
```

rasterImage 函数来绘制图像，如果想知道 `rasterImage` 的内容可以继续看 `getAnywhere(rasterImage)`

```
getAnywhere(rasterImage)
```

```
## A single object matching 'rasterImage' was found
## It was found in the following places
##   package:graphics
##   namespace:graphics
## with value
##
## function (image, xleft, ybottom, xright, ytop, angle = 0, interpolate = TRUE,
##     ...)
## {
##   .External.graphics(C_raster, if (inherits(image, "nativeRaster")) image else as.raster(image),
##         as.double(xleft), as.double(ybottom), as.double(xright),
##         as.double(ytop), as.double(angle), as.logical(interpolate),
##         ...)
##     invisible()
## }
## <bytecode: 0x55a5d9dcb458>
## <environment: namespace:graphics>
```

通过查看函数的帮助 `?rasterImage`，我们需要重点关注一下参数 *image* 传递的 raster 对象。

```
plot(c(100, 250), c(300, 450), type = "n", xlab = "", ylab = "")
x <- rep(0, 15)
x[seq(from = 2, to = 14, by = 2)] <- 1
image <- as.raster(matrix(x, ncol = 5, nrow = 3))
rasterImage(image, 100, 300, 150, 350, interpolate = FALSE)
# 插值平滑
rasterImage(image, 100, 400, 150, 450)
# 缩小比例
rasterImage(image, 200, 300, 200 + xinch(.5), 300 + yinch(.3),
  interpolate = FALSE
)
# 旋转图像
rasterImage(image, 200, 400, 250, 450,
```

© 黄湘云


```
  angle = 15, interpolate = FALSE
)
```

图 32.1: raster 图像

32.2 可视化

32.2.1 斐济地震带分布

相比于 **plotly**，**echarts4r** 更加轻量，这得益于 JavaScript 库 Apache ECharts。前者 MIT 协议，后者采用 Apache-2.0 协议，都可以商用。Apache ECharts 是 Apache 旗下顶级开源项目，由百度前端技术团队贡献，中文文档也比较全，学习起来门槛会低一些。

```
library(echarts4r)
quakes |>
  e_charts(x = long) |>
  e_geo(
    roam = TRUE,
    boundingCoords = list(
      c(185, -10),
      c(165, -40)
    )
  ) |>
  e_scatter(
    serie = lat,
    size = mag, # 点的大小映射到震级
    # legend = F, # 是否移除图例
    name = "斐济地震带",
    coord_system = "geo"
  ) |>
  e_visual_map(
    serie = mag, scale = e_scale,
```

© 黄湘云


```
    inRange = list(color = terrain.colors(10))
  ) |>
  e_tooltip()
```

32.3 美国各个城镇的失业率分布

以 2009 年美国各个城镇的失业率数据为例，数据来自 **maps** 包的 unemp 数据集，它有三个变量，fips 城镇代码[3]，pop 人口，unemp 失业率。失业率本是连续的数据，将其分级划分区间，每个失业率区间用不同颜色表示。

32.3.1 maps

maps 包提供城镇地图数据，数据集 county.fips 各个城镇的名称 polyname 及行政代码 fips，和 unemp 数据集关联可以知道各个城镇的失业率，再与城镇地图数据关联，就可以将数据绘制在地图上。county.fips 没有夏威夷、阿拉斯加、波多黎各的地图数据，导致 unemp 数据集里阿拉斯加、夏威夷、波多黎各和部分弗吉尼亚的小城市无法映射到地图上。

```
# 代码调整自帮助文档 ?map
library(maps)
library(mapproj)
# 失业率数据
data(unemp)
# 行政编码
data(county.fips)
# 准备调色板
# colors <- c("#F1EEF6", "#D4B9DA", "#C994C7", "#DF65B0", "#DD1C77", "#980043")
colors <- viridisLite::plasma(9)
# 失业率划分区间
unemp$colorBuckets <- as.numeric(cut(unemp$unemp, c(seq(0, 10, by = 2), 15, 20, 25, 100)))
# 准备图例文本
leg.txt <- c("<2%", "2-4%", "4-6%", "6-8%", "8-10%", "10-15%", "15-20%", "20-25%", ">25%")
# 根据区域单元的名称匹配地图数据上每个区域的 FIPS
cnty.fips <- county.fips$fips[match(
  map("county", plot = FALSE)$names,
  county.fips$polyname
)]
# 根据 FIPS 给地图上每个区域的失业率匹配颜色
colorsmatched <- unemp$colorBuckets[match(cnty.fips, unemp$fips)]
par(mar = c(1.5, 0, 2, 0))
# 绘制区县地图
map("county",
  col = colors[colorsmatched], fill = TRUE, resolution = 0,
```

[3]可类比我国行政区划代码，自 1980 年以来，每年都会发布一次，涉及一些市、区、县、乡、镇、街道等的变更。

© 黄湘云


```
  lty = 0, projection = "polyconic", mar = c(0.5, 0.5, 2, 0.5)
)
# 添加州边界线
map("state",
  col = "white", fill = FALSE, add = TRUE,
  lty = 1, lwd = 0.2, projection = "polyconic"
)
# 添加图标题
title("2009 年美国各个城镇的失业率")
mtext(text = "数据源：美国人口调查局", side = 1, adj = 0)
# 添加图例
legend("top", leg.txt, horiz = TRUE, fill = colors, cex = 0.85)
```

图 32.2: 2009 年美国各个城镇的失业率分布

函数 match() 返回一个向量，向量的长度与 x 一致，向量的元素是整型的，表示 x 中的元素出现在 table 中的位置。

```
match(x = c("A", "B"), table = c("A"))
```

```
## [1]  1 NA
```

© 邱怡轩


```
match(x = c("A", "B"), table = c("C", "A"))
```

```
## [1]  2 NA
```

```
match(x = c("A", "B"), table = c("C", "A", "D"))
```

```
## [1]  2 NA
```

32.3.2 latticeExtra

```
# 地图数据
us_county <- map("county", plot = FALSE, fill = TRUE, projection = "polyconic")
# 失业率数据和行政区域名称关联
unemp_df <- merge(unemp, county.fips, by = "fips")
# 绘图
latticeExtra::mapplot(polyname ~ unemp,
  data = unemp_df,
  map = us_county,
  colramp = viridisLite::plasma,
  border = NA,
  # cuts = 10, # 等距分桶的数，和参数 breaks 二选一
  breaks = c(seq(0, 10, by = 2), 15, 20, 25, 30, 35),
  subset = polyname %in% us_county$names,
  scales = list(draw = F),
  xlab = "",
  par.settings = list(
    # 副标题放在左下角
    par.sub.text = list(
      font = 2,
      just = "left",
      x = grid::unit(5, "mm"),
      y = grid::unit(5, "mm")
    ),
    # 轴线设置白色以隐藏
    axis.line = list(col = "white")
  ),
  sub = "数据源：美国人口调查局",
  main = "2009 年美国各个城镇的失业率"
)
```

32.3.3 ggplot2

usmapdata 包提供美国国家、州和城镇边界地图数据，下面以此数据为基础，借助 **ggplot2** 包 [?] 绘制失业率专题地图，未收集到失业率数据的城镇填充灰色，图中中文采用 **showtext** 包 [Qiu, 2015] 处理，如图 32.4 所示。

© 黄湘云

图 32.3: 2009 年美国各个城镇的失业率分布

```
# 失业率数据和行政编码数据结合
# unemp_df <- merge(unemp, county.fips, by = "fips")
# 从 usmapdata 包获取地图数据
county_df <- usmapdata::us_map("counties")
# 行政编码是一串数字组成的字符串
county_df$fips <- as.numeric(county_df$fips)
# 地图数据和失业率数据结合
choropleth <- merge(county_df, unemp_df, by = "fips", all.x = TRUE)
# 还原地图数据的顺序
choropleth <- choropleth[order(choropleth$order), ]
# 失业率分级
choropleth$rate_d <- cut(choropleth$unemp, breaks = c(seq(0, 10, by = 2), 15, 20, 25, 30, 35))
# 准备州边界线数据
state_df <- usmapdata::us_map("states")
# 绘图
library(ggplot2)
ggplot(choropleth, aes(x, y, group = group)) +
  geom_polygon(aes(fill = rate_d), colour = alpha("gray95", 1/4), size = 0.2) +
  geom_polygon(data = state_df, colour = "gray80", fill = NA, size = 0.3) +
  scale_fill_viridis_d(option = "plasma", na.value = "gray80") +
  labs(
    fill = "失业率(%)", title = "2009 年美国各个城镇的失业率",
    caption = "数据源：美国人口调查局"
```

© 黄湘云


```
  ) +
  theme_void() +
  theme(plot.title = element_text(hjust = 0.5))
```

图 32.4: 2009 年美国各个城镇的失业率分布

32.3.4 sf

相比于前任 **sp** 包，**sf** 包将是新一代空间数据操作、分析和可视化的基石，引入 GDAL、PROJ 和 GEOS 三大基础框架，和更庞大的空间数据社区接轨，不局限于 R 语言社区的一亩三分地。**sf** 包支持 Base R 绘图，以此绘制失业率专题地图，如图32.5所示，可见效果丝毫不逊于 **lattice** 和 **ggplot2** 版本，而且在兼容性、代码量、稳定性和性能等方面有明显优势。

```
library(sf)
# 准备地图数据
us_states_shifted <- readRDS(file = "data/us_states_shifted.rds")
us_county_shifted <- readRDS(file = "data/us_county_shifted.rds")
# 准备用于合并操作的主键
us_county_shifted <- within(us_county_shifted, {
  polyname <- tolower(paste(STATE_NAME, NAME, sep = ","))
})
# 将失业率数据和地图数据合并
```

© 基础统计

```
us_county_data <- merge(x = us_county_shifted, y = unemp_df, by = "polyname", all.x = T)

plot(us_county_data["unemp"],
  pal = viridisLite::plasma,
  breaks = c(seq(0, 10, by = 2), 15, 20, 25, 30, 35),
  border = alpha("gray95", 1/8), key.pos = 4, reset = FALSE,
  main = "2009 年美国各个城镇的失业率"
)
# 添加州边界
plot(st_geometry(us_states_shifted), col = NA, border = "gray", add = T)
mtext(text = "数据源：美国人口调查局", side = 1, adj = 0)
```

图 32.5: 2009 年美国各个城镇的失业率分布

```
# 绘制失业率地图
par(mar = c(1, 0, 2, 0))
plot(st_geometry(us_county_data),
  col = "gray80",
  border = alpha("gray95", 1 / 4),
  main = "2009 年美国各个城镇的失业率", reset = FALSE
)
plot(us_county_data["unemp"],
  pal = viridisLite::plasma,
  breaks = c(seq(0, 10, by = 2), 15, 20, 35),
  border = alpha("gray95", 1 / 8), key.pos = 4, add = TRUE
)
```

© 王建湘


```
# 添加州边界
plot(st_geometry(us_states_shifted), col = NA, border = "gray90", add = T)
mtext(text = "数据源：美国人口调查局", side = 1, adj = 0)
# 添加图例
legend("top",
  legend = c(
    "<2%", "2-4%", "4-6%", "6-8%", "8-10%",
    "10-15%", "15-20%", ">20%", "NA"
  ),
  horiz = T, border = NA, box.col = "gray",
  fill = c(viridisLite::plasma(8), "gray80"), cex = 0.75
)
```

图 32.6: 2009 年美国各个城镇的失业率分布

32.3.5 mapsf

© 王沛勇

第三十三章　案例研究

```
library(magrittr)
library(ggplot2)
library(gganimate)

library(formattable)
library(packagemetrics)
```

提升回归模型的 10 个提示 10 quick tips to improve your regression modeling

easystats 包含 insight [Lüdecke et al., 2019] 和 bayestestR [Makowski et al., 2019] 等共 9 个 R 包，tidymodels 也包含差不多量的 R 包。

rms Regression Modeling Strategies

gtsummary modelsummary 整理模型输出，提供丰富的格式输出，如 PDF, Text/Markdown, LaTeX, MS Word, RTF, JPG, and PNG.

```
library(gtsummary)
library(modelsummary)
```

R for Data Science Online Learning Community 在线学习社区以 tidytuesday 闻名遐迩。

```
#padding: 25
#fontsize: 18
#stroke: #26A63A
#linewidth: 2

[Import] -> [Understand]

[Understand |
  [Wrangle] -> [Visualize]
  [Visualize] -> [Model]
  [Model] -> [Wrangle]
]

[Understand] -> [Communicate]
```

统计建模：两种文化 [Breiman, 2001]

这些案例来自 Kaggle、Tudesday 或者自己找的数据集，而不是论文里，或者 R 包里的小数据

© 黄湘云

表 33.1: 不同生长环境下植物的干重

	1	2	3	4	5	6	7	8	9	10
ctrl	4.17	5.58	5.18	6.11	4.50	4.61	5.17	4.53	5.33	5.14
trt1	4.81	4.17	4.41	3.59	5.87	3.83	6.03	4.89	4.32	4.69
trt2	6.31	5.12	5.54	5.50	5.37	5.29	4.92	6.15	5.80	5.26

集，应该更加真实，贴近实际问题，考虑更多细节

33.1 统计学家生平

世纪统计学家 100 位统计学家，寿命的影响因素，关联分析，图展示数据本身的

注明每位统计学家所在的年代经历的重大事件，如欧洲中世纪霍乱，第二次世界大战，文化大革命，用图形来讲故事，展现数据可视化的魅力，参考文献 [Johnson and Kotz, 1997]

33.2 R 语言发展历史

R 语言发展历史和现状，用图来表达

33.3 不同实验条件下植物生长情况

PlantGrowth 数据集收集自 Annette J. Dobson 所著书籍《An Introduction to Statistical Modelling》[Dobson, 1983] 第 2 章第 2 节的案例 — 研究植物在两种不同试验条件下的生长情况，植物通过光合作用吸收土壤的养分和空气中的二氧化碳，完成积累，故以植物的干重来刻画植物的生长情况，首先将几乎相同的种子随机地分配到实验组和对照组，基于完全随机实验设计（completely randomized experimental design），经过预定的时间后，将植物收割，干燥并称重，结果如表 33.1 所示

```
# do.call("cbind", lapply(split(PlantGrowth, f = PlantGrowth$group), subset, select = "weight"))
## 或者
library(magrittr)
split(PlantGrowth, f = PlantGrowth$group) %>% # 分组
  lapply(., subset, select = "weight") %>% # 计算
  Reduce("cbind", .) %>% # 合并
  setNames(., levels(PlantGrowth$group)) %>% # 重命名 `colnames<-`(., levels(PlantGrowth$group))
  t %>%
  knitr::kable(.,
    caption = "不同生长环境下植物的干重", row.names = TRUE,
    align = "c"
  )
```

设立对照组（控制组）ctrl 和实验组 trt1 和 trt2，比较不同的处理方式对植物干重的影响

©李清华

```
summary(PlantGrowth)
```

```
##      weight       group
##  Min.   :3.590   ctrl:10
##  1st Qu.:4.550   trt1:10
##  Median :5.155   trt2:10
##  Mean   :5.073
##  3rd Qu.:5.530
##  Max.   :6.310
```

每个组都有 10 颗植物，生长情况如图33.1所示

图 33.1: 植物干重

实验条件 trt1 和 trt2 对植物生长状况有显著的影响，为了量化这种影响，建立线性回归模型

```
fit_sublm <- lm(weight ~ group,
  data = PlantGrowth,
  subset = group %in% c("ctrl", "trt1")
)
anova(fit_sublm)
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df Sum Sq Mean Sq F value Pr(>F)
## group      1 0.6882 0.68820  1.4191  0.249
## Residuals 18 8.7292 0.48496
```

```
summary(fit_sublm)
```

```
##
## Call:
## lm(formula = weight ~ group, data = PlantGrowth, subset = group %in%
##     c("ctrl", "trt1"))
```

© 黄湘云


```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0710 -0.4938  0.0685  0.2462  1.3690
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0320     0.2202  22.850 9.55e-15 ***
## grouptrt1    -0.3710     0.3114  -1.191    0.249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6964 on 18 degrees of freedom
## Multiple R-squared:  0.07308,    Adjusted R-squared:  0.02158
## F-statistic: 1.419 on 1 and 18 DF,  p-value: 0.249
```

下面再通过检验的方式比较实验组和对照组相比，是否有显著作用

```r
# 控制组和实验组1比较
t.test(weight ~ group, data = PlantGrowth, subset = group %in% c("ctrl", "trt1"))
```

```
##
##  Welch Two Sample t-test
##
## data:  weight by group
## t = 1.1913, df = 16.524, p-value = 0.2504
## alternative hypothesis: true difference in means between group ctrl and group trt1 is not equal to 0
## 95 percent confidence interval:
##  -0.2875162  1.0295162
## sample estimates:
## mean in group ctrl mean in group trt1
##              5.032              4.661
```

```r
# 控制组和实验组2比较
t.test(weight ~ group, data = PlantGrowth, subset = group %in% c("ctrl", "trt2"))
```

```
##
##  Welch Two Sample t-test
##
## data:  weight by group
## t = -2.134, df = 16.786, p-value = 0.0479
## alternative hypothesis: true difference in means between group ctrl and group trt2 is not equal to 0
## 95 percent confidence interval:
##  -0.98287213 -0.00512787
## sample estimates:
## mean in group ctrl mean in group trt2
##              5.032              5.526
```

统计学基础 ©

检验结果表明，实验条件 trt2 会对植物生长产生显著效果，而实验条件 trt1 不会。在假定同方差的情况下，建立线性回归模型，同时考虑实验条件 trt1 和 trt2

```
# 模型拟合
fit_lm <- lm(weight ~ group, data = PlantGrowth)

## 模型输出
summary(fit_lm)
```

```
##
## Call:
## lm(formula = weight ~ group, data = PlantGrowth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0710 -0.4180 -0.0060  0.2627  1.3690
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0320     0.1971  25.527   <2e-16 ***
## grouptrt1    -0.3710     0.2788  -1.331   0.1944
## grouptrt2     0.4940     0.2788   1.772   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6234 on 27 degrees of freedom
## Multiple R-squared:  0.2641, Adjusted R-squared:  0.2096
## F-statistic: 4.846 on 2 and 27 DF,  p-value: 0.01591
```

```
## 方差分析
anova(fit_lm)
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## group      2  3.7663  1.8832  4.8461 0.01591 *
## Residuals 27 10.4921  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## 参数估计
coef(summary(fit_lm))
```

```
##             Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)    5.032  0.1971284 25.526514 1.936575e-20
## grouptrt1     -0.371  0.2787816 -1.330791 1.943879e-01
## grouptrt2      0.494  0.2787816  1.771996 8.768168e-02
```

©黄湘云

表 33.2: 线性回归的输出

	估计值	标准差	t 统计量	P 值
α	5.032	0.1971	25.5265	0.0000
β_1	-0.371	0.2788	-1.3308	0.1944
β_2	0.494	0.2788	1.7720	0.0877

模型输出整理成表 33.2 所示

还可以将模型转化为数学公式

```
# 理论模型
equatiomatic::extract_eq(fit_lm)
```

$$\text{weight} = \alpha + \beta_1(\text{group}_{\text{trt1}}) + \beta_2(\text{group}_{\text{trt2}}) + \epsilon \tag{33.1}$$

```
# 拟合模型
equatiomatic::extract_eq(fit_lm, use_coefs = TRUE)
```

$$\widehat{\text{weight}} = 5.03 - 0.37(\text{group}_{\text{trt1}}) + 0.49(\text{group}_{\text{trt2}}) \tag{33.2}$$

进一步地，我们在线性模型的基础上考虑每个实验组有不同的方差，先做方差齐性检验。

```
bartlett.test(weight ~ group, data = PlantGrowth)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  weight by group
## Bartlett's K-squared = 2.8786, df = 2, p-value = 0.2371
```

```
fligner.test(weight ~ group, data = PlantGrowth)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  weight by group
## Fligner-Killeen:med chi-squared = 2.3499, df = 2, p-value = 0.3088
```

检验的结果显示，可以认为三个组的方差没有显著差异，但我们还是考虑每个组有不同的方差，看看放开假设能获得多少提升，后续会发现，从对数似然的角度来看，实际提升量很小，只有 7.72%

上面同时比较多个总体的方差，会发现方差没有显著差异，那么接下来在假定方差齐性的条件下，比较均值的差异是否显著?

```
# 参数检验，假定异方差
oneway.test(weight ~ group, data = PlantGrowth, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
```

统计学习 ©

```
##
## data:  weight and group
## F = 5.181, num df = 2.000, denom df = 17.128, p-value = 0.01739
```

```r
# 参数检验，假定方差齐性
oneway.test(weight ~ group, data = PlantGrowth, var.equal = TRUE)
```

```
##
##  One-way analysis of means
##
## data:  weight and group
## F = 4.8461, num df = 2, denom df = 27, p-value = 0.01591
```

```r
# 非参数检验
kruskal.test(weight ~ group, data = PlantGrowth)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  weight by group
## Kruskal-Wallis chi-squared = 7.9882, df = 2, p-value = 0.01842
```

检验结果显示它们的均值是有显著差异的！

```r
# 固定效应模型
fit_gls <- nlme::gls(weight ~ 1,
  weights = nlme::varIdent(form = ~ 1 | group),
  data = PlantGrowth, method = "REML"
)
summary(fit_gls)
```

```
## Generalized least squares fit by REML
##   Model: weight ~ 1
##   Data: PlantGrowth
##        AIC      BIC    logLik
##   70.48628 75.95547 -31.24314
##
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | group
##  Parameter estimates:
##      ctrl      trt1      trt2
## 1.0000000 1.5825700 0.9230865
##
## Coefficients:
##                Value Std.Error  t-value p-value
## (Intercept) 5.199759 0.1162421 44.73214       0
##
```

© 基础统计


```
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -1.74647988 -0.91870713 -0.07591108  0.60676033  2.03987301
##
## Residual standard error: 0.5896195
## Degrees of freedom: 30 total; 29 residual
```

```
# 随机效应模型
fit_lme <- nlme::lme(weight ~ 1, random = ~ 1 | group, data = PlantGrowth)
summary(fit_lme)
```

```
## Linear mixed-effects model fit by REML
##   Data: PlantGrowth
##        AIC      BIC    logLik
##   67.44473 71.54662 -30.72237
##
## Random effects:
##  Formula: ~1 | group
##         (Intercept)  Residual
## StdDev:   0.3865976 0.6233746
##
## Fixed effects:  weight ~ 1
##             Value Std.Error DF  t-value p-value
## (Intercept) 5.073 0.2505443 27 20.24792       0
##
## Standardized Within-Group Residuals:
##          Min           Q1          Med           Q3          Max
## -1.854449795 -0.688750457  0.006389611  0.406096866  2.059729645
##
## Number of Observations: 30
## Number of Groups: 3
```

$\sigma_i^2 = Var(\epsilon_{ij}), i = 1, 2, 3$ 表示第 i 组的方差，

$$y_{ij} = \mu + \epsilon_{ij}, i = 1, 2, 3$$

其中 μ 是固定的未知参数，我们和之前假定同方差情形下的模型比较一下，现在异方差情况下模型提升的情况，从对数似然的角度来看

```
logLik(fit_lm)
## 'log Lik.' -26.80952 (df=4)
logLik(fit_lm, REML = TRUE)
## 'log Lik.' -29.00481 (df=4)
logLik(fit_gls)
## 'log Lik.' -31.24314 (df=4)
logLik(fit_lme)
## 'log Lik.' -30.72237 (df=3)
```

统计学习基础 ©

进一步地，我们考虑两水平模型，认为不同的实验组其均值和方差都不一样，检验三样本均值是否相等？

$\mu_1 = \mu_2 = \mu_3$ 检验，这里因为每组的样本量都一样，因此考虑 Turkey 的 T 法检验，检验均值是否有显著差别，实际上这里因为实验组数量只有 2 个，可以两两比对，如前所述。但是这里我们想扩展一下，考虑多组比较的问题。

和上面用 `gls` 拟合的模型是一致的。

$$y_{ij} = \mu_i + \epsilon_{ij}, \tag{33.3}$$

$$\mu_i = \mu_\theta + \xi_i. \quad i = 1, \ldots, 3; \quad j = 1, \ldots, 10. \tag{33.4}$$

其中 μ_i 是随机的未知变量，服从均值为 μ_θ 方差为 $Var(\xi_i) = \tau^2$ 的正态分布

我们用 **MASS** 包提供的 `glmmPQL()` 函数拟合该数据集

```
fit_lme_pql <- MASS::glmmPQL(weight ~ 1,
  random = ~ 1 | group, verbose = FALSE,
  family = gaussian(), data = PlantGrowth
)
summary(fit_lme_pql)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: PlantGrowth
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | group
##         (Intercept)  Residual
## StdDev:   0.2944234 0.6233746
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects:  weight ~ 1
##             Value Std.Error DF  t-value p-value
## (Intercept) 5.073 0.2080656 27 24.38174       0
##
## Standardized Within-Group Residuals:
##          Min           Q1          Med           Q3          Max
## -1.922640850 -0.734727623  0.004564386  0.405111223  1.991538416
##
## Number of Observations: 30
## Number of Groups: 3
```

我们再借助 **brms** 包从贝叶斯的角度来分析数据，并建模

©黄湘云


```r
# 贝叶斯模型
fit_brm <- brms::brm(weight ~ group, data = PlantGrowth)
# 参考 https://www.xiangyunhuang.com.cn/2019/05/normal-hierarchical-model/
library(Rcpp)
fit_lme_brm <- brms::brm(weight ~ 1 + (1 | group),
  data = PlantGrowth, family = gaussian(),
  refresh = 0, seed = 2019
)
summary(fit_lme_brm)
```

33.4 橘树生长情况

Orange 数据集包含三个变量，记录了加利福尼亚南部的一个小树林中的五棵橘树的生长情况，在 **datasets** 包里，数据集保存为 `c("nfnGroupedData", "nfGroupedData", "groupedData", "data.frame")` 类型的数据，同时具有着四个类的特点。

- Tree: 有序的指示变量，根据 5 棵橘树的最大直径划分，测量值很可能是根据林务员常用的“胸围周长”
- age: 橘树的树龄，自 1968 年 12 月 31 日起按天计算
- circumference: 橘树树干的周长，单位是毫米

查看部分数据的情况

```r
head(Orange)
```

```
## Grouped Data: circumference ~ age | Tree
##   Tree  age circumference
## 1    1  118            30
## 2    1  484            58
## 3    1  664            87
## 4    1 1004           115
## 5    1 1231           120
## 6    1 1372           142
```

查看变量的属性

```r
str(Orange)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':  35 obs. of  3 variables:
##  $ Tree         : Ord.factor w/ 5 levels "3"<"1"<"5"<"2"<..: 2 2 2 2 2 2 2 4 4 4 ...
##  $ age          : num  118 484 664 1004 1231 ...
##  $ circumference: num  30 58 87 115 120 142 145 33 69 111 ...
##  - attr(*, "formula")=Class 'formula'  language circumference ~ age | Tree
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Time since December 31, 1968"
##   ..$ y: chr "Trunk circumference"
```

© 黄湘云

表 33.3: 躯干周长（毫米）随时间（天）的变化

Tree	118	484	664	1004	1231	1372	1582
1	30	58	87	115	120	142	145
2	33	69	111	156	172	203	203
3	30	51	75	108	115	139	140
4	32	62	112	167	179	209	214
5	30	49	81	125	142	174	177

```
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(days)"
##   ..$ y: chr "(mm)"
```

说明 5 棵树之间的大小关系是 `3 < 1 < 5 < 2 < 4`，这里的数字 1，2，3，4，5 只是对树的编号，第一次测量时树的大小关系在 R 内用有序因子来表示。

```
levels(Orange$Tree)
```

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

表 33.3 记录了 5 颗橘树自 1968 年 12 月 31 日以来的生长情况

```
# aggregate(data = Orange, circumference ~  age, FUN = mean)
library(magrittr)
reshape(
  data = Orange, v.names = "circumference", idvar = "Tree",
  timevar = "age", direction = "wide", sep = ""
) %>%
  knitr::kable(.,
    caption = "躯干周长（毫米）随时间（天）的变化",
    row.names = FALSE, col.names = gsub("(circumference)", "", names(.)),
    align = "c"
  )
```

图 33.2 以直观的方式展示 5 颗橘树的生长变化，相比于表 33.3 我们能更加明确读取数据中的变化

```
library(ggplot2)
p <- ggplot(data = Orange, aes(x = age, y = circumference, color = Tree)) +
  geom_point() +
  geom_line() +
  theme_minimal() +
  labs(x = "age (day)", y = "circumference (mm)")
p
```

```
library(gganimate)
p + transition_reveal(age)
```

© 黄湘云

图 33.2: 橘树生长模型

© 王建平

33.5 R 包网络分析

首先我们从 CRAN 官网下载 R 包描述信息

```
pdb <- tools::CRAN_package_db()
```

接着，我们可以看看 CRAN 上发布的 R 包数量

```
length(pdb[, "Package"])
```

```
## [1] 19045
```

经过与官网发布的数据来对比，我们发现这里计算的结果与实际不符，多出来了几十个 R 包，所以我们再观察一下是否有重复的 R 包描述信息

```
pdb[, "Package"][duplicated(pdb[, "Package"])]
```

```
##  [1] "boot"       "class"      "cluster"    "codetools"  "foreign"
##  [6] "KernSmooth" "lattice"    "MASS"       "Matrix"     "mgcv"
## [11] "nlme"       "nnet"       "rpart"      "spatial"    "survival"
## [16] "boot"       "class"      "cluster"    "codetools"  "foreign"
## [21] "KernSmooth" "lattice"    "MASS"       "Matrix"     "mgcv"
## [26] "nlme"       "nnet"       "rpart"      "spatial"    "survival"
## [31] "RODBC"      "XML"
```

不难发现，果然有！所以去掉重复的 R 包信息，就是 CRAN 上实际发布的 R 包数量

```
dim(subset(pdb, subset = !duplicated(Package)))[1]
```

```
## [1] 19013
```

接下来就是分析去掉重复信息后的数据矩阵 pdb

```
pdb <- subset(pdb, subset = !duplicated(Package))
```

33.5.1 R 核心团队

R 核心团队除了维护开发 Base R 包以外，还开发了哪些 R 包，我们依据这些开发者邮箱 `<Firstname>.<Lastname>@R-project.o` 的特点，从数据集 pdb 中提取他们开发的 R 包

```
core_pdb <- subset(pdb,
  subset = grepl(
    x = Maintainer,
    pattern = "(@[Rr]-project\\.org)"
  ),
  select = c("Package", "Maintainer")
)
dim(core_pdb[order(core_pdb[, "Maintainer"]), ])
```

```
## [1] 158   2
```

这么少，是不是有点意外，看来很多大佬更喜欢用自己的邮箱，比如 Paul Murrell，他的邮箱是 paul@stat.auckland.ac.nz

© 非常清晰


```
subset(pdb,
  subset = grepl(x = Maintainer, pattern = "(Paul Murrell)"),
  select = c("Package", "Maintainer")
)
```

```
##            Package                              Maintainer
## 2602       compare Paul Murrell <p.murrell@auckland.ac.nz>
## 5931         gdiff Paul Murrell <paul@stat.auckland.ac.nz>
## 6240        gggrid Paul Murrell <paul@stat.auckland.ac.nz>
## 6723      gridBase Paul Murrell <paul@stat.auckland.ac.nz>
## 6724    gridBezier Paul Murrell <paul@stat.auckland.ac.nz>
## 6725     gridDebug Paul Murrell <p.murrell@auckland.ac.nz>
## 6727  gridGeometry Paul Murrell <paul@stat.auckland.ac.nz>
## 6728  gridGraphics Paul Murrell <paul@stat.auckland.ac.nz>
## 6729  gridGraphviz Paul Murrell <p.murrell@auckland.ac.nz>
## 6733       gridSVG Paul Murrell <paul@stat.auckland.ac.nz>
## 6735      grImport Paul Murrell <p.murrell@auckland.ac.nz>
## 6736     grImport2 Paul Murrell <paul@stat.auckland.ac.nz>
## 7067       hexView Paul Murrell <paul@stat.auckland.ac.nz>
## 9566      metapost Paul Murrell <paul@stat.auckland.ac.nz>
## 13355    rasterize Paul Murrell <paul@stat.auckland.ac.nz>
## 14116    RGraphics Paul Murrell <paul@stat.auckland.ac.nz>
## 14556        roloc Paul Murrell <paul@stat.auckland.ac.nz>
## 14557 rolocISCCNBS Paul Murrell <paul@stat.auckland.ac.nz>
## 18537       vwline Paul Murrell <paul@stat.auckland.ac.nz>
```

所以这种方式不行了，只能列举所有 R Core Team 成员，挨个去匹配，幸好 `contributors()` 函数已经收集了成员名单，不需要我们去官网找了。

```
core_team <- read.table(
  text = "
Douglas Bates
John Chambers
Peter Dalgaard
Robert Gentleman
Kurt Hornik
Ross Ihaka
Tomas Kalibera
Michael Lawrence
Friedrich Leisch
Uwe Ligges
Thomas Lumley
Martin Maechler
Martin Morgan
Paul Murrell
Martyn Plummer
```

© 基础统计

```
Brian Ripley
Deepayan Sarkar
Duncan Temple Lang
Luke Tierney
Simon Urbanek
Heiner Schwarte
Guido Masarotto
Stefano Iacus
Seth Falcon
Duncan Murdoch
David Meyer
Simon Wood
", header = FALSE, sep = "\n",
  check.names = FALSE, stringsAsFactors = FALSE,
  colClasses = "character", comment.char = "", col.names = "name"
)
```

R 核心团队维护的 R 包及其最新发布的日期

```
core_pdb <- subset(pdb,
  subset = grepl(
    x = Maintainer,
    pattern = paste("(", core_team$name, ")", collapse = "|", sep = "")
  ),
  select = c("Package", "Maintainer", "Published")
)
```

清理 Maintainer 字段中的邮箱部分，方便表格展示

```
clean_maintainer <- function(x) {
  # 去掉邮箱
  x <- gsub("<([^<>]*)>", "", x)
  # 去掉 \n \t \' \" 和 '
  x <- gsub("(\\\n)|(\\\t)|(\\\")|(\\\')|(')", "", x)
  # 去掉末尾空格
  x <- gsub(" +$", "", x)
}
core_pdb[, "Maintainer"] <- clean_maintainer(core_pdb[, "Maintainer"])
```

我们可以看到 R 核心团队总共开发维护有 173 个 R 包

```
dim(core_pdb)
```

```
## [1] 173   3
```

篇幅所限，就展示部分人和 R 包，见表 33.4 按照拼音顺序 Brian Ripley 是第一位

```
knitr::kable(head(core_pdb[order(
  core_pdb[, "Maintainer"],
  core_pdb[, "Published"]
```

© 黄湘云

表 33.4: R Core Team 维护的 R 包（展示部分）

Package	Maintainer	Published
mix	Brian Ripley	2017-06-12
boot	Brian Ripley	2021-05-03
KernSmooth	Brian Ripley	2021-05-03
tree	Brian Ripley	2021-08-17
RODBC	Brian Ripley	2021-09-16
fastICA	Brian Ripley	2021-09-25

```
), ], 6),
caption = "R Core Team 维护的 R 包（展示部分）",
booktabs = TRUE, row.names = FALSE
)
```

分组计数，看看核心开发者维护的 R 包有多少

```
aggregate(data = core_pdb, Package ~ Maintainer, FUN = length) |>
  ggplot(aes(x = reorder(Maintainer, Package), y = Package)) +
  geom_col() +
  coord_flip() +
  labs(x = "R 核心团队", y = "R 包数量") +
  theme_minimal(base_family = "Noto Serif CJK SC")
```

© 清华大学出版社

表 33.5: 谢益辉维护的 R Markdown 生态

Package	Title
animation	A Gallery of Animations in Statistics and Utilities to Create Animations
blogdown	Create Blogs and Websites with R Markdown
bookdown	Authoring Books and Technical Documents with R Markdown
DT	A Wrapper of the JavaScript Library 'DataTables'
evaluate	Parsing and Evaluation Tools that Provide More Details than the Default
formatR	Format R Code Automatically
fun	Use R for Fun
highr	Syntax Highlighting for R Source Code
knitr	A General-Purpose Package for Dynamic Report Generation in R
markdown	Render Markdown with the C Library 'Sundown'
mime	Map Filenames to MIME Types
MSG	Data and Functions for the Book Modern Statistical Graphics
pagedown	Paginate the HTML Output of R Markdown with CSS for Print
printr	Automatically Print R Objects to Appropriate Formats According to the 'knitr' Output Format
Rd2roxygen	Convert Rd to 'Roxygen' Documentation
rmarkdown	Dynamic Documents for R
rolldown	R Markdown Output Formats for Storytelling
servr	A Simple HTTP Server to Serve Static Files or Dynamic Documents
testit	A Simple Package for Testing R Packages
tinytex	Helper Functions to Install and Maintain TeX Live, and Compile LaTeX Documents
xaringan	Presentation Ninja
xfun	Supporting Functions for Packages Maintained by 'Yihui Xie'

33.5.2 高产的开发者

这些人的个人简介

接下来，我们再来查看一些比较高产的 R 包开发者谢益辉都维护了哪些 R 包，如表 33.5 所示

```
yihui_pdb <- subset(pdb,
  subset = grepl("Yihui Xie", Maintainer),
  select = c("Package", "Title")
)
yihui_pdb[, "Title"] <- gsub("(\\\n)", " ", yihui_pdb[, "Title"])
knitr::kable(yihui_pdb,
  caption = "谢益辉维护的 R Markdown 生态",
  booktabs = TRUE, row.names = FALSE
)
```

Jeroen Ooms 维护从 C++ 世界搬运进来的库，如图像处理 magick 包、视频处理 av 包、PDF 文档操作 qpdf 包

©清华大学出版社


```
subset(pdb, subset = grepl("Jeroen Ooms", Maintainer),
       select = 'Package', drop = TRUE)
```

```
##  [1] "antiword"    "askpass"     "av"          "base64"      "bcrypt"
##  [6] "brotli"      "cld2"        "cld3"        "commonmark"  "credentials"
## [11] "curl"        "gert"        "gifski"      "gpg"         "graphql"
## [16] "hunspell"    "jose"        "jqr"         "js"          "jsonld"
## [21] "jsonlite"    "katex"       "magick"      "maketools"   "minimist"
## [26] "mongolite"   "opencpu"     "opencv"      "openssl"     "pdftools"
## [31] "protolite"   "qpdf"        "RAppArmor"   "rjade"       "RMySQL"
## [36] "rsvg"        "rzmq"        "sodium"      "spelling"    "ssh"
## [41] "sys"         "tesseract"   "unix"        "unrtf"       "V8"
## [46] "webp"        "webutils"    "writexl"     "xslt"
```

Dirk Eddelbuettel 维护 Rcpp 生态

```
subset(pdb, subset = grepl("Dirk Eddelbuettel", Maintainer),
       select = 'Package', drop = TRUE)
```

```
##  [1] "anytime"             "AsioHeaders"         "BH"
##  [4] "binb"                "corels"              "dang"
##  [7] "digest"              "drat"                "dtts"
## [10] "gaussfacts"          "gcbd"                "gettz"
## [13] "gunsales"            "inline"              "linl"
## [16] "littler"             "nanotime"            "pinp"
## [19] "pkgKitten"           "prrd"                "qlcal"
## [22] "random"              "RApiDatetime"        "RApiSerialize"
## [25] "Rblpapi"             "Rcpp"                "RcppAnnoy"
## [28] "RcppAPT"             "RcppArmadillo"       "RcppBDT"
## [31] "RcppCCTZ"            "RcppClassic"         "RcppClassicExamples"
## [34] "RcppCNPy"            "RcppDate"            "RcppDE"
## [37] "RcppEigen"           "RcppExamples"        "RcppFarmHash"
## [40] "RcppFastFloat"       "RcppGetconf"         "RcppGSL"
## [43] "RcppMsgPack"         "RcppNLoptExample"    "RcppQuantuccia"
## [46] "RcppRedis"           "RcppSimdJson"        "RcppSMC"
## [49] "RcppSpdlog"          "RcppStreams"         "RcppTOML"
## [52] "RcppXts"             "RcppZiggurat"        "RDieHarder"
## [55] "rfoaas"              "RInside"             "rmsfact"
## [58] "RProtoBuf"           "RPushbullet"         "RQuantLib"
## [61] "RVowpalWabbit"       "sanitizers"          "td"
## [64] "tidyCpp"             "tiledb"              "tint"
## [67] "ttdo"                "x13binary"
```

Hadley Wickham 维护 tidyverse 生态

```
subset(pdb, subset = grepl("Hadley Wickham", Maintainer),
       select = 'Package', drop = TRUE)
```

©黄湘云

```
##  [1] "assertthat"    "babynames"     "bigrquery"     "classifly"
##  [5] "conflicted"    "cubelyr"       "dbplyr"        "diffviewer"
##  [9] "downlit"       "dplyr"         "dtplyr"        "ellipsis"
## [13] "feather"       "forcats"       "fueleconomy"   "generics"
## [17] "ggplot2movies" "gtable"        "haven"         "hflights"
## [21] "highlight"     "httr"          "httr2"         "lazyeval"
## [25] "lobstr"        "lvplot"        "meifly"        "mockery"
## [29] "modelr"        "multidplyr"    "nasaweather"   "nycflights13"
## [33] "odbc"          "pins"          "pkgdown"       "plyr"
## [37] "productplots"  "profr"         "proto"         "pryr"
## [41] "rappdirs"      "reshape"       "reshape2"      "roxygen2"
## [45] "rvest"         "scales"        "sloop"         "stringr"
## [49] "testthat"      "tidyr"         "tidyverse"     "waldo"
## [53] "xml2"
```

Scott Chamberlain 是非营利性组织 rOpenSci 的联合创始人，但是没几个 R 包听说过

```
subset(pdb, subset = grepl("Scott Chamberlain", Maintainer),
       select = 'Package', drop = TRUE)
```

```
##  [1] "bold"        "brranching"  "charlatan"   "citecorp"    "ckanr"
##  [6] "conditionz"  "cowsay"      "crul"        "discgolf"    "elastic"
## [11] "fauxpas"     "finch"       "geojson"     "geojsonio"   "geojsonlint"
## [16] "getlandsat"  "ghql"        "gistr"       "handlr"      "hoardr"
## [21] "httpcode"    "httping"     "isdparser"   "jaod"        "mapr"
## [26] "microdemic"  "natserv"     "oai"         "openadds"    "pangaear"
## [31] "phylocomr"   "pubchunks"   "randgeo"     "rbace"       "rbhl"
## [36] "rbison"      "rcitoid"     "rcoreoa"     "rcrossref"   "rdatacite"
## [41] "rdryad"      "request"     "rgnparser"   "ritis"       "rorcid"
## [46] "rphylopic"   "rplos"       "rredlist"    "rvertnet"    "sofa"
## [51] "solrium"     "spocc"       "taxize"      "taxizedb"    "traits"
## [56] "vcr"         "webmockr"    "wellknown"   "wikitaxa"    "worrms"
## [61] "zbank"
```

33.5.3 社区开发者

接下来，我们想看看 R 包维护者数量有多少

```
length(unique(pdb[, "Maintainer"]))
```

```
## [1] 11031
```

可实际上没有这么多的开发者，因为存在这样的情况，以 R 包维护者 Hadley Wickham 为例，由于他曾使用过不同的邮箱，所以在维护者字段出现了不一致的情况，实际却是同一个人。

```
subset(pdb,
  subset = grepl("Hadley Wickham", Maintainer),
  select = c("Package", "Maintainer")
)
```

© 草稿


```
##             Package                              Maintainer
## 622       assertthat    Hadley Wickham <hadley@rstudio.com>
## 790        babynames    Hadley Wickham <hadley@rstudio.com>
## 1248       bigrquery    Hadley Wickham <hadley@rstudio.com>
## 2271       classifly   Hadley Wickham <h.wickham@gmail.com>
## 2684      conflicted    Hadley Wickham <hadley@rstudio.com>
....
```

因此，有必要先把 Maintainer 字段中的邮箱部分去掉，这样我们可以得到比较靠谱的 R 包维护者数量了！

```
pdb[, "Maintainer"] <- clean_maintainer(pdb[, "Maintainer"])
length(unique(pdb[, "Maintainer"]))
```

```
## [1] 10202
```

接下来，我们还想把 R 包维护者，按照其维护的 R 包数量排个序，用条形图33.3 表示，其中 Orphaned 表示之前的 R 包维护者不愿意继续维护了，后来有人接手维护，Orphaned 表示这一类接盘侠。

```
top_maintainer <- head(sort(table(pdb[, "Maintainer"]), decreasing = TRUE), 20)

par(mar = c(2, 7, 1, 1))
barCenters <- barplot(top_maintainer,
  col = "lightblue", axes = FALSE,
  axisnames = FALSE, horiz = TRUE, border = "white"
)
text(
  y = barCenters, x = par("usr")[3],
  adj = 1, labels = names(top_maintainer), xpd = TRUE
)
axis(1,
  labels = seq(0, 90, by = 10), at = seq(0, 90, by = 10),
  las = 1, col = "gray"
)
grid()
```

调用 ggplot2 包绘图要求输入的数据类型是 `data.frame`，所以我们首先将 `top_maintainer` 转化为数据框类型

```
top_maintainer <- as.data.frame(top_maintainer)
colnames(top_maintainer) <- c("Maintainer", "Freq")

ggplot(top_maintainer) +
  geom_bar(aes(x = Maintainer, y = Freq), stat = "identity") +
  coord_flip() +
  xlab("Maintainer") +
  ylab("Numbers of Package")
```

© 黄湘云

图 33.3: 维护 R 包数量最多的 20 个人

条形图在柱子很多的情况下，点线图是一种更加简洁的替代方式

```
ggplot(top_maintainer, aes(x = Freq, y = Maintainer)) +
  geom_segment(aes(x = 20, xend = Freq, yend = Maintainer), colour = "grey50") +
  geom_point(size = 2, colour = "red") +
  labs(x = " # of Packages ", y = " Maintainer ")
```

© 黄湘云

接下来，我们想看看开发者维护的 R 包数量的分布，仅从上图，我们知道有的人能维护 80 多个 R 包，总体的分布情况又是如何呢？如图所示，我们将纵轴刻度设置为 log 模式，随着开发的 R 包数量的增加，开发者人数是指数级递减，可见开发 R 包依然是一个门槛很高的工作！

```
barplot(table(table(pdb[, "Maintainer"])),
  col = "lightblue", log = "y", border = "white",
  xlab = "# of Packages", ylab = "# of Maintainers (log)",
  panel.first = grid()
)
```

只开发一个 R 包的人数达到 5276 人，占开发者总数的 67.31%，约为 2/3。

© 黄湘云

33.5.4 首次贡献 R 包

我们还想进一步了解这些人是不是就自己开发自己维护，基本没有其他人参与，答案是 Almost Sure. 这些人其实占了大部分，相比于前面的 R 核心开发团队或者 R Markdown 生态的维护者，他们绝大部分属于金字塔底部的人，二八定律似乎在这里再次得到印证。

```
sub_pdb <- subset(pdb, select = c("Package", "Maintainer", "Author"))
```

接着先清理一下 Maintainer 和 Author 字段，Author 字段的内容比起 Maintainer 复杂一些

```
clean_author <- function(x) {
  # 去掉中括号及其内容 [aut] [aut, cre]
  x <- gsub("(\\[.*?\\])", "", x)
  # 去掉小括号及其内容 ()
  x <- gsub("(\\(.*?\\))", "", x)
  # 去掉尖括号及其内容 < >
  x <- gsub("(<.*?>)", "", x)
  # 去掉 \n
  x <- gsub("(\\\n)", "", x)
  # 去掉制表符、双引号、单引号和 \', 如 'Hadley Wickham' 中的单引号 ' 等
  x <- gsub("(\\\t)|(\\\")|(\\\')|(')|(\\))", "", x)
  # Christian P. Robert, Universite Paris Dauphine, and Jean-Michel\n        Marin, Universite Montpellier
  x <- gsub("(and)", "", x)
  # 两个以上的空格替换为一个空格
  x <- gsub("( {2,})"," ",x)
  x
}

sub_pdb[, "Maintainer"] <- clean_maintainer(sub_pdb[, "Maintainer"])
sub_pdb[, "Author"] <- clean_author(sub_pdb[, "Author"])
```

维护多个 R 包的开发者数量

```
length(unique(sub_pdb[, "Maintainer"][duplicated(sub_pdb[, "Maintainer"])]))
```

```
## [1] 3355
```

总的开发者中去掉开发了多个 R 包的人，就剩下只维护 1 个 R 包的开发者，共有

```
first_ctb <- setdiff(
  sub_pdb[, "Maintainer"][!duplicated(sub_pdb[, "Maintainer"])],
  unique(sub_pdb[, "Maintainer"][duplicated(sub_pdb[, "Maintainer"])])
)
```

按照每个 R 包贡献者的数量分组，如图所示，有一个或者没有贡献者的占总数占 70.60%，说明这些 R 包的开发者基本在单干，有 4 个及以下的贡献者占总数（这个总数是指只开发了一个 R 包的那些开发者）的 90.85%。

```
ctb_num <- unlist(
  lapply(
    strsplit(
```

© 黄湘云


```
      subset(sub_pdb,
             subset = Maintainer %in% first_ctb,
             select = "Author", drop = TRUE # drop out data.frame return vector
      ),
      split = ","
    ), length
  )
)

hist(ctb_num, col = "lightblue", border = "white",
     probability = TRUE, labels = TRUE,
     xlab = "# of Contributors", ylab = "Proportion", main = "",
     panel.first = grid(), xlim = c(0, 100))
```

这些基本单干的 R 包开发者是否参与其它 R 包的贡献？如果不参与，则他们对社区的贡献非常有限，仅限于为社区带来数量上的堆积！

```
table(ctb_num)
```

```
## ctb_num
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 3062 1454  961  600  288  163  104   64   43   24   15   10    7   11    4    5
##   17   18   19   20   22   23   24   26   27   28   29   56   60  133
##    9    4    2    4    2    2    1    2    1    1    1    1    1    1
```

有意思的是，有一个开发者虽然只开发了一个 R 包，但是却引来 37 位贡献者（包括开发者本人在内），下面把这个颇受欢迎的 R 包找出来

```
# 找到开发者
first_ctb[which.max(ctb_num)]
```

```
## [1] "Ryan Curtin"
```

© 基础统计

```
# 找到 R 包
subset(sub_pdb, subset = grepl("Matt Dowle", Maintainer), select = "Package")
```

```
##         Package
## 3287 data.table
```

哇，大名鼎鼎的 data.table 包！！ I JUST find it!! 这是个异数，我们知道 data.table 在 R 社区享有盛名，影响范围很广，从 Matt Dowle 的 Github 主页 来看，他确实只开发了这一个 R 包！黑天鹅在这里出现了！如果按照谁的贡献者多谁影响力大的规律来看，有 10 个以上贡献者的其它几个 R 包也必定是名器！这里留给读者把它找出来吧！

33.5.5 贡献关系网络

接下来进入本节最核心的部分，分析所有的开发者之间的贡献网络，在第33.5.4节清理 Author 字段的正则表达式几乎不可能覆盖到所有的情况，所以既然 Maintainer 字段是比较好清理的，不妨以它作为匹配的模式去匹配 Author 字段，这样做的代价就是迭代次数会很多，增加一定的计算负担，但是为了更加准确的清理结果，也是拼了！

```
net_pdb <- subset(pdb, select = c("Maintainer", "Author"))
net_pdb[, "Maintainer"] <- clean_maintainer(net_pdb[, "Maintainer"])
total_maintainer <- unique(net_pdb[, "Maintainer"])
clean_author <- function(maintainer) {
  sapply(net_pdb[, "Author"], grepl, pattern = paste0("(", maintainer, ")"))
}
```

接下来是非常耗时的一步，实际是两层循环 1.2 亿次左右的查找计算，`grepl` 耗时 30 分钟左右，正则表达式本身的性能优化问题，`maintainer_author` 逻辑型矩阵占用内存空间 430 M 左右

```
maintainer_author <- Reduce("cbind", lapply(total_maintainer, clean_author))
colnames(maintainer_author) <- total_maintainer
rownames(maintainer_author) <- net_pdb[, "Maintainer"]
```

为了重复运行这段耗时很长的代码，我们将中间结果保存到磁盘，推荐保存为 R 支持的序列化后的数据格式 `*.rds`，相比于 `*.csv` 格式能极大地减少磁盘存储空间，读者可运行下面两行保存数据的代码，比较看看！

```
saveRDS(maintainer_author, file = "data/maintainer_author.rds")
write.table(maintainer_author,file = "data/maintainer_author.csv", row.names = TRUE, col.names = TRUE)
```

查看 `maintainer_author` 数据集占用内存空间的大小

```
format(object.size(maintainer_author), units = "auto")
```

看几个数字，R 包贡献者最多的有 62 人，这个 R 包的粉丝是真多！有一个开发者对 137 个 R 包的做出过贡献，其中包括自己开发的 R 包，快来快来抓住他！

```
max(rowSums(maintainer_author))
max(colSums(maintainer_author))
```

继续看看每个开发者对外贡献的量的分布情况，由图可知，绝大部分开发者对外输出不超过 3，其表示对其它 R 包的贡献不超过 3 个

© 王湘君


```
hist(colSums(maintainer_author)[colSums(maintainer_author) <= 10],
     probability = FALSE, xlab = "", main = "")
```

每个 R 包参与贡献的人数分布又是如何呢？如图所示，基本集中在 1~2 个人的样子

```
hist(rowSums(maintainer_author)[rowSums(maintainer_author) <= 20],
     xlab = "", main = "",probability = FALSE)
```

好了，接下来我们要深入挖掘贡献协作网络中的结构特点，看看是不是由几位领导人在完全掌控，还有一大群人其实是自己搞自己的那点事，写论文、发布 R 包、投稿等如此循环。其实这就是 R 社区的特点，也决定了它不会像 Python 那样应用性强，有足够多的工程开发人员加入。大多数人写 R 包只是为了配合发论文而已，并不关心有没有人来用自己的 R 包！此外，没有人来做功能整合和持续维护，所以发展缓慢！各自造轮子的事情太多！

接着，先从表面看看开发者和贡献者的关系矩阵，`maintainer_author` 是一个大型的超稀疏矩阵，非零元素最多的行、列分别只占 0.79% 和 0.95%，都不到百分之一。

```
# 非零元素最多的行
max(rowMeans(maintainer_author))
# 非零元素最多的列
max(colMeans(maintainer_author))
```

> 用稀疏索引的方式重新编码矩阵，然后用社群检测的算法找到其中的结构，网络关系图用 Gephi 画，igraph 肯定是不行了，参考文献 社会网络分析：探索人人网好友推荐系统 网络的统计建模分析 [1]

重新获取 `maintainer_author` 矩阵，存储指标向量，然后调用 Matrix 生成稀疏矩阵，后续的数据操作就好办了，因为 Matrix 包是内置的，它定义的稀疏矩阵类其它 R 包也都支持。先以一个简单的例子说明构造稀疏矩阵的过程

```
library(Matrix)
spM <- spMatrix(3, 4, i = c(1, 1, 2, 3, 3),
                j = c(4, 1, 2, 1, 3),
                x = c(4, 4, 1, 4, 8))
spM
```

```
## 3 x 4 sparse Matrix of class "dgTMatrix"
##
## [1,] 4 . . 4
## [2,] . 1 . .
## [3,] 4 . 8 .
```

```
image(spM)
```

`i` 和 `j` 表示矩阵中有值的位置，`x` 表示对应位置上的值，`i`，`j` 和 `x` 是三个长度相等的数值型向量，我们还可以调用 `image` 函数，把稀疏矩阵可视化出来，对于大型稀疏矩阵可视化其稀疏模式是重要的。

> 贡献网络可视化 [2]

[1] Statistical Modeling of Networks in R https://user2010.org/Invited/handcockuser2010.pdf

[2] Network Analysis and Visualization with R and igraph https://kateto.net/networks-r-igraph with PDF

© 黄湘云

图 33.4: 稀疏矩阵的图表示

```
clean_net_pdb <- function(maintainer) {
  index <- clean_author(maintainer)
  if (sum(index) == 0) {
    return(NULL)
  }
  data.frame(
    from_id = maintainer,
    to_id = net_pdb[, "Maintainer"][index],
    stringsAsFactors = FALSE
  )
}
```

```
# maintainer_author <- data.table::rbindlist(lapply(total_maintainer, clean_net_pdb))
# saveRDS(maintainer_author, file = "data/maintainer_author.rds")
toc <- system.time({
  maintainer_author_net <- Reduce("rbind", lapply(total_maintainer, clean_net_pdb))
}, gcFirst = TRUE)
```

分组统计开发者之间贡献次数，从开发者到

```
maintainer_author_net$weight <- 1
edges <- aggregate(weight ~ from_id + to_id, data = maintainer_author_net, sum)

dup_edges <- edges[edges[, 1] != edges[, 2], ]

library(geomnet)
ggplot(data = dup_edges, aes(from_id = from_id, to_id = to_id)) +
  geom_net(aes(linewidth = weight),
```

© 统计出版社


```
    layout.alg = "kamadakawai",
    labelon = FALSE, directed = TRUE, show.legend = FALSE, ealpha = 1,
    ecolour = "grey70", arrowsize = 0.1, size = 0.5
  ) +
  theme_net()
```

```
# https://smallstats.blogspot.com/2012/12/loading-huge-graphs-with-igraph-and-r.html
library(igraph)
# 贡献矩阵
ctb_df <- graph.data.frame(maintainer_author, directed = TRUE)

vertex.attrs <- list(name = unique(c(ctb_df$from_id, ctb_df$to_id)))
edges <- rbind(
  match(ctb_df$from_id, vertex.attrs$name),
  match(ctb_df$to_id, vertex.attrs$name)
)

ctb_net <- graph.empty(n = 0, directed = T)
ctb_net <- add.vertices(ctb_net, length(vertex.attrs$name), attr = vertex.attrs)
ctb_net <- add.edges(ctb_net, edges)
```

33.5.6 更新知多少

这节标题取其字面意思表达 CRAN 服务器的特殊日子 2012-10-29，那天 CRAN 更新了一大波 R 包，像一根擎天柱一样支撑这幅图！

```
update_pdb <- pdb[, c("Package", "Published")]
# 这天要更新的R包最多
sort(table(update_pdb[,"Published"]), decreasing = TRUE)[1]
```

```
## 2012-10-29
##         74
```

```
ggplot(update_pdb, aes(as.Date(Published))) +
  geom_bar(color = "skyblue4") +
  geom_line(
    data = data.frame(
      date = as.Date(c("2011-01-01", "2012-10-20")),
      count = c(80, 87)
    ), aes(x = date, y = count),
    arrow = arrow(angle = 15, length = unit(0.15, "inches"))
  ) +
  annotate("text", x = as.Date("2010-11-01"), y = 75, label = "(2012-10-29,87)") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(x = "Published Date", y = "Count") +
  theme_minimal()
```

图 33.5: R 包更新历史

当日发布的 R 包，不论是新增还是更新之前发布的 R 包都视为最新版，当日之前的都是旧版本，它们可能存在已经修复的 BUG！这句子好奇怪是吧，因为很多 R 包要么托管在 Github 上，要么托管在 R-Forge 上开发，而 CRAN 上的版本除了发布日外，一般来讲都会落后。如图所示待更新的 R 包在日期上的分布，有的已经 10 来年没有更新了，最老的 R 包可以追溯到 2008-09-08，它是 pack!!

```
subset(update_pdb, subset = Published == min(Published))
```

```
##       Package  Published
## 11529    pack 2008-09-08
```

```
update_pdb[which.min(as.Date(update_pdb[, "Published"])), 1]
```

```
## [1] "pack"
```

33.5.7 使用许可证

> 列举 R 社区使用的许可证及其区别和联系 R 开源还体现在许可证信息，顺便谈谈美国和中国技术封锁，开源社区可能面临的风险

社区主要使用 GPL 及其相关授权协议，因为 R 软件本身也是授权在 GPL-2 或 GPL-3 下

```
license_pdb <- head(sort(table(pdb[, "License"]), decreasing = TRUE), 20)
par(mar = c(2, 12, 0.5, 0))
plot(c(1, 1e1, 1e2, 1e3, 1e4), c(1, 5, 10, 15, 20),
     type = "n",panel.first = grid(),
     ann = FALSE, log = "x", axes = FALSE
)
axis(1,
     at = c(1, 1e1, 1e2, 1e3, 1e4),
     labels = expression(1, 10^1, 10^2, 10^3, 10^4)
```

© 斯坦福


```
)
text(
  y = seq(length(license_pdb)), x = 1, cex = 1, offset = 1,
  pos = 2, labels = names(license_pdb), xpd = TRUE
)
text(1e3, 15, "CRAN")
segments(x0 = 1, y0 = seq(length(license_pdb)),
         x1 = license_pdb, y1 = seq(length(license_pdb)),
         col = "lightblue", lwd = 4)
```

图 33.6: CRAN 上采用的发布协议

```
rforge_pdb <- available.packages(repos = "https://R-Forge.R-project.org")
license_rforge_pdb <- head(sort(table(rforge_pdb[, "License"]), decreasing = TRUE), 20)
par(mar = c(2, 12, 0.5, 0))
plot(c(1, 1e1, 1e2, 1e3), seq(from = 1, to = 20,length.out = 4),
  type = "n",panel.first = grid(),
  ann = FALSE, log = "x", axes = FALSE
)
axis(1,
  at = c(1, 1e1, 1e2, 1e3),
  labels = expression(1, 10^1, 10^2, 10^3)
)

text(
  y = seq(length(license_rforge_pdb)), x = 1, cex = 1, offset = 1,
  pos = 2, labels = names(license_rforge_pdb), xpd = TRUE
)
text(1e2, 15, "R-Forge")
```

© 黄湘云

```
segments(x0 = 1, y0 = seq(length(license_rforge_pdb)),
         x1 = license_rforge_pdb, y1 = seq(length(license_rforge_pdb)),
         lwd = 4, col = "lightblue")
```

图 33.7: R-Forge 开发者采用的发布协议

改进的方向是含义相同的进行合并，这需要研究一下各个许可证，然后使用对比型条形图合并上面两个图

CRAN 会检测 R 包的授权，只有授权协议包含在数据库中的才可以在 CRAN 上发布 https://svn.r-project.org/R/trunk/share/licenses/license.db

© 黄湘云

第三十四章　数据探索

DataExplorer

DALEX 提供探索性模型分析，支持 mlr、caret、keras、h2o 和 xgboost 等一系列统计建模分析的 R 包。

breakDown Model Agnostic Explainers for Individual Predictions

© 黄湘云

第九部分

机器学习

© 王湘君

介绍

机器学习与统计的关系

Torsten Hothorn 大数据和大知识 https://user.math.uzh.ch/hothorn/talks/big_data_Ittingen_2015.pdf
机器学习和随机森林 https://user.math.uzh.ch/hothorn/talks/tf_RAAN_2019.pdf

Machine Learning 的漫画解释 https://xkcd.com/1838/

mlpack 是一个机器学习的算法库，开发站点 https://github.com/mlpack/mlpack 和 文档

mlr3verse 机器学习与 R 语言 tidymodels 统计分析和建模与 R 语言 tidyverse 数据操作和可视化与 R 语言

DoubleML

© 黄湘云

第三十五章　梯度提升机

关于决策树和梯度提升的扩展包/库，近年来层出不穷。2001 年 Jerome H. Friedman 提出梯度提升机后[Friedman, 2001]，2003 年 Greg Ridgeway 开发了 gbm 包，目前 Brandon Greenwell 在维护。gbm 实现了 Freund and Schapire's AdaBoost 算法和 Friedman 的梯度提升机。h2o 是基于 Java 平台的机器学习平台，学习材料 h2o-tutorials。基于决策树的分类和回归方法 caret 和基于模型的提升方法 https://github.com/boost-R 偏向统计学习，侧重各类统计模型，仅提供 R 语言接口。xgboost 目前已然成为做梯度提升的决策树的工业标准，使用案例丰富，中文帮助文档 https://xgboost.apachecn.org/cn/latest/，也提供多种语言接口。类似的还有 compboost，其它比较小众的提升库还有 xLearn。catboost 开源的基于决策树的梯度提升库，支持分类特征，提供 R 和 Python 接口，详见官网 https://catboost.ai。LightGBM 提供了 R 包，微软的工具主要支持 Windows 平台和 VS 编译工具。Python 接口的中文文档 https://lightgbm.apachecn.org/，顺便一提，袁进辉等人开发的LightLDA 是大规模主题建模的框架。

35.1　XGBoost

```
library(xgboost)
```

© 基础

第三十六章　数值优化

R 语言提供了相当多的优化求解器，比较完整的概览见优化视图。本章介绍一些常用的优化算法及其 R 实现，涵盖线性规划、整数规划、二次规划、非线性规划等。

商业优化求解器的能力都覆盖非线性规划（NLP），线性（LP）、二次（QP）和锥规划（SOCP），混合整数线性规划（MILP），多目标优化，最小二乘和方程求解。此外，还有很多文档介绍，LINGO提供用户手册，Matlab 优化工具箱 提供 Optimization 工具箱使用指南，MOSEK (https://www.mosek.com/) 提供 MOSEK 建模食谱，LocalSolver 提供基本使用手册，Gurobi 提供 Gurobi 参考手册，CPLEX Optimization Studio。

开源社区有不少工具，也能求解常见的优化问题，如 Julia 的 JuMP (https://jump.dev/)，Octave (https://www.gnu.org/software/octave/) 内置的优化函数，Python 模块 SciPy 提供 Optimization 优化求解器，cvxopt 凸优化求解器，主要基于内点法，提供 Julia、Python、Matlab 接口，算法介绍见 锥优化 机器学习优化。课程见 Optimization for Machine Learning，书籍见Convex Optimization，相关综述见Convex Optimization: Algorithms and Complexity。

Berwin A. Turlach 开发的 quadprog 主要用于求解二次规划问题。Anqi Fu 开发的 CVXR 可解很多凸优化问题 [Fu et al., 2020]，详见网站 https://cvxr.rbind.io/，Jelmer Ypma 开发的 nloptr 可解无约束和有约束的非线性规划问题 [Ypma, 2020]，GPareto 求解多目标优化问题，帕雷托前沿优化和估计 [Binois and Picheny, 2019]。igraph 可以用来解决最短路径、最大网络流、最小生成树等图优化相关的问题。https://palomar.home.ece.ust.hk/MAFS6010R_lectures/Rsession_solvers.html 提供了一般的求解器介绍。ROI 包力图统一各个求解器的调用接口，打造一个优化算法的基础设施平台。Theußl et al. [2020] 详细介绍了目前优化算法发展情况及 R 社区提供的优化能力。GA 包实现了遗传算法，支持连续和离散的空间搜索，可以并行 [Scrucca, 2013, 2017]，是求解 TSP 问题的重要方法。NMOF 包实现了差分进化、遗传算法、粒子群算法、模拟退火算法等启发式优化算法，还提供网格搜索和贪婪搜索工具，Gilli et al. [2019] 提供了详细的介绍。Nash [2014] 总结了 R 语言环境下最优化问题的最佳实践。RcppEnsmallen 数值优化通用标准的优化方法，前沿最新的优化方法，包含小批量/全批量梯度下降技术、无梯度优化器，约束优化技术。RcppNumerical 无约束数值优化，一维/多维数值积分。

谷歌开源的运筹优化工具 or-tools 提供了约束优化、线性优化、混合整数优化、装箱和背包算法、TSP（Traveling Salesman Problem）、VRP（Vehicle Routing Problem）、图算法（最短路径、最小成本流、最大流等）等算法和求解器。「运筹 OR 帷幄」社区开源的 线性规划 一书值得一看。

```
# 加载 ROI 时不要自动加载插件
Sys.setenv(ROI_LOAD_PLUGINS = FALSE)
library(lpSolve)     # 线性规划求解器
library(ROI)         # 优化工具箱
library(ROI.plugin.alabama)  # 注册 alabama 求解非线性规划
library(ROI.plugin.nloptr)   # 注册 nloptr 求解非线性规划
```

© 基础统计

```
library(ROI.plugin.lpsolve)  # 注册 lpsolve 求解线性规划
library(ROI.plugin.quadprog) # 注册 quadprog 求解二次规划
library(ROI.plugin.scs)      # 注册 scs 求解凸锥规划
library(lattice)    # 图形绘制
library(kernlab)    # 优化问题和机器学习的关系

library(rootSolve)       # 非线性方程
library(BB)              # 非线性方程组
library(deSolve)         # ODE 常微分方程
library(scatterplot3d)   # 三维曲线图

library(shape)
library(ReacTran)        # PDE 偏微分方程
library(PBSddesolve)     # DAE 延迟微分方程

library(nlme)            # 混合效应模型
library(nlmeODE)         # ODE 应用于混合效应模型

library(Sim.DiffProc)    # SDE 随机微分方程

# library(nlmixr)          # Population ODE modeling
```

表 36.1 对目前的优化器按优化问题做了分类

36.1 线性规划

clpAPI 线性规划求解器。glpk 的两个 R 接口 – glpkAPI 和 Rglpk 提供线性规划和混合整数规划的求解能力。lp_solve 的两个 R 接口 – lpSolveAPI 和 lpSolve 也提供类似的能力。ompr 求解混合整数线性规划问题。

举个例子，如下

$$
\begin{aligned}
&\min_x \quad -6x_1 - 5x_2 \\
&s.t. \begin{cases} x_1 + 4x_2 \le 16 \\ 6x_1 + 4x_2 \le 28 \\ 2x_1 - 5x_2 \le 6 \end{cases}
\end{aligned}
$$

写成矩阵形式

$$
\begin{aligned}
&\min_x \quad \begin{bmatrix} -6 \\ -5 \end{bmatrix}^T x \\
&s.t. \begin{cases} \begin{bmatrix} 1 & 4 \\ 6 & 4 \\ 2 & -5 \end{bmatrix} x \le \begin{bmatrix} 16 \\ 28 \\ 6 \end{bmatrix} \end{cases}
\end{aligned}
$$

© 王建国

表 36.1: ROI 插件按优化问题分类

	Linear	Quadratic	Conic	Functional
No				
Box				optimx
Linear	clp*, cbc$^{\star+}$, glpk$^{\star+}$, lpsolve$^{\star+}$, msbinlp$^{\star+}$, symphony$^{\star+}$	ipop, quadprog*, qpoases		
Quadratic		cplex^{+}, gurobi$^{\star+}$, mosek$^{\star+}$, neos^{+}		
Conic			ecos$^{\star+}$, scs*	
Functional				alabama, deoptim, nlminb, nloptr

* 求解器受限于凸优化问题
$^{+}$ 求解器可以处理整型约束

对应成 R 代码如下

```
# lpSolve 添加约束条件
library(lpSolve)
# 目标
f.obj <- c(-6, -5)
# 约束
f.con <- matrix(c(1, 4, 6, 4, 2, -5), nrow = 3, byrow = TRUE)
# 方向
f.dir <- c("<=", "<=", "<=")
# 右手边
f.rhs <- c(16, 28, 6)
res <- lp("min", f.obj, f.con, f.dir, f.rhs)
res$objval
```

```
## [1] -31.4
```

```
res$solution
```

```
## [1] 2.4 3.4
```

©黄湘云

36.2 整数规划

36.2.1 一般整数规划

$$
\begin{aligned}
&\max_x \quad 0.2x_1 + 0.6x_2 \\
&s.t. \begin{cases} 5x_1 + 3x_2 \le 250 \\ -3x_1 + 2x_2 \le 4 \\ x_1, x_2 \ge 0, \quad x_1, x_2 \in \mathbb{Z} \end{cases}
\end{aligned}
$$

```
# 目标
f.obj <- c(0.2, 0.6)
# 约束
f.con <- matrix(c(5, 3, -3, 2), nrow = 2, byrow = TRUE)
# 方向
f.dir <- c("<=", "<=")
# 右手边
f.rhs <- c(250, 4)
# 限制两个变量都是整数
res <- lp("max", f.obj, f.con, f.dir, f.rhs, int.vec=1:2)
res$objval
```

```
## [1] 29.2
```

```
res$solution
```

```
## [1] 26 40
```

36.2.2 0-1 整数规划

$$
\begin{aligned}
&\max_x \quad 0.2x_1 + 0.6x_2 \\
&s.t. \begin{cases} 5x_1 + 3x_2 \le 250 \\ -3x_1 + 2x_2 \le 4 \\ x_1, x_2 \in \{0, 1\} \end{cases}
\end{aligned}
$$

```
# 目标
f.obj <- c(0.2, 0.6)
# 约束
f.con <- matrix(c(5, 3, -3, 2), nrow = 2, byrow = TRUE)
# 方向
f.dir <- c("<=", "<=")
# 右手边
f.rhs <- c(250, 4)
# 限制两个变量都是0-1整数
res <- lp("max", f.obj, f.con, f.dir, f.rhs, int.vec=1:2, all.bin = TRUE)
res$objval
```

```
## [1] 0.8
```

```
res$solution
```

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

36.2.3 混合整数规划

Rsymphony 是混合整数规划求解器 SYMPHONY 的 R 语言接口[1]。

```
library(Rsymphony)
## Simple linear program.
## maximize: 2 x_1 + 4 x_2 + 3 x_3
## subject to: 3 x_1 + 4 x_2 + 2 x_3 <= 60
## 2 x_1 + x_2 + x_3 <= 40
## x_1 + 3 x_2 + 2 x_3 <= 80
## x_1, x_2, x_3 are non-negative real numbers

# 简单线性规划
obj <- c(2, 4, 3)
mat <- matrix(c(3, 2, 1, 4, 1, 3, 2, 1, 2), nrow = 3)
dir <- c("<=", "<=", "<=")
rhs <- c(60, 40, 80)
max <- TRUE
Rsymphony_solve_LP(obj, mat, dir, rhs, max = max)

# 混合整数规划
obj <- c(3, 1, 3)
mat <- matrix(c(-1, 0, 1, 2, 4, -3, 1, -3, 2), nrow = 3)
dir <- c("<=", "<=", "<=")
rhs <- c(4, 2, 3)
max <- TRUE
types <- c("I", "C", "I")
Rsymphony_solve_LP(obj, mat, dir, rhs, types = types, max = max)

# 有边界约束的混合整数规划
## Same as before but with bounds replaced by
## -Inf < x_1 <= 4
## 0 <= x_2 <= 100
## 2 <= x_3 < Inf
bounds <- list(
  lower = list(ind = c(1L, 3L), val = c(-Inf, 2)),
```

[1]以 MacOS 为例安装 symphony 软件

```
brew tap coin-or-tools/coinor
brew install symphony
```

©梁洪文

```
  upper = list(ind = c(1L, 2L), val = c(4, 100))
)
Rsymphony_solve_LP(obj, mat, dir, rhs,
  types = types, max = max,
  bounds = bounds
)
```

一部分变量要求是整数

$$
\begin{aligned}
&\max_x \quad 3x_1 + 7x_2 - 12x_3 \\
&s.t. \begin{cases} 5x_1 + 7x_2 + 2x_3 \le 61 \\ 3x_1 + 2x_2 - 9x_3 \le 35 \\ x_1 + 3x_2 + x_3 \le 31 \\ x_1, x_2 \ge 0, \quad x_2, x_3 \in \mathbb{Z}, \quad x_3 \in [-10, 10] \end{cases}
\end{aligned}
$$

矩阵形式如下

$$
\begin{aligned}
&\min_x \quad \begin{bmatrix} 3 \\ 7 \\ -12 \end{bmatrix}^T x \\
&s.t. \begin{cases} \begin{bmatrix} 5 & 7 & 2 \\ 3 & 2 & -9 \\ 1 & 3 & 1 \end{bmatrix} x \le \begin{bmatrix} 61 \\ 35 \\ 31 \end{bmatrix} \end{cases}
\end{aligned}
$$

```
op <- OP(
  objective = L_objective(c(3, 7, -12)),
  # 指定变量类型：第1个变量是连续值，第2、3个变量是整数
  types = c("C", "I", "I"),
  constraints = L_constraint(
    L = matrix(c(
      5, 7, 2,
      3, 2, -9,
      1, 3, 1
    ), ncol = 3, byrow = TRUE),
    dir = c("<=", "<=", "<="),
    rhs = c(61, 35, 31)
  ),
  # 添加约束：第3个变量的下、上界分别是 -10 和 10
  bounds = V_bound(li = 3, ui = 3, lb = -10, ub = 10, nobj = 3),
  maximum = TRUE
)
op
```

```
## ROI Optimization Problem:
##
## Maximize a linear objective function of length 3 with
```

© 王建平


```
## - 1 continuous objective variable,
## - 2 integer objective variables,
##
## subject to
## - 3 constraints of type linear.
## - 1 lower and 1 upper non-standard variable bound.
```

```
res <- ROI_solve(op, solver = "lpsolve")
res$solution
```

```
## [1]  0.3333333  8.0000000 -2.0000000
```

```
res$objval
```

```
## [1] 81
```

36.3 二次规划

36.3.1 凸二次规划

在 R 中使用 **quadprog** [Turlach, 2019] 包求解二次规划[2]，**quadprogXT** 包用来求解带绝对值约束的二次规划，**pracma** [Borchers, 2021] 包提供 `quadprog()` 函数就是对 **quadprog** 包的 `solve.QP()` 进行封装，调用风格更像 Matlab。**quadprog** 包实现了 Goldfarb and Idnani (1982, 1983) 提出的对偶方法，主要用来求解带线性约束的严格凸二次规划问题。quadprog 求解的二次型的形式如下：

$$\min_b -d^\top b + \frac{1}{2} b^\top D b, \quad A^\top b \geq b_0$$

```
solve.QP(Dmat, dvec, Amat, bvec, meq = 0, factorized = FALSE)
```

参数 `Dmat`、`dvec`、`Amat`、`bvec` 分别对应二次规划问题中的 D, d, A, b_0。下面举个二次规划的具体例子

$$D = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \quad d = (-3, 2), \quad A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 0 & -1 \end{bmatrix}, \quad b_0 = (2, -2, -3)$$

即目标函数

$$Q(x, y) = x^2 + y^2 - xy + 3x - 2y + 4$$

它的可行域如图36.1所示

```
plot(0, 0,
  xlim = c(-2, 5.5), ylim = c(-1, 3.5), type = "n",
  xlab = "x", ylab = "y", main = "Feasible Region"
)
polygon(c(2, 5, -1), c(0, 3, 3), border = TRUE, lwd = 2, col = "gray")
```

调用 **quadprog** 包的 `solve.QP()` 函数求解此二次规划问题

[2] https://rwalk.xyz/solving-quadratic-progams-with-rs-quadprog-package/

©黄湘云

图 36.1: 可行域

```
library(quadprog)
Dmat <- matrix(c(2, -1, -1, 2), nrow = 2, byrow = TRUE)
dvec <- c(-3, 2)
A <- matrix(c(1, 1, -1, 1, 0, -1), ncol = 2, byrow = TRUE)
bvec <- c(2, -2, -3)
Amat <- t(A)
sol <- solve.QP(Dmat = Dmat, dvec = dvec, Amat = Amat, bvec = bvec)
sol
```

```
## $solution
## [1] 0.1666667 1.8333333
##
## $value
## [1] -0.08333333
##
## $unconstrained.solution
## [1] -1.3333333  0.3333333
##
## $iterations
## [1] 2 0
##
## $Lagrangian
## [1] 1.5 0.0 0.0
##
```

© 基础统计


```
## $iact
## [1] 1
```

ROI 默认的二次规划的标准形式为 $\frac{1}{2}x^\top Qx + a^\top x$，在传递参数值的时候注意和上面的区别。

```
library(ROI)
op <- OP(
  objective = Q_objective(Q = Dmat, L = -dvec),
  constraints = L_constraint(A, rep(">=", 3), bvec),
  maximum = FALSE # 默认求最小
)
nlp <- ROI_solve(op, solver = "nloptr.slsqp", start = c(1, 2))
nlp$objval
```

```
## [1] -0.08333333
```

```
nlp$solution
```

```
## [1] 0.1666667 1.8333333
```

对变量 x 添加整型约束，原二次规划即变成混合整数二次规划（Mixed Integer Quadratic Programming，简称 MIQP）

```
# 目前开源的求解器都不能处理 MIQP 问题
op <- OP(
  objective = Q_objective(Q = Dmat, L = -dvec),
  constraints = L_constraint(A, rep(">=", 3), bvec),
  types = c("I", "C"),
  maximum = FALSE # 默认求最小
)
nlp <- ROI_solve(op, solver = "nloptr.slsqp", start = c(1, 2))
nlp$objval
nlp$solution
```

在可行域上画出等高线，标记目标解的位置，图中红点表示无约束下的解，黄点表示线性约束下的解

```
qp_sol <- sol$solution # 二次规划的解
uc_sol <- sol$unconstrained.solution # 无约束情况下的解
# 画图
library(lattice)
x <- seq(-2, 5.5, length.out = 500)
y <- seq(-1, 3.5, length.out = 500)
grid <- expand.grid(x = x, y = y)
# 二次规划的目标函数
grid$z <- with(grid, x^2 + y^2 - x * y + 3 * x - 2 * y + 4)
levelplot(z ~ x * y, grid,
  cuts = 40,
  panel = function(...) {
    panel.levelplot(...)
    panel.polygon(c(2, 5, -1), c(0, 3, 3),
```

© 黄湘云


```
      border = TRUE,
      lwd = 2, col = "transparent"
    )
    panel.points(
      c(uc_sol[1], qp_sol[1]),
      c(uc_sol[2], qp_sol[2]),
      lwd = 5, col = c("red", "yellow"), pch = 19
    )
  },
  colorkey = TRUE,
  col.regions = terrain.colors(40)
)
```

图 36.2: 无约束和有约束条件下的解

36.3.2 半正定二次优化

kernlab 提供基于核的机器学习方法，可用于分类、回归、聚类、异常检测、分位回归、降维等场景，包含支撑向量机、谱聚类、核 PCA、高斯过程和二次规划求解器，将优化方法用于机器学习，展示二者的关系。

R 包 kernlab 的函数 `ipop()` 实现内点法可以求解半正定的二次规划问题，对应到上面的例子，就是要求 $A \geq 0$，而 R 包 quadprog 只能求解正定的二次规划问题，即要求 $A > 0$。

以二分类问题为例，采用 SMO (Sequential Minimization Optimization) 求解器，将 SVM 的二次优化问题分解。

© 王湘君


```
library(kernlab)
set.seed(123)
x <- rbind(matrix(rnorm(120), 60, 2), matrix(rnorm(120, mean = 3), 60, 2))
y <- matrix(c(rep(1, 60), rep(-1, 60)))
svp <- ksvm(x, y, type = "C-svc")
plot(svp, data = x)
```

图 36.3: 二分类问题

36.4 非线性规划

开源的非线性优化求解器，推荐使用 nloptr，它支持全局优化，同时推荐 ROI，它有统一的接口函数。

36.4.1 一元非线性优化

下面考虑一个稍微复杂的一元函数优化问题，求复合函数的极值

$$g(x) = \int_0^x -\sqrt{t}\exp(-t^2)\mathrm{d}t, \quad f(y) = \int_0^y g(s)\exp(-s)\mathrm{d}s$$

© 基础统计


```
g <- function(x) {
  integrate(function(t) {
    -sqrt(t) * exp(-t^2)
  }, lower = 0, upper = x)$value
}

f <- function(y) {
  integrate(function(s) {
    Vectorize(g, "x")(s) * exp(-s)
  }, lower = 0, upper = y)$value
}

optimize(f, interval = c(10, 100), maximum = FALSE)
```

```
## $minimum
## [1] 66.84459
##
## $objective
## [1] -0.3201572
```

提示

计算积分的时候，输入了一系列 s 值，参数是向量，而函数 g 只支持输入参数是单个值，`g(c(1,2))` 会报错，因此上面对函数 `g()` 用了向量化函数 `Vectorize()` 操作。

```
g(1)
```

```
## [1] -0.453392
```

类似地，同时计算多个目标函数 `f(y)` 的值，也需要 `Vectorize()` 实现向量化操作。

```
Vectorize(f, "y")(c(1, 2))
```

```
## [1] -0.1103310 -0.2373865
```

36.4.2 多元非线性无约束优化

下面这些用来测试优化算法的函数来自维基百科

36.4.2.1 Himmelblau 函数

Himmelblau 函数是一个多摸函数，常用于比较优化算法的优劣。

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

它在四个位置取得一样的极小值，分别是 $f(-3.7793, -3.2832) = 0$，$f(-2.8051, 3.1313) = 0$，$f(3, 2) = 0$，$f(3.5844, -1.8481) = 0$。函数图像见图 36.4。

```
# 目标函数
fn <- function(x) {
```

©清华大学出版社

```
  (x[1]^2 + x[2] - 11)^2 + (x[1] + x[2]^2 - 7)^2
}

df <- expand.grid(
  x = seq(-5, 5, length = 101),
  y = seq(-5, 5, length = 101)
)

df$fnxy = apply(df, 1, fn)

library(lattice)
# 减少三维图形的边空
lattice.options(
  layout.widths = list(
    left.padding = list(x = -.6, units = "inches"),
    right.padding = list(x = -1.0, units = "inches")
  ),
  layout.heights = list(
    bottom.padding = list(x = -.8, units = "inches"),
    top.padding = list(x = -1.0, units = "inches")
  )
)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = -240, x = -70, y = 0)
)
```

```
# 梯度函数
gr <- function(x) {
  numDeriv::grad(fn, c(x[1], x[2]))
}
optim(par = c(-1.2, 1), fn = fn, gr = gr, method = "BFGS")
```

```
## $par
## [1] -2.805118  3.131313
##
## $value
## [1] 2.069971e-27
```

©黄湘云

图 36.4: Himmelblau 函数图像

© 王建国


```
##
## $counts
## function gradient
##       42       15
##
## $convergence
## [1] 0
##
## $message
## NULL
```

36.4.2.2 Peaks 函数

测试函数

$$f(x,y) = 3*(1-x)*\mathrm{e}^{-x^2-(y+1)^2} - 10*(\frac{x}{5} - x^3 - y^5)*\mathrm{e}^{-x^2-y^2} - \frac{1}{3}*\mathrm{e}^{-(x+1)^2-y^2}$$

```
peaks <- expression(3*(1-x)*exp^(-x^2 - (y+1)^2) - 10*(x/5 - x^3 - y^5)*exp^(-x^2-y^2) -1/3*exp^(-(x-
```

```
D(peaks, "x")
```

```
## -(3 * (1 - x) * (exp^(-x^2 - (y + 1)^2) * (log(exp) * (2 * x))) +
##     3 * exp^(-x^2 - (y + 1)^2) + (10 * (1/5 - 3 * x^2) * exp^(-x^2 -
##     y^2) - 10 * (x/5 - x^3 - y^5) * (exp^(-x^2 - y^2) * (log(exp) *
##     (2 * x)))) - 1/3 * (exp^(-(x + 1)^2 - y^2) * (log(exp) *
##     (2 * (x + 1)))))
```

```
D(peaks, "y")
```

```
## -(3 * (1 - x) * (exp^(-x^2 - (y + 1)^2) * (log(exp) * (2 * (y +
##     1)))) - (10 * (x/5 - x^3 - y^5) * (exp^(-x^2 - y^2) * (log(exp) *
##     (2 * y))) + 10 * (5 * y^4) * exp^(-x^2 - y^2)) - 1/3 * (exp^(-(x +
##     1)^2 - y^2) * (log(exp) * (2 * y))))
```

```
library(Deriv)
Simplify(D(peaks, "x"))
```

```
## -(10 * ((0.2 - 3 * x^2)/exp^(x^2 + y^2)) + 3/exp^((1 + y)^2 +
##     x^2) + log(exp) * (x * (6 * ((1 - x)/exp^((1 + y)^2 + x^2)) -
##     20 * ((x * (0.2 - x^2) - y^5)/exp^(x^2 + y^2))) - 0.666666666666667 *
##     ((1 + x)/exp^((1 + x)^2 + y^2))))
```

```
Simplify(D(peaks, "y"))
```

```
## -((6 * ((1 - x) * (1 + y)/exp^((1 + y)^2 + x^2)) - 0.666666666666667 *
##     (y/exp^((1 + x)^2 + y^2))) * log(exp) - y * (20 * (log(exp) *
##     (x * (0.2 - x^2) - y^5)/exp^(x^2 + y^2)) + 50 * (y^3/exp^(x^2 +
##     y^2))))
```

内部资料 ©

```
fn <- function(x) {
  3 * (1 - x[1])^2 * exp(-x[1]^2 - (x[2] + 1)^2) -
    10 * (x[1] / 5 - x[1]^3 - x[2]^5) * exp(-x[1]^2 - x[2]^2) -
    1 / 3 * exp(-(x[1] + 1)^2 - x[2]^2)
}
# 梯度函数
gr <- function(x) {
  numDeriv::grad(fn, c(x[1], x[2]))
}

optim(par = c(-1.2, 1), fn = fn, gr = gr, method = "BFGS")
```

```
## $par
## [1] -1.3473958  0.2045192
##
## $value
## [1] -3.049849
##
## $counts
## function gradient
##       28       10
##
## $convergence
## [1] 0
##
## $message
## NULL
```

在 $(-1.3473958, 0.2045192)$ 处取得极小值

```
df <- expand.grid(
  x = seq(-3, 3, length = 101),
  y = seq(-3, 3, length = 101)
)

df$fnxy = apply(df, 1, fn)

library(lattice)
wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
```

© 黄湘云


```
  screen = list(z = -240, x = -70, y = 0)
)
```

图 36.5: Peaks 多峰图像

函数来自 Octave 内置的 `peaks()` 函数，它有很多的局部极大值和极小值，可在 Octave Online 上输入命令 `help peaks` 查看其帮助文档。

36.4.2.3 Rosenbrock 函数

香蕉函数 定义如下：

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

```
fn <- function(x) {
  (100 * (x[2] - x[1]^2)^2 + (1 - x[1])^2)
}

df <- expand.grid(
  x = seq(-2.5, 2.5, length = 101),
  y = seq(-2.5, 2.5, length = 101)
```

© 黄湘


```
)
df$fnxy = apply(df, 1, fn)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = 120, x = -70, y = 0)
)
```

图 36.6: 香蕉函数图像

```
r <- raster::rasterFromXYZ(df, crs = CRS("+proj=longlat +datum=WGS84"))
rasterVis::vectorplot(r, par.settings = RdBuTheme())
```

```
# 梯度函数
gr <- function(x) {
  numDeriv::grad(fn, c(x[1], x[2]))
```

© 草稿


```
}
optim(par = c(-1.2, 1), fn = fn, gr = gr, method = "BFGS")
```

```
## $par
## [1] 1 1
##
## $value
## [1] 9.595012e-18
##
## $counts
## function gradient
##      110       43
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```
op <- OP(
  objective = F_objective(fn, n = 2L, G = gr),
  bounds = V_bound(ld = -3, ud = 3, nobj = 2L)
)
nlp <- ROI_solve(op, solver = "nloptr.lbfgs", start = c(-1.2, 1))
nlp$objval
```

```
## [1] 1.364878e-17
```

```
nlp$solution
```

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

36.4.2.4 Ackley 函数

Ackley 函数是一个非凸函数，有大量局部极小值点，获取全局极小值点是一个比较有挑战的事。它的 n 维形式如下：

$$f(\mathbf{x}) = -a\mathrm{e}^{-b\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - \mathrm{e}^{\frac{1}{n}\sum_{i=1}^{n}\cos(cx_i)} + a + \mathrm{e}$$

其中，$a = 20, b = 0.2, c = 2\pi$，对 $\forall i = 1, 2, \cdots, n$，$x_i \in [-10, 10]$，$f(\mathbf{x})$ 在 $\mathbf{x}^\star = (0, 0, \cdot, 0)$ 取得全局最小值 $f(\mathbf{x}^\star) = 0$，二维图像如图 36.7。

```
fn <- function(x, a = 20, b = 0.2, c = 2 * pi) {
  mean1 <- mean(x^2)
  mean2 <- mean(cos(c * x))
  -a * exp(-b * sqrt(mean1)) - exp(mean2) + a + exp(1)
}

df <- expand.grid(
```

© 王湘君

```
  x = seq(-10, 10, length.out = 201),
  y = seq(-10, 10, length.out = 201)
)

df$fnxy = apply(df, 1, fn)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = 120, x = -70, y = 0)
)
```

图 36.7: 二维 Ackley 函数图像

以 10 维的 Ackley 函数为例，先试一下普通的局部优化算法 — Nelder–Mead 算法，选择初值 $(2, 2, \cdots, 2)$，看下效果，再与全局优化算法比较。

© 草稿


```
op <- OP(
  objective = F_objective(fn, n = 10L),
  bounds = V_bound(ld = -10, ud = 10, nobj = 10L)
)

nlp <- ROI_solve(op, solver = "nloptr.neldermead", start = rep(2, 10))
nlp$solution
```

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

```
nlp$objval
```

```
## [1] 6.593599
```

可以说完全没有优化效果，已经陷入局部极小值。根据nloptr 全局优化算法的介绍，这里采用 directL 算法，因为是全局优化，不用选择初值。

```
# 调全局优化器
nlp <- ROI_solve(op, solver = "nloptr.directL")
nlp$solution
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

```
nlp$objval
```

```
## [1] 4.440892e-16
```

```
fn(x = c(2, 2))
```

```
## [1] 6.593599
```

```
fn(x = rep(2, 10))
```

```
## [1] 6.593599
```

36.4.2.5 Radistrigin 函数

这里，还有另外一个例子，Radistrigin 函数也是多摸函数

$$f(\mathbf{x}) = \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$$

```
fn <- function(x) {
  sum(x^2 - 10 * cos(2 * pi * x) + 10)
}

df <- expand.grid(
  x = seq(-4, 4, length.out = 201),
  y = seq(-4, 4, length.out = 201)
)
```

© 基础统计


```
df$fnxy = apply(df, 1, fn)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = 120, x = -65, y = 0)
)
```

图 36.8: Radistrigin 函数

设置 10 维的优化

```
op <- OP(
  objective = F_objective(fn, n = 10L),
  bounds = V_bound(ld = -50, ud = 50, nobj = 10L)
)
```

© 基础统计

调全局优化器求解非凸优化问题

```
nlp <- ROI_solve(op, solver = "nloptr.directL")
nlp$solution
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

```
nlp$objval
```

```
## [1] 0
```

36.4.2.6 Schaffer 函数

$$f(x_1, x_2) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$$

在 $\mathbf{x}^\star = (0, 0)$ 处取得全局最小值 $f(\mathbf{x}^\star) = 0$

```
fn <- function(x) {
  0.5 + ((sin(x[1]^2 - x[2]^2))^2 - 0.5) / (1 + 0.001*(x[1]^2 + x[2]^2))^2
}

df <- expand.grid(
  x = seq(-50, 50, length = 201),
  y = seq(-50, 50, length = 201)
)
df$fnxy = apply(df, 1, fn)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = 120, x = -70, y = 0)
)
```

```
df <- expand.grid(
  x = seq(-2, 2, length = 101),
  y = seq(-2, 2, length = 101)
)
df$fnxy = apply(df, 1, fn)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
```

© 黄湘云

图 36.9: Schaffer 函数

© 黄湘云


```
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = 120, x = -70, y = 0)
)
```

图 36.10: Schaffer 函数

36.4.2.7 Hölder 函数

Hölder 桌面函数

$$f(x_1, x_2) = -|\sin(x_1)\cos(x_2)\exp\left(|1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}|\right)|$$

在 $(8.05502, 9.66459)$、$(-8.05502, 9.66459)$、$(8.05502, -9.66459)$、$(-8.05502, -9.66459)$ 同时取得最小值 -19.2085。

```
fn <- function(x) {
  -abs(sin(x[1]) * cos(x[2])) * exp(abs(1 - sqrt(x[1]^2 + x[2]^2) / pi))
}
```

© 黄湘


```
df <- expand.grid(
  x = seq(-10, 10, length = 101),
  y = seq(-10, 10, length = 101)
)
df$fnxy = apply(df, 1, fn)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = 120, x = -60, y = 0)
)
```

图 36.11: Hölder 函数

© 基础模型

36.4.2.8 Trid 函数

$n \geq 2$ 维 Trid 函数

$$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$$

$\forall i = 1, 2, \cdots, n$，$f(x)$ 在 $x_i = i(n+1-i)$ 处取得全局极小值 $f(\mathbf{x}^\star) = -n(n+4)(n-1)/6$，取值区间 $x \in [-n^2, n^2], \forall i = 1, 2, \cdots, n$

```
fn <- function(x) {
  n <- length(x)
  sum((x - 1)^2) - sum(x[-1] * x[-n])
}

df <- expand.grid(
  x = seq(-4, 4, length = 101),
  y = seq(-4, 4, length = 101)
)
df$fnxy = apply(df, 1, fn)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = -60, x = -70, y = 0)
)
```

36.4.2.9 超级复杂函数

有如下复杂的目标函数

$$\begin{aligned} \min_x \quad & \cos(x_1)\cos(x_2) - \textstyle\sum_{i=1}^{5}\Big((-1)^i \cdot i \cdot 2 \cdot \exp\big(-500 \cdot ((x_1 - i \cdot 2)^2 + (x_2 - i \cdot 2)^2)\big)\Big) \\ s.t. \quad & -50 \leq x_1, x_2 \leq 50 \end{aligned}$$

```
subfun <- function(i, m) {
  (-1)^i * i * 2 * exp(-500 * ((m[1] - i * 2)^2 + (m[2] - i * 2)^2))
}

fn <- function(x) {
  cos(x[1]) * cos(x[2]) -
    sum(mapply(FUN = subfun, i = 1:5, MoreArgs = list(m = x)))
}
```

© 黄湘云

图 36.12: Trid 函数

© 黄湘云

目标函数的图像见图 36.13，搜索区域 $[-50, 50] \times [-50, 50]$ 内几乎没有变化的梯度，给寻优过程带来很大困难。

```
df <- expand.grid(
  x = seq(-50, 50, length.out = 101),
  y = seq(-50, 50, length.out = 101)
)

df$fnxy = apply(df, 1, fn)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = 120, x = -65, y = 0)
)
```

图 36.13: 函数图像

© 黄湘云

将区域 $[0,12]\times[0,12]$ 的图像绘制出来，不难发现，有不少局部陷阱。

```
df <- expand.grid(
  x = seq(0, 12, length.out = 201),
  y = seq(0, 12, length.out = 201)
)

df$fnxy = apply(df, 1, fn)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(x[1]),
  ylab = expression(x[2]),
  zlab = list(expression(italic(f) ~ group("(", list(x[1], x[2]), ")")), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = 120, x = -65, y = 0)
)
```

图 36.14: 局部放大函数图像

© 王建湘

最优解在 $(7.999982, 7.999982)$ 取得，目标函数值为 -7.978832。

```
fn(x = c(7.999982, 7.999982))
```

```
## [1] -7.978832
```

面对如此复杂的函数，调用全局优化器

```
op <- OP(
  objective = F_objective(fn, n = 2L),
  bounds = V_bound(ld = -50, ud = 50, nobj = 2L)
)
nlp <- ROI_solve(op, solver = "nloptr.directL")
nlp$solution
```

```
## [1] 22.22222  0.00000
```

```
nlp$objval
```

```
## [1] -0.9734211
```

实际上，还是陷入局部最优解。

```
SETS:
P/1..5/;
Endsets
Min=@cos(x1) * @cos(x2) - @Sum(P(j): (-1)^j * j * 2 * @exp(-500 * ((x1 - j * 2)^2 + (x2 - j * 2)^2)));
@Bnd(-50, x1, 50);
@Bnd(-50, x2, 50);
```

Lingo 18.0 启用全局优化求解器后，在 $(x_1 = 7.999982, x_2 = 7.999982)$ 取得最小值 -7.978832。而默认未启用全局优化求解器的情况下，在 $(x_1 = 18.84956, x_2 = -40.84070)$ 取得局部极小值 -1.000000。

36.4.3 多元非线性约束优化

R 自带的函数 `nlminb()` 可求解无约束、箱式约束优化问题，`constrOptim()` 还可求解线性不等式约束优化，其中包括带线性约束的二次规划。`optim()` 提供一大类优化算法，且包含随机优化算法—模拟退火算法，可求解无约束、箱式约束优化问题。

36.4.3.1 普通箱式约束

有如下箱式约束优化问题，目标函数和香蕉函数有些相似。

$$\begin{array}{ll} \min_x & (x_1-1)^2 + 4\sum_{i=1}^{n-1}(x_{i+1}-x_i^2)^2 \\ s.t. & 2 \le x_1, x_2, \cdots, x_n \le 4 \end{array}$$

```
fn <- function(x) {
  n <- length(x)
  sum(c(1, rep(4, n - 1)) * (x - c(1, x[-n])^2)^2)
}
```

©黄湘云

n 维目标函数是非线性的，给定初值 $(3, 3, \cdots, 3)$，下面求解 25 维的箱式约束，

```
nlminb(start = rep(3, 25), objective = fn, lower = rep(2, 25), upper = rep(4, 25))
```

```
## $par
##  [1] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
##  [9] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
## [17] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.109093
## [25] 4.000000
##
## $objective
## [1] 368.1059
##
## $convergence
## [1] 0
##
## $iterations
## [1] 6
##
## $evaluations
## function gradient
##       10      177
##
## $message
## [1] "relative convergence (4)"
```

nlminb() 出于历史兼容性的原因尚且存在，最优解的第 24 个分量没有在可行域的边界上。使用 constrOptim() 函数求解，默认求极小，需将箱式或线性不等式约束写成矩阵形式，即 $Ax \geq b$ 的形式，参数 ui 是 $k \times n$ 的约束矩阵 A，ci 是右侧 k 维约束向量 b。以上面的优化问题为例，将箱式约束 $2 \leq x_1, x_2 \leq 4$ 转化为矩阵形式，约束矩阵和向量分别为：

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad b = (2, 2, -4, -4)$$

```
constrOptim(
  theta = rep(3, 25), # 初始值
  f = fn, # 目标函数
  method = "Nelder-Mead", # 没有提供梯度，则必须用 Nelder-Mead 方法
  ui = rbind(diag(rep(1, 25)), diag(rep(-1, 25))),
  ci = c(rep(2, 25), rep(-4, 25))
)
```

```
## $par
##  [1] 2.006142 2.002260 2.003971 2.003967 2.004143 2.004255 2.001178 2.002990
##  [9] 2.003883 2.006029 2.017345 2.009236 2.000949 2.007793 2.025831 2.007896
## [17] 2.004514 2.004381 2.008771 2.015695 2.005803 2.009127 2.017988 2.257782
```

©黄湘云


```
## [25] 3.999846
##
## $value
## [1] 378.4208
##
## $counts
## function gradient
##    12048       NA
##
## $convergence
## [1] 1
##
## $message
## NULL
##
## $outer.iterations
## [1] 25
##
## $barrier.value
## [1] -0.003278963
```

从求解的结果来看，convergence = 1 意味着迭代次数到达默认的极限 maxit = 500，结合 `nlminb()` 函数的求解结果来看，实际上还没有收敛。如果没有提供梯度，则必须用 Nelder-Mead 方法，下面增加迭代次数到 1000。

```
constrOptim(
  theta = rep(3, 25), # 初始值
  f = fn, # 目标函数
  method = "Nelder-Mead",
  control = list(maxit = 1000),
  ui = rbind(diag(rep(1, 25)), diag(rep(-1, 25))),
  ci = c(rep(2, 25), rep(-4, 25))
)
```

```
## $par
##  [1] 2.000081 2.000142 2.001919 2.000584 2.000007 2.000003 2.001097 2.001600
##  [9] 2.000207 2.000042 2.000250 2.000295 2.000580 2.002165 2.000453 2.000932
## [17] 2.000456 2.000363 2.000418 2.000474 2.009483 2.001156 2.003173 2.241046
## [25] 3.990754
##
## $value
## [1] 370.8601
##
## $counts
## function gradient
##    18036       NA
##
```

©黄湘云

```
## $convergence
## [1] 1
##
## $message
## NULL
##
## $outer.iterations
## [1] 19
##
## $barrier.value
## [1] -0.003366467
```

还是没有收敛，可见 Nelder-Mead 方法在这个优化问题上收敛速度比较慢。下面考虑调用基于梯度的优化算法 — BFGS 方法。

```r
# 输入 n 维向量，输出 n 维向量
gr <- function(x) {
  n <- length(x)
  c(2 * (x[1] - 2), rep(0, n - 1))
  +8 * c(0, x[-1] - x[-n]^2)
  -16 * c(x[-n], 0) * c(x[-1] - x[-n]^2, 0)
}

constrOptim(
  theta = rep(3, 25), # 初始值
  f = fn, # 目标函数
  grad = gr,
  method = "BFGS",
  control = list(maxit = 1000),
  ui = rbind(diag(rep(1, 25)), diag(rep(-1, 25))),
  ci = c(rep(2, 25), rep(-4, 25))
)
```

```
## $par
##  [1] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
##  [9] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
## [17] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000001
## [25] 3.000000
##
## $value
## [1] 373
##
## $counts
## function gradient
##     3721      464
##
## $convergence
```

©黄湘云

```
## [1] 0
##
## $message
## NULL
##
## $outer.iterations
## [1] 3
##
## $barrier.value
## [1] -0.003327104
```

相比于 Nelder-Mead 方法，目标值 373 更大，可见已陷入局部最优解，下面通过 ROI 包，分别调用求解器 L-BFGS 和 directL，发现前者同样陷入局部最优解，而后者可以获得与 `nlminb()` 函数一致的结果。

```
# 调用改进的 BFGS 算法
op <- OP(
  objective = F_objective(fn, n = 25L, G = gr),
  bounds = V_bound(ld = 2, ud = 4, nobj = 25L)
)
nlp <- ROI_solve(op, solver = "nloptr.lbfgs", start = rep(3, 25))
nlp$objval
```

```
## [1] 373
```

```
nlp$solution
```

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

```
# 调全局优化算法
nlp <- ROI_solve(op, solver = "nloptr.directL")
nlp$objval
```

```
## [1] 368.1061
```

```
nlp$solution
```

```
##  [1] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
##  [9] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
## [17] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.109093
## [25] 4.000000
```

下面再与函数 `optim()` 提供的 L-BFGS-B 算法比较

```
optim(
  par = rep(3, 25), fn = fn, gr = NULL, method = "L-BFGS-B",
  lower = rep(2, 25), upper = rep(4, 25)
)
```

```
## $par
##  [1] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
##  [9] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000
```

© 基础统计

```
## [17] 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.000000 2.109093
## [25] 4.000000
##
## $value
## [1] 368.1059
##
## $counts
## function gradient
##        6        6
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

值得注意的是，当提供梯度信息的时候，虽然求解速度提升了，但是最优解变差了。

```
optim(
  par = rep(3, 25), fn = fn, gr = gr, method = "L-BFGS-B",
  lower = rep(2, 25), upper = rep(4, 25)
)
```

```
## $par
##  [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3
##
## $value
## [1] 373
##
## $counts
## function gradient
##        2        2
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: NORM OF PROJECTED GRADIENT <= PGTOL"
```

36.4.3.2 非线性严格不等式约束

第一个例子，目标函数是非线性的，约束条件也是非线性的，非线性不等式约束不包含等号。

© 王建湘

$$
\begin{aligned}
&\min_x \quad (x_1+3x_2+x_3)^2+4(x_1-x_2)^2 \\
&s.t. \begin{cases} x_1+x_2+x_3=1 \\ 6x_2+4x_3-x_1^3>3 \\ x_1,x_2,x_3>0 \end{cases}
\end{aligned}
$$

```
# 目标函数
fn <- function(x) (x[1] + 3 * x[2] + x[3])^2 + 4 * (x[1] - x[2])^2
# 目标函数的梯度
gr <- function(x) {
  c(
    2 * (x[1] + 3 * x[2] + x[3]) + 8 * (x[1] - x[2]), # 对 x[1] 求偏导
    6 * (x[1] + 3 * x[2] + x[3]) - 8 * (x[1] - x[2]), # 对 x[2] 求偏导
    2 * (x[1] + 3 * x[2] + x[3]) # 对 x[3] 求偏导
  )
}
# 等式约束
heq <- function(x) {
  x[1] + x[2] + x[3] - 1
}
# 等式约束的雅可比矩阵
# 这里只有一个等式约束，所以雅可比矩阵行数为 1
heq.jac <- function(x) {
  matrix(c(1, 1, 1), ncol = 3, byrow = TRUE)
}
# 不等式约束
# 要求必须是严格不等式，不能带等号，方向是 x > 0
hin <- function(x) {
  c(6 * x[2] + 4 * x[3] - x[1]^3 - 3, x[1], x[2], x[3])
}
# 不等式约束的雅可比矩阵
# 其实是有 4 个不等式约束，3 个目标变量约束，雅可比矩阵行数是 4
hin.jac <- function(x) {
  matrix(c(
    -3 * x[1]^2, 6, 4,
    1, 0, 0,
    0, 1, 0,
    0, 0, 1
  ), ncol = 3, byrow = TRUE)
}
```

调用 **alabama** 包的求解器

```
set.seed(12)
# 初始值
p0 <- runif(3)
# 求目标函数的极小值
```

©黄湘云

```r
ans <- alabama::constrOptim.nl(
  par = p0,
  # 目标函数
  fn = fn,
  gr = gr,
  # 等式约束
  heq = heq,
  heq.jac = heq.jac,
  # 不等式约束
  hin = hin,
  hin.jac = hin.jac,
  # 不显示迭代过程
  control.outer = list(trace = FALSE)
)
ans
```

```
## $par
## [1] 7.390292e-04 4.497160e-12 9.992610e-01
##
## $value
## [1] 1.000002
##
## $counts
## function gradient
##     1230      163
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##           [,1]      [,2]      [,3]
## [1,] 120517098 120517087 120517091
## [2,] 120517087 120517115 120517095
## [3,] 120517091 120517095 120517091
##
## $outer.iterations
## [1] 13
##
## $lambda
## [1] 4.481599
##
## $sigma
```

© 王湘君


```
## [1] 120517089
##
## $barrier.value
## [1] 0.003472071
##
## $K
## [1] 4.269112e-08
```

ans 是 `constrOptim.nl()` 返回的一个 list，convergence = 0 表示迭代成功收敛，value 表示目标函数在迭代终止时的取直，par 表示满足约束条件，成功收敛的情况下，目标函数的参数值，counts 表示迭代过程中目标函数及其梯度计算的次数。

```r
# 不提供梯度函数，照样可以求解
ans <- alabama::constrOptim.nl(par = p0, fn = fn, heq = heq, hin = hin)
```

注意

等式和不等式约束的雅可比矩阵必须以 matrix 数据类型存储，而不能以 vector 类型存储。要注意和后面 ROI 包的调用形式区别。

实际上，可以用 ROI 调用 alabama 求解器的方式，这种方式可以简化目标函数梯度和约束条件的表示

```r
# 目标函数
fn <- function(x) (x[1] + 3 * x[2] + x[3])^2 + 4 * (x[1] - x[2])^2
# 目标函数的梯度
gr <- function(x) {
  c(
    2 * (x[1] + 3 * x[2] + x[3]) + 8 * (x[1] - x[2]),
    6 * (x[1] + 3 * x[2] + x[3]) - 8 * (x[1] - x[2]),
    2 * (x[1] + 3 * x[2] + x[3])
  )
}
heq <- function(x) {
  x[1] + x[2] + x[3]
}
heq.jac <- function(x) {
  c(1, 1, 1)
}
hin <- function(x) {
  6 * x[2] + 4 * x[3] - x[1]^3
}
hin.jac <- function(x) {
   c(-3 * x[1]^2, 6, 4)
}
```

通过 ROI 调用 alabama 求解器

```r
set.seed(2020)
# 初始值
```

© 黄湘云

```r
p0 <- runif(3)
# 定义目标规划
op <- OP(
  objective = F_objective(F = fn, n = 3L, G = gr), # 4 个目标变量
  constraints = F_constraint(
    F = list(heq = heq, hin = hin),
    dir = c("==", ">"),
    rhs = c(1, 3),
    # 等式和不等式约束的雅可比
    J = list(heq.jac = heq.jac, hin.jac = hin.jac)
  ),
  bounds = V_bound(ld = 0, ud = +Inf, nobj = 3L),
  maximum = FALSE # 求最小
)
nlp <- ROI_solve(op, solver = "alabama", start = p0)
nlp$solution
```

```
## [1] 1.674812e-06 9.994336e-08 9.999982e-01
```

```r
nlp$objval
```

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

36.4.3.3 非线性和箱式约束

与上面的例子不同，下面这个例子的不等式约束包含等号，还有箱式约束，优化问题来源于Ipopt 官网，提供的初始值为 $x_0 = (1, 5, 5, 1)$，最优解为 $x_\star = (1.00000000, 4.74299963, 3.82114998, 1.37940829)$。优化问题的具体内容如下：

$$
\begin{array}{ll}
\min_x & x_1x_4(x_1+x_2+x_3)+x_3 \\
s.t. & \begin{cases} x_1^2+x_2^2+x_3^2+x_4^2 = 40 \\ x_1x_2x_3x_4 \geq 25 \\ 1 \leq x_1, x_2, x_3, x_4 \leq 5 \end{cases}
\end{array}
$$

考虑用 ROI 调 nloptr 实现，看结果是否和例子一致，nloptr 支持不等式约束包含等号，支持箱式约束。

```r
# 一个 4 维的目标函数
fn <- function(x) {
  x[1] * x[4] * (x[1] + x[2] + x[3]) + x[3]
}
# 目标函数的梯度
gr <- function(x) {
  c(
    x[4] * (2 * x[1] + x[2] + x[3]), x[1] * x[4],
    x[1] * x[4] + 1, x[1] * (x[1] + x[2] + x[3])
  )
}
```

© 王燕清


```r
# 等式约束
heq <- function(x) {
  sum(x^2)
}
# 等式约束的雅可比
heq.jac <- function(x) {
  2 * c(x[1], x[2], x[3], x[4])
}
# 不等式约束
hin <- function(x) {
  prod(x)
}
# 不等式约束的雅可比
hin.jac <- function(x) {
  c(prod(x[-1]), prod(x[-2]), prod(x[-3]), prod(x[-4]))
}
# 定义目标规划
op <- OP(
  objective = F_objective(F = fn, n = 4L, G = gr), # 4 个目标变量
  constraints = F_constraint(
    F = list(heq = heq, hin = hin),
    dir = c("==", ">="),
    rhs = c(40, 25),
    # 等式和不等式约束的雅可比
    J = list(heq.jac = heq.jac, hin.jac = hin.jac)
  ),
  bounds = V_bound(ld = 1, ud = 5, nobj = 4L),
  maximum = FALSE # 求最小
)
```

```r
# 目标函数初始值
fn(c(1, 5, 5, 1))
```

```
## [1] 16
```

```r
# 目标函数最优值
fn(c(1.00000000, 4.74299963, 3.82114998, 1.37940829))
```

```
## [1] 17.01402
```

求解一般的非线性约束问题，求解器 nloptr.mma / nloptr.cobyla 仅支持非线性不等式约束，不支持等式约束，而 nlminb 只支持等式约束，因此，下面分别调用 nloptr.auglag、nloptr.slsqp 和 nloptr.isres 来求解上述优化问题。

```r
nlp <- ROI_solve(op, solver = "nloptr.auglag", start = c(1, 5, 5, 1))
nlp$solution
```

```
## [1] 1.000000 4.743174 3.820922 1.379440
```

统计学习 ©

```
nlp$objval
```

```
## [1] 17.01402
```

```
nlp <- ROI_solve(op, solver = "nloptr.slsqp", start = c(1, 5, 5, 1))
nlp$solution
```

```
## [1] 1.000000 4.742996 3.821155 1.379408
```

```
nlp$objval
```

```
## [1] 17.01402
```

```
nlp <- ROI_solve(op, solver = "nloptr.isres", start = c(1, 5, 5, 1))
nlp$solution
```

```
## [1] 1.156667 4.649759 3.950369 1.198390
```

```
nlp$objval
```

```
## [1] 17.47464
```

可以看出，nloptr 提供的优化能力可以覆盖Ipopt 求解器，推荐使用 nloptr.slsqp 求解器。

36.4.3.4 非线性混合整数约束

$$
\begin{aligned}
\max_x \quad & 1.5(x_1-\sin(x_1-x_2))^2+0.5x_2^2+x_3^2-x_1x_2-2x_1+x_2x_3 \\
s.t. & \begin{cases} -20<x_1<20 \\ -20<x_2<20 \\ -10<x_3<10 \\ x_1,x_2\in\mathbb{R}, \quad x_3\in\mathbb{Z} \end{cases}
\end{aligned}
$$

```
fn <- function(x) {
  1.5 * (x[1] - sin(x[1] - x[2]))^2 + 0.5 * x[2]^2 + x[3]^2
  -x[1] * x[2] - 2 * x[1] + x[2] * x[3]
}
gr <- function(x) {
  c(
    3 * (x[1] - sin(x[1] - x[2])) * (1 - cos(x[1] - x[2])) - x[2] - 2,
    3 * (x[1] - sin(x[1] - x[2])) * cos(x[1] - x[2]) - x[2] - x[1] + x[3],
    2 * x[3] + x[2]
  )
}
```

目前 ROI 还解不了

```
# 初始值
p0 <- c(2.1, 5.1, 5)
# 定义目标规划
op <- OP(
  objective = F_objective(F = fn, n = 3L, G = gr), # 3 个目标变量
```

© 基础统计


```
  types = c("C", "C", "I"), # 目标变量的类型
  bounds = V_bound(lb = c(-20, -20, -10), ub = c(20, 20, 10), nobj = 3L),
  maximum = FALSE # 求最小
)
nlp <- ROI_solve(op, solver = "auto", start = p0)
nlp$solution
```

目标函数在 $(4.49712, 9.147501, -4)$ 取得最小值 -86.72165

```
fn(x = c(4.49712, 9.147501, -4))
```

```
## [1] -86.72165
```

36.4.3.5 含复杂目标函数

下面这个目标函数比较复杂，约束条件也是非线性的

$$\max_x \quad \frac{(\sin(2\pi x_1))^3 \sin(2\pi x_2)}{x_1^3(x_1+x_2)}$$
$$s.t. \begin{cases} x_1^2 - x_2 + 1 \le 0 \\ 1 - x_1 + (x_2 - 4)^2 \ge 0 \\ 0 \le x_1, x_2 \le 10 \end{cases}$$

```
# 目标函数
fn <- function(x) (sin(2*pi*x[1]))^3 * sin(2*pi*x[2])/(x[1]^3*(x[1] + x[2]))
# 目标函数的梯度
gr <- function(x) {
  numDeriv::grad(fn, c(x[1], x[2]))
}

hin <- function(x) {
  c(
    x[1]^2 - x[2] + 1,
    1 - x[1] + (x[2] - 4)^2
  )
}

hin.jac <- function(x) {
  matrix(c(
    2 * x[1], -1,
    -1, 2 * x[2]
  ),
  ncol = 2, byrow = TRUE
  )
}

# 初始值
```

© 基础统计

```
p0 <- c(2, 5)
# 定义目标规划
op <- OP(
  objective = F_objective(F = fn, n = 2L, G = gr), # 2 个目标变量
  constraints = F_constraint(
    F = list(hin = hin),
    dir = c("<=", "<="),
    rhs = c(0, 0),
    # 不等式约束的雅可比
    J = list(hin.jac = hin.jac)
  ),
  bounds = V_bound(ld = 0, ud = 10, nobj = 2L),
  maximum = TRUE # 求最大
)
nlp <- ROI_solve(op, solver = "nloptr.isres", start = p0)
nlp$solution
```

```
## [1] 1.227972 4.245369
```

```
nlp$objval
```

```
## [1] 0.09582504
```

下面再给一个来自 Octave 优化文档 的示例，该优化问题包含多个非线性的等式约束。

$$
\begin{array}{ll}
\min_x & \mathrm{e}^{\prod_{i=1}^5 x_i} - \frac{1}{2}(x_1^3 + x_2^3 + 1)^2 \\
s.t. & \begin{cases} \sum_{i=1}^5 x_i^2 - 10 = 0 \\ x_2 x_3 - 5 x_4 x_5 = 0 \\ x_1^3 + x_2^3 + 1 = 0 \end{cases}
\end{array}
$$

```
# 一个 5 维的目标函数
fn <- function(x) {
  exp(prod(x)) - 0.5 * (x[1]^3 + x[2]^3 + 1)^2
}
# 目标函数的梯度
gr <- function(x) {
  c(
    exp(prod(x))*prod(x[-1]) - 3*(x[1]^3 + x[2]^3 + 1)*x[1]^2,
    exp(prod(x))*prod(x[-2]) - 3*(x[1]^3 + x[2]^3 + 1)*x[2]^2,
    exp(prod(x))*prod(x[-3]),
    exp(prod(x))*prod(x[-4]),
    exp(prod(x))*prod(x[-5])
  )
}
# 等式约束
heq <- function(x) {
  c(
```

©清华大学出版社

```
    sum(x^2) - 10,
    x[2] * x[3] - 5 * x[4] * x[5],
    x[1]^3 + x[2]^3 + 1
  )
}
# 等式约束的雅可比
heq.jac <- function(x) {
  matrix(c(2 * x[1], 2 * x[2], 2 * x[3], 2 * x[4], 2 * x[5],
    0, x[3], x[2], -5 * x[5], -5 * x[4],
    3 * x[1]^2, 3 * x[2]^2, 0, 0, 0),
    ncol = 5, byrow = TRUE
  )
}
```

```
# 定义目标规划
op <- OP(
  objective = F_objective(F = fn, n = 5L, G = gr), # 5 个目标变量
  constraints = F_constraint(
    F = list(heq = heq),
    dir = "==",
    rhs = 0,
    # 等式的雅可比
    J = list(heq.jac = heq.jac)
  ),
  bounds = V_bound(ld = -Inf, ud = Inf, nobj = 5L),
  maximum = FALSE # 求最小
)
```

调用 SQP（序列二次规划）求解器

```
nlp <- ROI_solve(op, solver = "nloptr.slsqp", start = c(-1.8, 1.7, 1.9, -0.8,-0.8))
nlp$solution
```

```
## [1] -1.7171435  1.5957096  1.8272458 -0.7636431 -0.7636431
```

计算结果和 Octave 的示例一致。

36.4.3.6 含复杂约束条件

$$
\begin{aligned}
\min_x \quad & \exp(\sin(50\cdot x)) + \sin(60\cdot \exp(y)) + \sin(70\cdot \sin(x)) \\
& + \sin(\sin(80\cdot y)) - \sin(10\cdot (x+y)) + \frac{(x^2+y^2)^{\sin(y)}}{4} \\
s.t. \quad & \begin{cases} x - \left((\cos(y))^x - x\right)^y = 0 \\ -50 \le x_1, x_2 \le 50 \end{cases}
\end{aligned}
$$

Lingo 代码如下：

```
Min = @exp(@sin(50 * x)) + @sin(60 * @exp(y)) + @sin(70 * @sin(x))
```

```
    + @sin(@sin(80 * y)) - @sin(10 * (x + y)) + (x^2 + y^2)^@sin(y) / 4;

x - (( @cos(y) )^x - x)^y = 0;

@bnd(-50, x, 50);
@bnd(-50, y, 50);
```

启用全局优化求解器，求解 14 分钟，在 $(0.08256372, 24.56510)$ 取得极小值 -2.863497。不启用全局优化器就没法解，Lingo 会报错，找不到最优解，勉强找到一个可行解 $(0.06082750, 44.12793)$，目标值为 -1.29816。

```r
fn <- function(x) {
  exp(sin(50 * x[1])) + sin(60 * exp(x[2])) +
    sin(70 * sin(x[1])) + sin(sin(80 * x[2])) -
    sin(10 * (x[1] + x[2])) + (x[1]^2 + x[2]^2)^(sin(x[2])) / 4
}
gr <- function(x){
  numDeriv::grad(fn, c(x[1], x[2]))
}
heq <- function(x){
  x[1] - ( (cos(x[2]))^x[1] - x[1] )^x[2]
}
heq.jac <- function(x){
  numDeriv::grad(heq, c(x[1], x[2]))
}

fn(x = c(0.06082750, 44.12793))
```

```
## [1] -1.29816
```

```r
fn(x = c(1, 0))
```

```
## [1] 1.966877
```

```r
heq(x = c(0.06082750, 44.12793))
```

```
## [1] 1.923673e-08
```

```r
heq(x = c(1, 0))
```

```
## [1] 0
```

```r
# 定义目标规划
op <- OP(
  objective = F_objective(F = fn, n = 2L, G = gr), # 2 个目标变量
  constraints = F_constraint(
    F = list(heq = heq),
    dir = "==",
    rhs = 0,
    J = list(heq.jac = heq.jac)
  ),
```

© 草稿


```
  bounds = V_bound(ld = -50, ud = 50, nobj = 2L),
  maximum = FALSE # 求最小
)
```

nloptr.auglag 无法求解此优化问题

```
nlp <- ROI_solve(op, solver = "nloptr.auglag", start = c(1, 0))
nlp$solution
```

调 nloptr.isres 求解器，每次执行都会得到不同的局部最优解

```
nlp <- ROI_solve(op, solver = "nloptr.isres", start = c(1, 0))
nlp$solution
```

```
## [1] -9.58856 22.92133
```

```
nlp$objval
```

```
## [1] -2.912641
```

比如下面三组

```
fn(x = c(40.95941, 41.52914))
```

```
## [1] -1.025926
```

```
heq(x = c(40.95941, 41.52914))
```

```
## [1] NaN
```

```
fn(x = c(-21.88091, 28.96994))
```

```
## [1] -1.467513
```

```
heq(x = c(-21.88091, 28.96994))
```

```
## [1] NaN
```

```
fn(x = c(-49.921967437, 4.8499336803))
```

```
## [1] -3.466596
```

```
heq(x = c(-49.921967437, 4.8499336803))
```

```
## [1] -8.515447e+208
```

36.5 非线性方程

36.5.1 一元非线性方程

牛顿-拉弗森方法

```
library(rootSolve)
```

©黄湘云

36.5.2 非线性方程组

```
library(BB)
```

二项混合泊松分布的参数最大似然估计

```
poissmix.loglik <- function(p, y) {
  # Log-likelihood for a binary Poisson mixture distribution
  i <- 0:(length(y) - 1)

  loglik <- y * log(p[1] * exp(-p[2]) * p[2]^i / exp(lgamma(i + 1)) +
    (1 - p[1]) * exp(-p[3]) * p[3]^i / exp(lgamma(i + 1)))

  sum(loglik)
}
# Data from Hasselblad (JASA 1969)
# 介绍实际应用场景
poissmix.dat <- data.frame(death = 0:9,
                           freq = c(162, 267, 271, 185, 111, 61, 27, 8, 3, 1))
lo <- c(0, 0, 0) # lower limits for parameters
hi <- c(1, Inf, Inf) # upper limits for parameters
p0 <- runif(3, c(0.2, 1, 1), c(0.8, 5, 8))
# a randomly generated vector of length 3
y <- c(162, 267, 271, 185, 111, 61, 27, 8, 3, 1)

ans1 <- spg(
  par = p0, fn = poissmix.loglik, y = y, lower = lo, upper = hi,
  control = list(maximize = TRUE, trace = FALSE)
)
ans2 <- BBoptim(
  par = p0, fn = poissmix.loglik, y = y,
  lower = lo, upper = hi, control = list(maximize = TRUE)
)
```

```
## iter:  0  f-value:  -2136.431  pgrad:  236.9752
## iter:  10  f-value:  -1995.89  pgrad:  2.961353
## iter:  20  f-value:  -2041.139  pgrad:  2.57697
## iter:  30  f-value:  -1989.974  pgrad:  0.4742151
## iter:  40  f-value:  -1989.949  pgrad:  0.2614752
## iter:  50  f-value:  -1989.946  pgrad:  0.01959506
## iter:  60  f-value:  -1989.946  pgrad:  0.002494289
##   Successful convergence.
```

```
ans2
```

```
## $par
## [1] 0.3598829 1.2560906 2.6634011
##
```

©黄湘云


```
## $value
## [1] -1989.946
##
## $gradient
## [1] 0.0001000444
##
## $fn.reduction
## [1] -146.4848
##
## $iter
## [1] 68
##
## $feval
## [1] 170
##
## $convergence
## [1] 0
##
## $message
## [1] "Successful convergence"
##
## $cpar
## method      M
##      2     50
```

计算最大似然处的黑塞矩阵以及参数的标准差

```
hess <- numDeriv::hessian(x = ans2$par, func = poissmix.loglik, y = y)
# Note that we have to supplied data vector 'y'
hess
```

```
##           [,1]      [,2]      [,3]
## [1,] -907.1105  270.2287  341.2543
## [2,]  270.2287 -113.4794  -61.6819
## [3,]  341.2543  -61.6819 -192.7822
```

```
se <- sqrt(diag(solve(-hess)))
se
```

```
## [1] 0.1946836 0.3500308 0.2504769
```

从不同初始值出发尝试寻找全局最大值，实际找的是一系列局部最大值

```
# 3 randomly generated starting values
p0 <- matrix(runif(30, c(0.2, 1, 1), c(0.8, 8, 8)), 10, 3, byrow = TRUE)
ans <- multiStart(
  par = p0, fn = poissmix.loglik, action = "optimize",
  y = y, lower = lo, upper = hi, control = list(maximize = TRUE)
)
```

©黄湘云

```
## Parameter set :  1 ...
## iter:  0  f-value:  -2076.377  pgrad:  266.5811
## iter:  10  f-value:  -1991.788  pgrad:  3.394882
## iter:  20  f-value:  -1990.932  pgrad:  8.266675
## iter:  30  f-value:  -1989.958  pgrad:  0.2441652
## iter:  40  f-value:  -1989.946  pgrad:  0.001411991
##   Successful convergence.
## Parameter set :  2 ...
## iter:  0  f-value:  -3999.343  pgrad:  6.350898
## iter:  10  f-value:  -2015.457  pgrad:  2.400803
##   Successful convergence.
## Parameter set :  3 ...
## iter:  0  f-value:  -2526.385  pgrad:  3.959104
## iter:  10  f-value:  -1997.785  pgrad:  4.651176
## iter:  20  f-value:  -2041.124  pgrad:  130.6335
## iter:  30  f-value:  -1989.979  pgrad:  0.4133676
## iter:  40  f-value:  -1989.953  pgrad:  0.2001525
## iter:  50  f-value:  -1989.946  pgrad:  0.02953584
##   Successful convergence.
## Parameter set :  4 ...
## iter:  0  f-value:  -4036.966  pgrad:  7.725057
## iter:  10  f-value:  -1993.146  pgrad:  3.356279
## iter:  20  f-value:  -1992.445  pgrad:  3.162911
## iter:  30  f-value:  -1999.964  pgrad:  3.124857
## iter:  40  f-value:  -1990.201  pgrad:  0.9762675
## iter:  50  f-value:  -1989.962  pgrad:  0.3950169
## iter:  60  f-value:  -1989.946  pgrad:  0.0507498
## iter:  70  f-value:  -1989.946  pgrad:  0.0001978151
##   Successful convergence.
## Parameter set :  5 ...
## iter:  0  f-value:  -2048.809  pgrad:  2.862445
## iter:  10  f-value:  -1992.344  pgrad:  2.68979
## iter:  20  f-value:  -1990.604  pgrad:  7.2791
## iter:  30  f-value:  -1989.978  pgrad:  0.3772993
## iter:  40  f-value:  -1989.946  pgrad:  0.004172307
## iter:  50  f-value:  -1989.946  pgrad:  0.004260983
##   Successful convergence.
## Parameter set :  6 ...
## iter:  0  f-value:  -4777.283  pgrad:  7.596832
## iter:  10  f-value:  -1991.838  pgrad:  11.02078
## iter:  20  f-value:  -1990.272  pgrad:  0.5307333
## iter:  30  f-value:  -1989.963  pgrad:  2.230793
## iter:  40  f-value:  -1989.946  pgrad:  0.008421921
## iter:  50  f-value:  -1989.946  pgrad:  0.0001841727
##   Successful convergence.
```

© 基础统计


```
## Parameter set :  7 ...
## iter:  0  f-value:  -2019.928  pgrad:  3.485709
## iter:  10  f-value:  -1990.626  pgrad:  1.833378
## iter:  20  f-value:  -1989.999  pgrad:  1.098717
## iter:  30  f-value:  -1989.947  pgrad:  0.3092782
## iter:  40  f-value:  -1989.946  pgrad:  0.007039489
##   Successful convergence.
## Parameter set :  8 ...
## iter:  0  f-value:  -2764.625  pgrad:  4.891128
## iter:  10  f-value:  -2001.398  pgrad:  2.273737e-06
##   Successful convergence.
## Parameter set :  9 ...
## iter:  0  f-value:  -2167.165  pgrad:  195.5499
## iter:  10  f-value:  -2001.54  pgrad:  2.194864
## iter:  20  f-value:  -2000.825  pgrad:  0.6559458
## iter:  30  f-value:  -1992.777  pgrad:  7.064828
## iter:  40  f-value:  -1991.747  pgrad:  3.357115
## iter:  50  f-value:  -1989.983  pgrad:  2.772795
## iter:  60  f-value:  -1989.946  pgrad:  0.03392643
## iter:  70  f-value:  -1989.946  pgrad:  0.0003728928
##   Successful convergence.
## Parameter set :  10 ...
## iter:  0  f-value:  -2100.94  pgrad:  317.5313
## iter:  10  f-value:  -1991.327  pgrad:  2.7843
## iter:  20  f-value:  -1990.415  pgrad:  1.435174
## iter:  30  f-value:  -1990.046  pgrad:  3.248585
## iter:  40  f-value:  -1989.946  pgrad:  0.06813025
## iter:  50  f-value:  -1989.946  pgrad:  0.001450644
##   Successful convergence.
```

```r
# selecting only converged solutions
pmat <- round(cbind(ans$fvalue[ans$conv], ans$par[ans$conv, ]), 4)
dimnames(pmat) <- list(NULL, c("fvalue", "parameter 1", "parameter 2", "parameter 3"))
pmat[!duplicated(pmat), ]
```

```
##         fvalue parameter 1 parameter 2 parameter 3
## [1,] -1989.946      0.6401      2.6634      1.2561
## [2,] -1997.263      0.4922      2.4559      1.8567
## [3,] -1989.946      0.3599      1.2561      2.6634
## [4,] -2000.039      0.7931      2.0681      2.4778
## [5,] -1989.946      0.3599      1.2560      2.6634
```

用一个具体的参数估计问题，求极大似然点，混合正态分布隐函数方程组求解非线性方程组 [Varadhan and Gilbert, 2009]

© 王湘君

36.6 多目标规划

多目标规划的基本想法是将多目标问题转化为单目标问题，常见方法有理想点法、线性加权法、非劣解集法、极大极小法。理想点法是先在给定约束条件下分别求解单个目标的最优值，构造新的单目标函数。线性加权法是给每个目标函数赋予权重系数，各个权重系数之和等于 1。非劣解集法是先求解其中一个单目标函数的最优值，然后将其设为等式约束，将其最优值从最小值开始递增，然后求解另一个目标函数的最小值。极大极小法是采用标准的简面体爬山法和通用全局优化法求解多目标优化问题。

R 环境中，GPareto 主要用来求解多目标规划问题。试验设计和过程优化与 R 语言 的 约束优化 章节，优化和解方程。另外，《Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques》[Deb, 2005] 多目标优化方法

$$
\begin{array}{l}
\min_x \left\{ \begin{array}{l} f_1(x) = 0.5x_1 + 0.6x_2 + 0.7\exp(\frac{x_1+x_3}{10}) \\ f_2(x) = (x_1 - 2x_2)^2 + (2x_2 - 3x_3)^2 + (5x_3 - x_1)^2 \end{array} \right. \\
s.t. \quad x_1 \in [10, 80], x_2 \in [20, 90], x_3 \in [15, 100]
\end{array}
$$

```
library(DiceKriging)
library(emoa)
library(GPareto)
library(DiceDesign)
```

```
library(Ternary)
TernaryPlot(
  atip = "Top", btip = "Bottom", ctip = "Right",
  axis.col = "red", col = rgb(0.8, 0.8, 0.8)
)
HorizontalGrid(grid.lines = 2, grid.col = "blue", grid.lty = 1)
```

36.7 经典优化问题

旅行商问题、背包问题、指派问题、选址问题、网络流量问题

规划快递员送餐的路线：从快递员出发地到各个取餐地，再到顾客家里，如何规划路线使得每个顾客下单到拿到餐的时间间隔小于 50 分钟，完成送餐，快递员的总时间最少？

36.8 回归与优化

简单线性回归

nlsr

是否能给大家提供一些思路？

Lasso [Tibshirani, 1996]

Least Angle Regression [Efron et al., 2004]

© 基础统计

为了解决 Lasso 的有偏估计问题，自适应 Lasso、松弛 Lasso SCAD (Smoothly Clipped Absolute Deviation)[Kim et al., 2008] MCP (Minimax Concave Penalty)[Zhang, 2010]

由于缺少高效的求解算法，Lasso 在高维小样本特征选择研究中没有广泛流行，最小角回归 (Least Angle Regression, LAR) 算法 [Efron et al., 2004] 的出现有力促进了 Lasso 在高维小样本数据中的应用

bestsubset 最优子集回归

经典的普通最小二乘、广义最小二乘、岭回归、逐步回归、Lasso 回归、最优子集回归都可转化为优化问题，一般形式如下

$$\underbrace{\hat{\theta}_{\lambda_n}}_{\text{待估参数}} \in \arg\min_{\theta\in\Omega}\left\{\underbrace{\mathcal{L}(\theta; Z_1^n)}_{\text{损失函数}} + \lambda_n \underbrace{\mathcal{R}(\theta)}_{\text{正则化项}}\right\}.$$

下面尝试以 nloptr 包的优化器来展示求解过程，并与 Base R、**glmnet** 和 **MASS** 实现的回归模型比较。

$$\arg\min_{\beta,\lambda} \frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1$$

其中，$X \in \mathbb{R}^{m\times n}$，$y \in \mathbb{R}^m$，$\beta \in \mathbb{R}^n$，$0 < \lambda \in \mathbb{R}$

$$y = X\beta + \epsilon$$

$\hat{\beta} = (X^\top X)^{-1}X^\top y, \quad \hat{Y} = X(X^\top X)^{-1}X^\top y$

```
set.seed(123)
n <- 200
p <- 50
x <- matrix(rnorm(n * p), n)
y <- rnorm(n)
lm(y ~ x + 0)
# y 的估计
# 教科书版
fit_base = function(x, y) {
  x %*% solve(t(x) %*% x) %*% t(x) %*% y
}
# 先向量计算，然后矩阵计算
fit_vector = function(x, y) {
  x %*% (solve(t(x) %*% x) %*% (t(x) %*% y))
}
# X'X 是对称的，防止求逆
fit_inv = function(x, y) {
  x %*% solve(crossprod(x), crossprod(x, y))
}
```

QR 分解 $X_{n\times p} = Q_{n\times p}R_{p\times p}$，$n > p$，$Q^\top Q = I$，$R$ 是上三角矩阵，$\hat{Y} = X(X^\top X)^{-1}X^\top y = QQ^\top y$

```
fit_qr <- function(x, y) {
  decomp <- qr(x)
```

© 草稿

```
  qr.qy(decomp, qr.qty(decomp, y))
}
lm.fit(x, y)
```

若 $A = X^\top X$ 是正定矩阵，则 $A = LL^\top$，L 是下三角矩阵

```
fit_chol <- function(x, y) {
  decomp <- chol(crossprod(x))
  lxy <- backsolve(decomp, crossprod(x, y), transpose = TRUE)
  b <- backsolve(decomp, lxy)
  x %*% b
}
```

```
## Using C/C++
system.time(RcppEigen::fastLmPure(x, y, method = 1)) ## QR
system.time(RcppEigen::fastLmPure(x, y, method = 2)) ## Cholesky
system.time(RcppArmadillo::fastLmPure(x, y, method = 1)) ## QR
system.time(RcppArmadillo::fastLmPure(x, y, method = 2)) ## Cholesky
```

36.9 对数似然

随机变量 X 服从参数为 $\lambda > 0$ 的指数分布，密度函数 $p(x)$ 为

$$p(x) = \begin{cases} \lambda \mathrm{e}^{-\lambda x}, x \geq 0 \\ 0, \quad x < 0 \end{cases}$$

其中，$\lambda > 0$，下面给定一系列模拟样本观察值 $x_1, x_2, \cdots, x_n$，估计参数 λ。对数似然函数 $\ell(\lambda) = \log \prod_{i=1}^n f(x_i) = n \log \lambda - \lambda \sum_{i=1}^n x_i$。解此方程即可得到 λ 的极大似然估计 $\lambda_{mle} = \frac{1}{\bar{X}}$，极大值 $\ell(\lambda_{mle}) = -n(1 + \log \bar{X})$。

根据上述样本，计算样本均值 $(\mu - 1.5 * \sigma/\sqrt{n}, \mu + 1.5 * \sigma/\sqrt{n})$ 和方差 $(0.8\sigma, 1.5\sigma)$。已知正态分布 $f(x) = \frac{1}{\sqrt{2\pi}\sigma} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$ 的对数似然形式 $\ell(\mu, \sigma^2) = \log \prod_{i=1}^n f(x_i) = \sum_{i=1}^n \log f(x_i)$。正态分布的密度函数的对数可用 `dnorm(..., log = TRUE)` 计算。

生成服从指数分布的样本，计算样本的均值和方差，依据均值和方差构造区间，然后将区间网格化，在此网格上绘制正态分布的对数似然函数。绕那么大一个圈子，其实就是绘制正态分布的对数似然函数。

```
set.seed(2021)
n <- 20 # 随机数的个数
x <- rexp(n, rate = 5) # 服从指数分布的随机数
m <- 40 # 网格数

mu <- seq(
  mean(x) - 1.5 * sd(x) / sqrt(n),
  mean(x) + 1.5 * sd(x) / sqrt(n),
  length.out = m
)
```

© 黄湘云


```
sigma <- seq(0.8 * sd(x), 1.5 * sd(x), length.out = m)
df <- expand.grid(x = mu, y = sigma)
# 正态分布的对数似然
loglik <- function(b, x0) -sum(dnorm(x0, b[1], b[2], log = TRUE))

df$fnxy = apply(df, 1, loglik, x0 = x)

wireframe(
  data = df, fnxy ~ x * y,
  shade = TRUE, drape = FALSE,
  xlab = expression(mu),
  ylab = expression(sigma),
  zlab = list(expression(-loglik(mu, sigma)), rot = 90),
  scales = list(arrows = FALSE, col = "black"),
  par.settings = list(axis.line = list(col = "transparent")),
  screen = list(z = 120, x = -70, y = 0)
)
```

图 36.15: 正态分布参数的负对数似然函数

© 统计学习与机器学习

36.10 微分方程

ode45 求解偏微分方程

pracma 实现了 ode23, ode23s, ode45 等几个自适应的 Runge-Kutta 求解器，**deSolve** 包求解 ODE（常微分方程），DAE（微分代数方程），DDE（延迟微分方程，包含刚性和非刚性方程）和 PDE（偏微分方程），**bvpSolve** 包求解 DAE/ODE 方程的边值问题。**ReacTran** [Soetaert and Meysman, 2012] 可将偏微分方程转为常微分方程组，解决反应运输问题，在笛卡尔、极坐标、圆柱形和球形网格上离散偏微分方程。sundials 提供一系列非线性方程、常微分方程、微分代数方程求解器，Satyaprakash Nayak 开发了相应的 **sundialr** 包。

36.10.1 常微分方程

洛伦兹系统是一个常微分方程组，系统参数的默认值为 $(\sigma = 10, \rho = 28, \beta = 8/3)$，初值为 $(-13, -14, 47)$。

$$\begin{cases} \frac{\partial x}{\partial t} = \sigma(y - x) \\ \frac{\partial y}{\partial t} = x(\rho - z) - y \\ \frac{\partial x}{\partial t} = xy - \beta z \end{cases}$$

```
library(deSolve)
# 参数
pars <- c(a = -8 / 3, b = -10, c = 28)
# 初值
state <- c(X = 1, Y = 1, Z = 1)
# 时间间隔
times <- seq(0, 100, by = 0.01)
# 定义方程组
lorenz_fun <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {
    dX <- a * X + Y * Z
    dY <- b * (Y - Z)
    dZ <- -X * Y + c * Y - Z
    list(c(dX, dY, dZ))
  })
}
out <- ode(
  y = state, times = times,
  func = lorenz_fun, parms = pars
)
```

调用 **scatterplot3d** 绘制三维曲线图，如图36.16 所示

```
library(scatterplot3d)

scatterplot3d(
  x = out[, "X"], y = out[, "Y"], z = out[, "Z"],
```

© 王建湘


```
  col.axis = "black", type = "l", color = "gray",
  xlab = expression(x), ylab = expression(y), zlab = expression(z),
  col.grid = "gray", main = "Lorenz"
)
```

图 36.16: 洛伦兹曲线

36.10.2 偏微分方程

ReacTran 的几个关键函数介绍

一维热传导方程

$$\left\{ \frac{\partial y}{\partial t} = D\frac{\partial^2 y}{\partial x^2} \right.$$

参数 $D = 0.01$，边界条件 $y_{t,x=0} = 0, y_{t,x=1} = 1$，初始条件 $y_{t=0,x} = \sin(\pi x)$。

```
library(ReacTran)

N <- 100
xgrid <- setup.grid.1D(x.up = 0, x.down = 1, N = N)
```

© 科学出版社

```
x <- xgrid$x.mid
D.coeff <- 0.01
Diffusion <- function(t, Y, parms) {
  tran <- tran.1D(
    C = Y, C.up = 0, C.down = 1,
    D = D.coeff, dx = xgrid
  )
  list(
    dY = tran$dC,
    flux.up = tran$flux.up,
    flux.down = tran$flux.down
  )
}
yini <- sin(pi * x)
times <- seq(from = 0, to = 5, by = 0.01)
out <- ode.1D(
  y = yini, times = times, func = Diffusion,
  parms = NULL, dimens = N
)
```

```
image(out,
  grid = xgrid$x.mid, xlab = "times",
  ylab = "Distance", main = "PDE", add.contour = TRUE
)
```

二维拉普拉斯方程

$$\left\{ \frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial y^2} = 0 \right.$$

边界条件

$$\begin{cases} u_{x=0,y} = u_{x=1,y} = 0 \\ \frac{\partial u_{x,y=0}}{\partial y} = 0 \\ \frac{\partial u_{x,y=1}}{\partial y} = \pi \sinh(\pi) \sin(\pi x) \end{cases}$$

它有解析解

$$u(x, y) = \sin(\pi x) \cosh(\pi y)$$

其中 $x \in [0, 1], y \in [0, 1]$

```
fn <- function(x, y) {
  sin(pi * x) * cosh(pi * y)
}
x <- seq(0, 1, length.out = 101)
```

© 黄湘云

图 36.17: 一维热传导方程的数值解热力图

统计出版社 ©

```
y <- seq(0, 1, length.out = 101)
z <- outer(x, y, fn)
```

```
image(z, col = terrain.colors(20))
contour(z, method = "flattest", add = TRUE, lty = 1)
```

图 36.18: 解析解的二维图像

```
persp(z,
  theta = 30, phi = 20,
  r = 50, d = 0.1, expand = 0.5, ltheta = 90, lphi = 180,
  shade = 0.1, ticktype = "detailed", nticks = 5, box = TRUE,
  col = drapecol(z, col = terrain.colors(20)),
  border = "transparent",
  xlab = "X", ylab = "Y", zlab = "Z",
  main = ""
)
```

求解 PDE

```
dx <- 0.2
xgrid <- setup.grid.1D(-100, 100, dx.1 = dx)
x <- xgrid$x.mid
```

© 黄湘石

图 36.19: 解析解的三维透视图像

©李东风

```
N <- xgrid$N

uini <- exp(-0.05 * x^2)
vini <- rep(0, N)
yini <- c(uini, vini)
times <- seq(from = 0, to = 50, by = 1)

wave <- function(t, y, parms) {
  u1 <- y[1:N]
  u2 <- y[-(1:N)]
  du1 <- u2
  du2 <- tran.1D(C = u1, C.up = 0, C.down = 0, D = 1, dx = xgrid)$dC
  return(list(c(du1, du2)))
}

out <- ode.1D(
  func = wave, y = yini, times = times, parms = NULL,
  nspec = 2, method = "ode45", dimens = N, names = c("u", "v")
)
```

36.10.3 延迟微分方程

```
library(PBSddesolve)    # DAE 延迟微分方程
```

PBSddesolve [Couture-Beil et al., 2019] **PBSmodelling PBSmapping**

nlmeODE 通过微分方程整合用于混合效应模型的 odesolve 和 nlme 包。

36.10.4 随机微分方程

Sim.DiffProc

随机微分方程入门：基于 R 语言的模拟和推断

```
library(Sim.DiffProc)
```

种群 ODE 建模，

nlmixr 借助 RxODE 求解基于常微分方程的非线性混合效应模型

36.11 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
```

©黄湘云

```
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] Deriv_4.1.3               quadprog_1.5-8
##  [3] kableExtra_1.3.4          tibble_3.1.6
##  [5] Sim.DiffProc_4.8          nlmeODE_1.1
##  [7] nlme_3.1-157              PBSddesolve_1.12.6
##  [9] ReacTran_1.4.3.1          shape_1.4.6
## [11] scatterplot3d_0.3-41      deSolve_1.31
## [13] BB_2019.10-1              rootSolve_1.8.2.3
## [15] kernlab_0.9-29            lattice_0.20-45
## [17] ROI.plugin.scs_1.1-1      ROI.plugin.quadprog_1.0-0
## [19] ROI.plugin.lpsolve_1.0-1  ROI.plugin.nloptr_1.0-0
## [21] ROI.plugin.alabama_1.0-0  ROI_1.0-0
## [23] lpSolve_5.6.15
##
## loaded via a namespace (and not attached):
##  [1] svglite_2.1.0           sysfonts_0.8.8          digest_0.6.29
##  [4] utf8_1.2.2              slam_0.1-50             R6_2.5.1
##  [7] alabama_2015.3-1        evaluate_0.15           httr_1.4.2
## [10] pillar_1.7.0            rlang_1.0.2             curl_4.3.2
## [13] rstudioapi_0.13         nloptr_2.0.0            rmarkdown_2.13
## [16] webshot_0.5.2           stringr_1.4.0           munsell_0.5.0
## [19] compiler_4.1.3          numDeriv_2016.8-1.1     xfun_0.30
## [22] systemfonts_1.0.4       pkgconfig_2.0.3         htmltools_0.5.2
## [25] bookdown_0.25           viridisLite_0.4.0       fansi_1.0.3
## [28] crayon_1.5.1            MASS_7.3-56             grid_4.1.3
## [31] lifecycle_1.0.1         registry_0.5-1          magrittr_2.0.3
## [34] scales_1.1.1            cli_3.2.0               stringi_1.7.6
## [37] xml2_1.3.3              ellipsis_0.3.2          vctrs_0.4.0
```

```
## [40] lpSolveAPI_5.5.2.0-17.7 tools_4.1.3            glue_1.6.2
## [43] parallel_4.1.3          fastmap_1.1.0          yaml_2.3.5
## [46] colorspace_2.0-3        scs_3.0-0              rvest_1.0.2
## [49] knitr_1.38
```

© 黄湘云

©黄湘云

附录 A　命令行操作

Bash 文件查找、查看（内容、大小）、移动（重命名）、删除、创建、修改权限

Linux 命令行工具是非常强大的，命令行中的数据科学 https://www.datascienceatthecommandline.com/，Linux 命令行 https://github.com/jaywcjlove/linux-command

optparse、docopt 、littler 包提供了很多便捷的命令行工具，sys，fs 在 R 中运行操作系统命令

如表A.1所示，总结了 R 和 Shell 命令的等价表示，下面以 `list.files()` 和 `ls` 为例，介绍其等价的内容

表 A.1: R 和 Shell 命令的等价表示 [1]

	R	Shell
查看文件	`list.files()`	`ls`
查看目录	`list.dirs()`	`dir`
目录层次	`fs::dir_tree()`	`tree`

A.1　查看文件

`ls/mkdir/mv/du`

查看文件

```bash
ls -a
```

列出目录下所有文件

```bash
ls -1
```

一行显示一个文件或文件夹

[1]CentOS 系统默认没有安装 tree 软件，需要先安装才能使用此命令 `sudo dnf install -y tree`

©黄湘云

```bash
ls -l
```

按从 aA-zZ 的顺序列出所有文件以及所属权限

```bash
ls -rl
```

相比于 `ls -l` 文件是逆序排列

```bash
ls -lh
```

列出文件或文件夹（不包含子文件夹）的大小

```bash
ls -ld
```

列出当前目录本身，而不是其所包含的内容

A.2 创建文件夹

```bash
mkdir images
```

创建文件用 `touch` 如 `touch .Rprofile`

```bash
# 删除文件夹及子文件夹，递归删除
rm -rf images/
# 删除文件
rm .Rprofile
```

A.3 移动文件

在当前目录下

© 黄湘云


```bash
# 移动文件夹 images 下的所有文件到 figures 文件夹下
mv images/* figures/
# images 文件夹移动到 figures 文件夹下
mv images/ figures/
# 移动特定的文件
mv images/*.png figures/
```

同一目录下有两个文件 `R-3.5.1.tar.gz` 未下载完整 和 `R-3.5.1.tar.gz.1` 完全下载

```bash
# 删除 R-3.5.1.tar.gz
rm R-3.5.1.tar.gz
# 重命名 R-3.5.1.tar.gz.1
mv R-3.5.1.tar.gz.1  R-3.5.1.tar.gz
```

A.4 查看文件大小

当前目录下各文件夹的大小，`-h` 表示人类（相对于机器来说）可读的方式显示，如 Kb、Mb、Gb，`-d` 表示目录

```bash
du -h -d 1 ./
```

```bash
# 对当前目录下的文件/夹 按大小排序
du -sh * | sort -nr
```

A.5 终端模拟器

oh-my-zsh 是 Z Shell 扩展，开发在 Github 上 https://github.com/ohmyzsh/ohmyzsh。

zsh 相比于 bash，在语法高亮、自动补全等方面有优势

```
sudo dnf install -y zsh
sh -c "$(curl -fsSL https://raw.github.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

RStudio 集成的终端支持 Zsh，操作路径 Tools -> Global Options -> Terminal，见图 A.1

© 科学出版社

图 A.1: RStudio IDE 集成的 Zsh 终端模拟器

© 基础工

A.6 压缩和解压缩

最常见的压缩文件格式有 .tar、.tar.gz、.tar.bz2、.zip 和 .rar，分别对应于 Tar https://www.gnu.org/software/tar/、Gzip https://www.gzip.org/ 、Bzip2 https://www.bzip.org/ 、UnZip/Zip http://www.info-zip.org 和 WinRAR https://www.rarlab.com/。Tar 提供了基本的打包和解包工具，Gzip 和 Bzip2 在 Tar 打包的基础上提供了压缩功能，UnZip/Zip 是兼容 Windows 原生压缩/解压缩功能的程序，WinRAR 是广泛流行于 Windows 系统的压缩/解压缩收费软件，除了 WinRAR，其它都是免费甚至开源软件。下面以 .tar.gz 和.tar.bz2 两种格式的压缩文件为例，介绍文件压缩和解压缩的操作，其它文件格式的操作类似[2]。WinRAR https://www.rarlab.com/ 是收费的压缩和解压缩工具，也支持 Linux 和 macOS 系统，鉴于它是收费软件，这里就不多展开介绍了，详情请见官网。

```
sudo dnf install -y tar gzip zip unzip
# 将目录 ~/tmp 压缩成文件 filename.tar.gz
tar -czf **.tar.gz ~/tmp
# 将文件 filename.tar.gz 解压到目录 ~/tmp
tar -xzf **.tar.gz -C ~/tmp
```

```
sudo dnf install -y bzip2
# 将目录 ~/tmp 压缩成文件 filename.tar.bz2
tar -cjf filename.tar.bz2 ~/tmp
# 将文件 filename.tar.bz2 解压到目录 ~/tmp
tar -xjf filename.tar.bz2 -C ~/tmp
```

解压不带 tar 的.gz 文件，比如 tex.eps.gz 解压后变成 tex.eps

```
gzip filename.gz -d ~/tmp
```

A.7 从仓库安装 R

A.7.1 Ubuntu

安装 openssh zsh 和 Git

```
sudo apt-get install zsh openssh-server
sudo add-apt-repository -y ppa:git-core/ppa
sudo apt update && sudo apt install git
sh -c "$(curl -fsSL https://raw.githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)
```

只考虑最新的 Ubuntu 18.04 因为本书写成的时候，该版本应该已经大规模使用了，默认版本的安装和之前版本的安装就不再展示了。安装最新版 R-3.5.x，无论安装哪个版本，都要先导入密钥

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv E084DAB9
```

- Ubuntu 14.04.5 提供的默认版本 R 3.0.2，安装 R 3.5.x 系列之前的版本，如 R 3.4.4

  ```
  sudo apt-add-repository -y "deb http://cran.rstudio.com/bin/linux/ubuntu `lsb_release -cs`/"
  sudo apt-get install r-base-dev
  ```

  添加完仓库后，都需要更新源 sudo apt-get update，安装 R 3.5.x 系列最新版

  ```
  sudo apt-add-repository -y "deb https://mirrors.tuna.tsinghua.edu.cn/CRAN/bin/linux/ubuntu trust
  ```

- Ubuntu 16.04.5 提供的默认版本 R 3.4.4，这是 R 3.4.x 系列的最新版，安装目前最新的 R 3.5.x 版本需要

[2]zip 格式的文件需要额外安装 zip 和 unzip 两款软件实现压缩和解压缩。

© 王贵湘


```
sudo apt-add-repository -y "deb https://mirrors.tuna.tsinghua.edu.cn/CRAN/bin/linux/ubuntu xenial-cra
```

- Ubuntu 18.04.1 提供的默认版本 R 3.4.4，安装目前的最新版本需要

```
sudo apt-add-repository -y "deb https://mirrors.tuna.tsinghua.edu.cn/CRAN/bin/linux/ubuntu bionic-cra
```

接下来安装 R，详细安装指导见 CRAN 官网。

```
sudo apt-get install -y r-base-dev
```

Michael Rutter 维护了编译好的二进制版本 https://launchpad.net/~marutter，比如 rstan 包可以通过安装 r-cran-rstan 完成

```
# R packages for Ubuntu LTS. Based on CRAN Task Views.
sudo add-apt-repository -y ppa:marutter/c2d4u3.5
sudo apt-get install r-cran-rstan
```

A.7.2 CentOS

同样适用于 Fedora

安装指导[3]

A.8 源码安装

A.8.1 Ubuntu

1. 首先启用源码仓库，否则执行 `sudo apt-get build-dep r-base` 会报如下错误

```
E: You must put some 'source' URIs in your sources.list
```

```
sudo sed -i -- 's/#deb-src/deb-src/g' /etc/apt/sources.list && sudo sed -i -- 's/# deb-src/deb-src/g' /etc
sudo apt-get update
```

1. 安装编译 R 所需的系统依赖

```
sudo apt-get build-dep r-base-dev
```

1. 编译安装 R

```
./configure
make && make install
```

1. 自定义编译选项

```
./configure --help
```

A.8.2 CentOS

基于 CentOS 7 和 GCC 4.8.5，参考 R-admin 手册

[3]在 CentOS 7 上打造 R 语言编程环境

© 黄湘云

- 下载源码包

  最新发布的稳定版

```
curl -fLo ./R-latest.tar.gz https://mirrors.tuna.tsinghua.edu.cn/CRAN/src/base/R-latest.tar.gz
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 10 28.7M   10 3232k    0     0   107k      0  0:04:34  0:00:30  0:04:04  118k
```

- 安装依赖

```
sudo yum install -y yum-utils epel-release && sudo yum-builddep R-devel
sudo dnf update && sudo dnf builddep R-devel # Fedora 30
```

- 解压配置

```
mkdir R-latest && tar -xzf ./R-latest.tar.gz -C ./R-latest && cd R-3.5.2

./configure --enable-R-shlib --enable-byte-compiled-packages \
  --enable-BLAS-shlib --enable-memory-profiling
```

```
R is now configured for x86_64-pc-linux-gnu

  Source directory:          .
  Installation directory:    /usr/local

  C compiler:                gcc -std=gnu99  -g -O2
  Fortran 77 compiler:       gfortran  -g -O2

  Default C++ compiler:      g++   -g -O2
  C++98 compiler:            g++ -std=gnu++98 -g -O2
  C++11 compiler:            g++ -std=gnu++11 -g -O2
  C++14 compiler:
  C++17 compiler:
  Fortran 90/95 compiler:    gfortran -g -O2
  Obj-C compiler:            gcc -g -O2 -fobjc-exceptions

  Interfaces supported:      X11, tcltk
  External libraries:        readline, curl
  Additional capabilities:   PNG, JPEG, TIFF, NLS, cairo, ICU
  Options enabled:           shared R library, shared BLAS, R profiling, memory profiling

  Capabilities skipped:
  Options not enabled:

  Recommended packages:      yes
```

- 编译安装

© 基础统计

```
make -j 2 all
sudo make install
```

- BLAS 加持（可选）

BLAS 对于加快矩阵计算至关重要，编译 R 带 BLAS 支持，添加 OpenBLAS 支持 `--with-blas="-lopenblas"` 或 ATLAS 支持 `--with-blas="-L/usr/lib64/atlas -lsatlas"`

```
sudo yum install -y openblas openblas-threads openblas-openmp

./configure --enable-R-shlib --enable-byte-compiled-packages \
  --enable-BLAS-shlib --enable-memory-profiling \
  --with-blas="-lopenblas"
```

```
R is now configured for x86_64-pc-linux-gnu

Source directory:          .
Installation directory:    /usr/local

C compiler:                gcc -std=gnu99  -g -O2
Fortran 77 compiler:       gfortran  -g -O2

Default C++ compiler:      g++   -g -O2
C++98 compiler:            g++ -std=gnu++98 -g -O2
C++11 compiler:            g++ -std=gnu++11 -g -O2
C++14 compiler:
C++17 compiler:
Fortran 90/95 compiler:    gfortran -g -O2
Obj-C compiler:            gcc -g -O2 -fobjc-exceptions

Interfaces supported:      X11, tcltk
External libraries:        readline, **BLAS(OpenBLAS)**, curl
Additional capabilities:   PNG, JPEG, TIFF, NLS, cairo, ICU
Options enabled:           shared R library, shared BLAS, R profiling, memory profiling

Capabilities skipped:
Options not enabled:

Recommended packages:      yes
```

配置成功的标志，如 OpenBLAS

```
checking for dgemm_ in -lopenblas... yes
checking whether double complex BLAS can be used... yes
checking whether the BLAS is complete... yes
```

ATLAS 加持

© 基础统计


```
sudo yum install -y atlas
./configure --enable-R-shlib --enable-byte-compiled-packages \
  --enable-BLAS-shlib --enable-memory-profiling \
  --with-blas="-L/usr/lib64/atlas -lsatlas"
```

```
R is now configured for x86_64-pc-linux-gnu

Source directory:          .
Installation directory:    /usr/local

C compiler:                gcc -std=gnu99  -g -O2
Fortran 77 compiler:       gfortran  -g -O2

Default C++ compiler:      g++   -g -O2
C++98 compiler:            g++ -std=gnu++98 -g -O2
C++11 compiler:            g++ -std=gnu++11 -g -O2
C++14 compiler:
C++17 compiler:
Fortran 90/95 compiler:    gfortran -g -O2
Obj-C compiler:            gcc -g -O2 -fobjc-exceptions

Interfaces supported:      X11, tcltk
External libraries:        readline, **BLAS(generic)**, curl
Additional capabilities:   PNG, JPEG, TIFF, NLS, cairo, ICU
Options enabled:           shared R library, shared BLAS, R profiling, memory profiling

Capabilities skipped:
Options not enabled:

Recommended packages:      yes
```

ATLAS 配置成功

```
checking for dgemm_ in -L/usr/lib64/atlas -lsatlas... yes
checking whether double complex BLAS can be used... yes
checking whether the BLAS is complete... yes
```

后续步骤同上

A.9 忍者安装

从源码自定义安装：加速 Intel MKL 和大文件支持

https://software.intel.com/en-us/articles/using-intel-mkl-with-r

© 非常流水

A.10 配置

A.10.1 初始会话 .Rprofile

.Rprofile 文件位于 ~/ 目录下或者 R 项目的根目录下

查看帮助 ?.Rprofile

更多配置设置 startup

A.10.2 环境变量 .Renviron

.Renviron 文件位于 ~/ 目录下

A.10.3 编译选项 Makevars

Makevars 文件位于 ~/.R/ 目录下

A.11 命令行参数

commandArgs 从终端命令行中传递参数

- rdoc 高亮 R 帮助文档中的 R 函数、关键字 NULL。启用需要在 R 控制台中执行 rdoc::use_rdoc()
- radian 代码自动补全和语法高亮，进入 R 控制台，终端中输入 radian
- docopt 提供 R 命令行工具，如 littler 包，getopt 从终端命令行接受参数
- optparse 命令行选项参数的解析器

安装完 R-littler R-littler-examples (centos) 或 littler r-cran-littler (ubuntu) 后，执行

```
# centos
sudo ln -s /usr/lib64/R/library/littler/examples/install.r /usr/bin/install.r
sudo ln -s /usr/lib64/R/library/littler/examples/install2.r /usr/bin/install2.r
sudo ln -s /usr/lib64/R/library/littler/examples/installGithub.r /usr/bin/installGithub.r
sudo ln -s /usr/lib64/R/library/littler/examples/testInstalled.r /usr/bin/testInstalled.r
# ubuntu
sudo ln -s /usr/lib/R/site-library/littler/examples/install.r /usr/bin/install.r
sudo ln -s /usr/lib/R/site-library/littler/examples/install2.r /usr/bin/install2.r
sudo ln -s /usr/lib/R/site-library/littler/examples/installGithub.r /usr/bin/installGithub.r
sudo ln -s /usr/lib/R/site-library/littler/examples/testInstalled.r /usr/bin/testInstalled.r
```

这样可以载终端中安装 R 包了

```
install.r docopt
```

```
#!/usr/bin/env Rscript
# 安装 optparse 提供更加灵活的传参方式
# 也可参考 littler https://github.com/eddelbuettel/littler
# if("optparse" %in% .packages(TRUE)) install.packages('optparse',repos = "https://cran.rstudio.com")
```

©黄湘云


```
# https://cran.r-project.org/doc/manuals/R-intro.html#Invoking-R-from-the-command-line
# http://www.cureffi.org/2014/01/15/running-r-batch-mode-linux/
args = commandArgs(trailingOnly=TRUE)

# 函数功能：在浏览器中同时打开多个 PDF 文档
open_pdf <- function(pdf_path = "./figures/", n = 1) {
  # pdf_path:     PDF文件所在目录
  # n:            默认打开1个PDF文档
  # PDF文档目录
  pdfs <- list.files(pdf_path, pattern = '\\.pdf$')
  # PDF 文档路径
  path_to_pdfs <- paste(pdf_path, pdfs, sep = .Platform$file.sep)
  # 打开 PDF 文档
  invisible(lapply(head(path_to_pdfs, n), browseURL))
}

open_pdf(pdf_path, n = args[1])

# 使用： Rscript --vanilla code/batch-open-pdf.R 20
```

A.12 从源码安装 R

从源码编译 R 的需求大概有以下几点：

1. 爱折腾的极客：玩配置，学习 make 相关工具和 Linux 世界的依赖
2. 追求性能：如 LFS 支持 和 Intel MKL 加速
3. 环境限制：CentOS 或者红帽系统，自带的 R 版本比较落后

```
./configure --prefix=/opt/R/R-devel \
  --enable-R-shlib --enable-byte-compiled-packages \
  --enable-BLAS-shlib --enable-memory-profiling --with-blas="-lopenblas"
```

```
R is now configured for x86_64-pc-linux-gnu

  Source directory:            .
  Installation directory:      /opt/R/R-devel

  C compiler:                  gcc  -g -O2
  Fortran fixed-form compiler: gfortran -fno-optimize-sibling-calls -g -O2

  Default C++ compiler:        g++ -std=gnu++11  -g -O2
  C++14 compiler:              g++ -std=gnu++14  -g -O2
  C++17 compiler:              g++ -std=gnu++17  -g -O2
  C++20 compiler:              g++ -std=gnu++2a  -g -O2
  Fortran free-form compiler:  gfortran -fno-optimize-sibling-calls -g -O2
  Obj-C compiler:
```

© 草稿

```
  Interfaces supported:        X11, tcltk
  External libraries:          pcre2, readline, BLAS(OpenBLAS), curl
  Additional capabilities:     PNG, JPEG, TIFF, NLS, cairo, ICU
  Options enabled:            shared R library, shared BLAS, R profiling, memory profiling

  Capabilities skipped:
  Options not enabled:

  Recommended packages:        yes
```

配置好了以后，可以编译安装了

```
make && sudo make install
```

flexiblas 支持多种 BLAS 库自由切换

A.13 安装软件

本书在后续章节中陆续用到新的 R 包，其安装过程不会在正文中呈现，下面以在 CentOS 8 上安装 **sf** 包为例介绍。首先需要安装一些系统依赖，具体安装哪些依赖参见 **sf** 包开发站点 https://github.com/r-spatial/sf。

```
sudo dnf config-manager --set-disabled PowerTools # openblas-devel
sudo dnf install -y sqlite-devel gdal-devel \
  proj-devel geos-devel udunits2-devel
```

然后，在 R 命令行窗口中，执行安装命令：

```
install.packages('sf')
```

至此，安装完成。如遇本地未安装的新 R 包，可从其官方文档中找寻安装方式。如果你完全不知道自己应该安装哪些，考虑把下面的依赖都安装上

```
sudo dnf install -y \
  # magick
  ImageMagick-c++-devel \
  # pdftools
  poppler-cpp-devel \
  # gifski
  cargo
```

软件包管理器架构图，各个命令分别担负什么样的功能，每个命令学习的一般路径是什么，而不是详细介绍每个命令、每个参数的使用，只需给出一个命令的完整使用即可，其余给出一个查询命令帮助手册

```
dnf copr
dnf config-manager
```

© 黄湘云

A.14 安装 R 包

Iñaki Ucar 开发的 cran2copr 项目实现在 Fedora 上安装预编译好的二进制 R 包，项目目的类似 Debian 平台上的 cran2deb

1. devtools 是开发 R 包的常用工具，同时具有很重的依赖，请看

```
tools::package_dependencies('devtools', recursive = TRUE)
```

```
## $devtools
##  [1] "usethis"     "callr"       "cli"         "desc"        "ellipsis"
##  [6] "fs"          "httr"        "lifecycle"   "memoise"     "pkgbuild"
## [11] "pkgload"     "rcmdcheck"   "remotes"     "rlang"       "roxygen2"
## [16] "rstudioapi"  "rversions"   "sessioninfo" "stats"       "testthat"
## [21] "tools"       "utils"       "withr"       "processx"    "R6"
## [26] "glue"        "rprojroot"   "methods"     "curl"        "jsonlite"
## [31] "mime"        "openssl"     "cachem"      "crayon"      "prettyunits"
## [36] "digest"      "xopen"       "brew"        "commonmark"  "knitr"
## [41] "purrr"       "stringi"     "stringr"     "xml2"        "cpp11"
## [46] "brio"        "evaluate"    "magrittr"    "praise"      "ps"
## [51] "waldo"       "clipr"       "gert"        "gh"          "rappdirs"
## [56] "whisker"     "yaml"        "graphics"    "grDevices"   "fastmap"
## [61] "askpass"     "credentials" "sys"         "zip"         "gitcreds"
## [66] "ini"         "highr"       "xfun"        "diffobj"     "fansi"
## [71] "rematch2"    "tibble"      "pillar"      "pkgconfig"   "vctrs"
## [76] "utf8"
```

其中，依赖关系见表 A.2

表 A.2: devtools 的系统依赖

	curl	git2r	openssl
Ubuntu	libcurl-dev[4]	libgit2-dev	libssl-dev
CentOS	libcurl-devel	libgit2-devel	openssl-devel

1. sf 是处理空间数据的常用工具

```
tools::package_dependencies('sf', recursive = TRUE)
```

```
## $sf
##  [1] "methods"    "classInt"   "DBI"        "graphics"   "grDevices"
##  [6] "grid"       "magrittr"   "Rcpp"       "s2"         "stats"
## [11] "tools"      "units"      "utils"      "e1071"      "class"
## [16] "KernSmooth" "wk"         "MASS"       "proxy"
```

其主要的系统依赖分别是 GEOS 3.5.1, GDAL 2.2.2, PROJ 4.9.2

[4]libcurl-dev 是一个虚包 virtual package，由 libcurl4-openssl-dev 或 libcurl4-nss-dev 或 libcurl4-gnutls-dev 实际提供，选择其中一个安装即可。

© 高等教育出版社

```
sudo add-apt-repository -y ppa:ubuntugis/ubuntugis-unstable
sudo apt-get update
sudo apt-get install -y libudunits2-dev libgdal-dev libgeos-dev libproj-dev
```

这样也同时解决了 udunits2、rgdal 和 rgeos 等 3 个 R 包的系统依赖，其中 udunits2 使用如下命令安装

```
install.packages(' udunits2' , configure.args = '--with-udunits2-include=/usr/include/udunits2')
```

2. 图形设备支持 cairo png jpeg tiff

```
sudo apt-get install -y libcairo2-dev libjpeg-dev libpng-dev libtiff-dev
```

3. 图像处理 imager 和 magick

```
sudo yum install fftw-devel # CentOS
sudo apt-get install libfftw3-dev # Ubuntu
```

在 Ubuntu 系统上安装最新的 libmagick++-dev 库

```
sudo add-apt-repository -y ppa:opencpu/imagemagick
sudo apt-get update
sudo apt-get install -y libmagick++-dev
```

在 CentOS 系统上

```
sudo yum install -y ImageMagick-c++-devel
```

然后安装 R 包 `install.packages(c('imager', 'magick'))`

4. rgl 是绘制真三维图形的重量级 R 包

```
sudo apt-get install libcgal-dev libglu1-mesa-dev libx11-dev # Ubuntu
sudo yum install mesa-libGLU mesa-libGLU-devel # CentOS
```

然后安装 R 包

```
install.packages('rgl')
```

在 Ubuntu 系统上还可以这样安装

```
sudo add-apt-repository ppa:marutter/rrutter3.5
sudo apt-get update
sudo apt-get install r-cran-rgl
```

5. rJava 是 Java 语言和 R 语言之间实现通信交流的桥梁

```
sudo apt-get install -y default-jdk
sudo R CMD javareconf
```

然后安装 rJava 包 `install.packages(' rJava' )`

6. igraph 是网络数据分析的必备 R 包，为了发挥其最大性能，需要安装三个系统依赖

```
sudo apt-get install -y libgmp-dev libxml2-dev libglpk-dev
```

然后安装 R 包

© 王建国


```
install.packages('igraph')
```

7. gpuR 是基于 GPU 进行矩阵计算的扩展包，依赖 RcppEigen 确保安装 OpenCL 和 RViennaCL 或者安装 Nvidia 驱动和 CUDA，使用 gpuRcuda 和 gputools 扩展包，下面安装指导来自其 Wiki

```
# Install OpenCL headers
sudo apt-get install opencl-headers opencv-dev

# Install NVIDIA Drivers and CUDA
sudo add-apt-repository -y ppa:xorg-edgers/ppa
sudo apt-get update
sudo apt-get install nvidia-346 nvidia-settings
```

8. nloptr 是 NLopt 的 R 语言接口，首先安装 NLopt 程序库 `sudo apt-get install libnlopt-dev` 然后安装 R 包 `install.packages('nloptr')`，nloptr 被 700+ R 包依赖，如 **lme4**，**spaMM**，**glmmTMB**，**rstanarm** 等。

9. Rmpfr

```
sudo apt-get install libmpfr-dev
```

```
install.packages(' Rmpfr' )
```

10. geojson

```
sudo yum install jq-devel protobuf-devel
```

```
install.packages(c('geojson','geojsonio','jqr','protolite'))
```

11. lgcp

```
sudo yum install bwidget
```

```
install.packages(c('rpanel','lgcp'))
```

12. ijtiff

```
sudo yum install jbigkit-devel
```

```
install.packages('ijtiff')
```

13. webshot 包用于截图

```
sudo apt install phantomjs
```

```
install.packages(' webshot' )
```

14. gifski 包合成 GIF 动图

```
sudo apt-get install cargo
```

```
install.packages('gifski')
```

©黄海广

A.15 软件包管理器

A.15.1 dnf

1. 清理升级后的 CentOS 8 系统内核

查找系统安装的内核

```
rpm -qa | sort | grep kernel
```

```
kernel-4.18.0-147.8.1.el8_1.x86_64
kernel-4.18.0-193.6.3.el8_2.x86_64
kernel-core-4.18.0-147.8.1.el8_1.x86_64
kernel-core-4.18.0-193.6.3.el8_2.x86_64
kernel-headers-4.18.0-193.6.3.el8_2.x86_64
kernel-modules-4.18.0-147.8.1.el8_1.x86_64
kernel-modules-4.18.0-193.6.3.el8_2.x86_64
kernel-tools-4.18.0-193.6.3.el8_2.x86_64
kernel-tools-libs-4.18.0-193.6.3.el8_2.x86_64
```

仅保留一个版本的内核，其它旧的内核都删除掉

```
sudo dnf remove $(dnf repoquery --installonly --latest-limit=-1 -q)
```

```
模块依赖问题

 问题 1: conflicting requests
 - nothing provides module(perl:5.26) needed by module perl-DBD-MySQL:4.046:8010020191114030811:073fa5fe-0.x
 问题 2: conflicting requests
 - nothing provides module(perl:5.26) needed by module perl-DBI:1.641:8010020191113222731:16b3ab4d-0.x86_64
 问题 3: conflicting requests
 - nothing provides module(perl:5.26) needed by module perl-YAML:1.24:8010020191114031501:a5949e2e-0.x86_64
依赖关系解决。
=================================================================================================
 软件包                 架构            版本                        仓库             大小
=================================================================================================
移除:
 kernel                x86_64          4.18.0-147.8.1.el8_1        @BaseOS          0
 kernel-core           x86_64          4.18.0-147.8.1.el8_1        @BaseOS          58 M
 kernel-modules        x86_64          4.18.0-147.8.1.el8_1        @BaseOS          20 M

事务概要
=================================================================================================
移除  3 软件包

将会释放空间: 78 M
确定吗？[y/N]:  y
运行事务检查
```

© 基础


```
事务检查成功。
运行事务测试
事务测试成功。
运行事务
  准备中  :                                                                    1/1
  删除   : kernel-4.18.0-147.8.1.el8_1.x86_64                                    1/3
  运行脚本: kernel-4.18.0-147.8.1.el8_1.x86_64                                    1/3
  删除   : kernel-modules-4.18.0-147.8.1.el8_1.x86_64                            2/3
  运行脚本: kernel-modules-4.18.0-147.8.1.el8_1.x86_64                            2/3
  运行脚本: kernel-core-4.18.0-147.8.1.el8_1.x86_64                              3/3
  删除   : kernel-core-4.18.0-147.8.1.el8_1.x86_64                               3/3
  运行脚本: kernel-core-4.18.0-147.8.1.el8_1.x86_64                              3/3
  验证   : kernel-4.18.0-147.8.1.el8_1.x86_64                                    1/3
  验证   : kernel-core-4.18.0-147.8.1.el8_1.x86_64                               2/3
  验证   : kernel-modules-4.18.0-147.8.1.el8_1.x86_64                            3/3

已移除:
  kernel-4.18.0-147.8.1.el8_1.x86_64                kernel-core-4.18.0-147.8.1.el8_1.x86_64
  kernel-modules-4.18.0-147.8.1.el8_1.x86_64

完毕!
```

解决上述模块依赖问题的办法是重置三个 Perl 模块

```
sudo dnf module reset perl-DBD-MySQL perl-YAML perl-DBI
```

```
依赖关系解决。
================================================================================================
 软件包                 架构                 版本                 仓库                 大小
================================================================================================
重置模块:
 perl-DBD-MySQL
 perl-DBI
 perl-YAML

事务概要
================================================================================================

确定吗？[y/N]:  y
完毕!
```

A.15.2 apt

添加或删除 PPA (Personal Package Archive)，比如在 Ubuntu 20.04 及之前的版本上安装新版 Inkscape

```
sudo add-apt-repository ppa:inkscape.dev/stable
sudo add-apt-repository --remove ppa:inkscape.dev/stable
```

© 基础

```
sudo apt-get install build-essential # 修复依赖问题
sudo apt update # 更新资源列表
sudo apt-get upgrade # 更新软件包
sudo apt-get autoclean # 删除已卸的软件的备份
sudo apt-get clean # 删除已装或已卸的软件的备份
sudo apt-get autoremove --purge * # 推荐卸载软件的方式
apt-get list --upgradable # 列出可升级的包
```

找到并删除旧的内核

```
dpkg --list | grep linux-image
sudo apt-get purge linux-image-3.19.0-{18,20,21,25}
sudo update-grub2
```

```
# 搜索
apt-cache search octave | grep octave
# 查询
apt show octave
# 安装
sudo apt install octave
```

```
sudo apt-get install lsb-core
lsb_release -a
```

```
adduser cloud2016 # 添加用户
passwd cloud2016  # 用户密码设为 cloud
whereis sudoers   # 查找文件位置
chmod -v u+w /etc/sudoers # 给文件 sudoers 添加写权限
vim /etc/sudoers # 添加 cloud2016 管理员权限
chmod -v u-w /etc/sudoers # 收回权限
```

安装确认 openssh-server 服务

```
sudo apt install openssh-server
sudo /etc/init.d/ssh start
ps -aux | grep ssh
```

©黄湘云

附录 B　矩阵运算

Eigenvectors from Eigenvalues [Denton et al., 2019]

参考 matlib 和 Matrix 包，SparseM 更加强调稀疏矩阵的 Cholesky 分解和后退法，矩阵取子集和 Kronecker 积。矩阵计算一般介绍参考在线书籍 Stephen Boyd and Lieven Vandenberghe 最新著作 Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares [Boyd and Vandenberghe, 2018] 及其 Julia 语言实现，矩阵分解部分参考 Introduction to Linear Algebra, 5th Edition、Linear Algebra for Data Science with examples in R

fastmatrix、abind 各种矩阵操作。

Evan Chen 的书An Infinitely Large Napkin 介绍矩阵代数的内积空间、群、环、域等高级内容，作者提供免费的电子书。

分块矩阵操作，各类分解算法，及其 R 实现

```
library(Matrix)
```

以 attitude 数据集为例介绍各种矩阵操作

```
head(attitude)
```

```
##   rating complaints privileges learning raises critical advance
## 1     43         51         30       39     61       92      45
## 2     63         64         51       54     63       73      47
## 3     71         70         68       69     76       86      48
## 4     61         63         45       47     54       84      35
## 5     81         78         56       66     71       83      47
## 6     43         55         49       44     54       49      34
```

rating 总体评价 complaints 处理员工投诉 privileges 不允许特权 learning 学习机会 raises 根据表现晋升 critical 批评 advancel 进步

```
fit <- lm(rating ~. , data = attitude)
summary(fit) # 模型是显著的，很多变量的系数不显著
```

```
##
## Call:
## lm(formula = rating ~ ., data = attitude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9418  -4.3555   0.3158   5.5425  11.5990
```

© 基础统计


```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.78708   11.58926   0.931 0.361634
## complaints   0.61319    0.16098   3.809 0.000903 ***
## privileges  -0.07305    0.13572  -0.538 0.595594
## learning     0.32033    0.16852   1.901 0.069925 .
## raises       0.08173    0.22148   0.369 0.715480
## critical     0.03838    0.14700   0.261 0.796334
## advance     -0.21706    0.17821  -1.218 0.235577
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.068 on 23 degrees of freedom
## Multiple R-squared:  0.7326, Adjusted R-squared:  0.6628
## F-statistic:  10.5 on 6 and 23 DF,  p-value: 1.24e-05
```

```r
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: rating
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## complaints  1 2927.58 2927.58 58.6026 9.056e-08 ***
## privileges  1    7.52    7.52  0.1505    0.7016
## learning    1  137.25  137.25  2.7473    0.1110
## raises      1    0.94    0.94  0.0189    0.8920
## critical    1    0.56    0.56  0.0113    0.9163
## advance     1   74.11   74.11  1.4835    0.2356
## Residuals  23 1149.00   49.96
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
attitude_mat <- as.matrix.data.frame(attitude)
# 生成演示用的矩阵
demo_mat <- t(attitude_mat[, -1]) %*% attitude_mat[, -1]
```

B.1 矩阵乘法

```r
A <- matrix(c(1, 2, 2, 3), nrow = 2)
A
```

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

© 王建硕


```
B <- matrix(c(1, 2, 3, 4, 5, 6), nrow =2)
B
```

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

通常的矩阵乘法也叫矩阵内积

```
A %*% B
```

```
##      [,1] [,2] [,3]
## [1,]    5   11   17
## [2,]    8   18   28
```

```
A ** 2
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    4    9
```

```
A ^ 2
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    4    9
```

```
A ** A
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    4   27
```

B.2 Hadamard 积

Hadamard 积（法国数学家 Jacques Hadamard）也叫 Schur 积（德国数学家 Issai Schur ）或 entrywise 积是两个维数相同的矩阵对应元素相乘，特别地，A^2 表示将矩阵 A 的每个元素平方

$$(A \circ B)_{ij} = (A)_{ij}(B)_{ij}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \circ \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & a_{13}b_{13} \\ a_{21}b_{21} & a_{22}b_{22} & a_{23}b_{23} \\ a_{31}b_{31} & a_{32}b_{32} & a_{33}b_{33} \end{bmatrix}$$

```
A^2
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    4    9
```

©黄湘云

B.3 矩阵转置

```
t(B)
```

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

B.4 矩阵外积

```
A %o% B # outer(A, B, FUN = "*")
```

```
## , , 1, 1
##
##      [,1] [,2]
## [1,]    1    2
## [2,]    2    3
##
## , , 2, 1
##
##      [,1] [,2]
## [1,]    2    4
## [2,]    4    6
##
## , , 1, 2
##
##      [,1] [,2]
## [1,]    3    6
## [2,]    6    9
##
## , , 2, 2
##
##      [,1] [,2]
## [1,]    4    8
## [2,]    8   12
##
## , , 1, 3
##
##      [,1] [,2]
## [1,]    5   10
## [2,]   10   15
##
## , , 2, 3
```

© 王建平


```
##
##      [,1] [,2]
## [1,]    6   12
## [2,]   12   18
```

直积/克罗内克积

```
A %x% B # kronecker(A, B, FUN = "*")
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    3    5    2    6   10
## [2,]    2    4    6    4    8   12
## [3,]    2    6   10    3    9   15
## [4,]    4    8   12    6   12   18
```

B.5 矩阵乘方

矩阵 A 首先是一个方阵，对称性和正定性未知，n 个矩阵 A 相乘

统计之都论坛讨论如何求矩阵的乘方 https://d.cosx.org/d/5619-svd

```
"%^%" <- function(mat, pow) {
  if (!is.matrix(mat)) mat <- as.matrix(mat)
  stopifnot(!diff(dim(mat)))
  if (pow < 0) {
    pow <- -pow
    mat <- solve(mat)
  }
  pow <- round(pow)
  switch(pow + 1, return(diag(1, nrow(mat))), return(mat))
  get.exponents <- function(pow)
    if (pow == 0) NULL else c(k <- 2^floor(log2(pow)), get.exponents(pow - k))
  ans <- diag(nrow(mat))
  dlog2exp <- rev(-diff(c(log2(get.exponents(pow)), 0)))
  for (j in 1:length(dlog2exp)) {
    if (dlog2exp[j]) for (i in 1:dlog2exp[j]) mat <- mat %*% mat
    ans <- ans %*% mat
  }
  ans
}
```

奇异值分解

```
s <- svd(A)
all.equal(s$u%*%diag(s$d)%*%t(s$v),A)
```

```
## [1] TRUE
```

特征值及分解 $A = V\Lambda V^{-1}$ 求解矩阵 A 的 n 次方

© 草稿


```
eigen(A)
```

```
## eigen() decomposition
## $values
## [1]  4.236068 -0.236068
##
## $vectors
##           [,1]       [,2]
## [1,] 0.5257311 -0.8506508
## [2,] 0.8506508  0.5257311
```

```
eigen(A)$vectors %*% diag(eigen(A)$values) %*% solve(eigen(A)$vectors)
```

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

```
eigen(A)$vectors %*% diag(eigen(A)$values)^3 %*% solve(eigen(A)$vectors)
```

```
##      [,1] [,2]
## [1,]   21   34
## [2,]   34   55
```

```
A %*% A %*% A
```

```
##      [,1] [,2]
## [1,]   21   34
## [2,]   34   55
```

B.6 矩阵求幂

```
2^A
```

```
##      [,1] [,2]
## [1,]    2    4
## [2,]    4    8
```

```
exp(A)
```

```
##          [,1]      [,2]
## [1,] 2.718282  7.389056
## [2,] 7.389056 20.085537
```

expm 包含更多关于矩阵开方、取对数等计算

B.7 矩阵交叉积

交叉积 $A^\top A$

© 基础统计


```
crossprod(A, A)  #  t(x) %*% y
```

```
##      [,1] [,2]
## [1,]    5    8
## [2,]    8   13
```

```
tcrossprod(A, A) #  x %*% t(y)
```

```
##      [,1] [,2]
## [1,]    5    8
## [2,]    8   13
```

B.8 矩阵行列式

```
det(A)
```

```
## [1] -1
```

expm 包计算矩阵 $e^{\mathbf{A}}$

B.9 矩阵条件数

```
library(Matrix)
base::rcond(A)
```

```
## [1] 0.04
```

```
kappa(A)
```

```
## [1] 21.85714
```

```
Matrix::rcond(Matrix::Hilbert(6))
```

```
## [1] 3.439939e-08
```

```
Matrix::rcond(A)
```

```
## [1] 0.04
```

B.10 矩阵求逆

```
solve(A)
```

```
##      [,1] [,2]
## [1,]   -3    2
## [2,]    2   -1
```

应用之线性方程组

©李东风

```
B <- Hilbert(6)
b <- rowSums(B)
# not inv
solve(B,b)
```

```
## 6 x 1 Matrix of class "dgeMatrix"
##      [,1]
## [1,]    1
## [2,]    1
## [3,]    1
## [4,]    1
## [5,]    1
## [6,]    1
```

```
# inv
solve(B) %*% b
```

```
## 6 x 1 Matrix of class "dgeMatrix"
##      [,1]
## [1,]    1
## [2,]    1
## [3,]    1
## [4,]    1
## [5,]    1
## [6,]    1
```

Moore-Penrose generalized inverse 广义逆，如果 A 可逆则，广义逆就是逆

```
library(MASS) # ginv 来自 MASS 包
ginv(A)
```

```
##      [,1] [,2]
## [1,]   -3    2
## [2,]    2   -1
```

```
A %*% ginv(A) %*% A
```

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

```
ginv(A) %*% A %*% ginv(A)
```

```
##      [,1] [,2]
## [1,]   -3    2
## [2,]    2   -1
```

```
t(A %*% ginv(A))
```

```
##               [,1]         [,2]
## [1,]  1.000000e+00 8.881784e-16
```

© 王湘君


```
## [2,] -8.881784e-16 1.000000e+00
```

```
A %*% ginv(A)
```

```
##              [,1]          [,2]
## [1,] 1.000000e+00 -8.881784e-16
## [2,] 8.881784e-16  1.000000e+00
```

```
t(ginv(A) %*% A)
```

```
##              [,1]          [,2]
## [1,] 1.000000e+00 -8.881784e-16
## [2,] 8.881784e-16  1.000000e+00
```

```
ginv(A) %*% A
```

```
##               [,1]         [,2]
## [1,]  1.000000e+00 8.881784e-16
## [2,] -8.881784e-16 1.000000e+00
```

B.11 矩阵伴随

伴随矩阵 $A * A^\star = A^\star * A = |A| * I, A^\star = |A| * A^{-1}$

- $|A^\star| = |A|^{n-1}, A \in \mathbb{R}^{n\times n}, n \geq 2$
- $(A^\star)^\star = |A|^{n-2}A, A \in \mathbb{R}^{n\times n}, n \geq 2$
- $(A^\star)^\star$ A 的 n 次伴随是?

```
det(A)*solve(A)
```

```
##      [,1] [,2]
## [1,]    3   -2
## [2,]   -2    1
```

B.12 矩阵范数

向量和矩阵的范数，包括 1，2，无穷范数，其他操作看 Matrix 包，尤其关于稀疏矩阵计算部分

1-范数 列和绝对值最大的
∞ - 范数 行和绝对值最大的
Frobenius - 范数 Euclidean 范数
M - 范数 矩阵里模最大的元素，矩阵里面的元素可能含有复数，所以取模最大
2 - 范数 又称谱范数，矩阵最大的奇异值，如果是方阵，就是最大的特征值

```
norm(A, type = "1") # max(abs(colSums(A)))
```

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

```
norm(A, type = "I") # max(abs(rowSums(A)))
```

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

©黄文彬

```
norm(A, type = "F")
```

```
## [1] 4.242641
```

```
norm(A, type = "M") #
```

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

```
norm(A, type = "2") # max(svd(A)$d)
```

```
## [1] 4.236068
```

显然，$1-, \infty-, M-$ 的范数计算比 $F-$ 范数快，函数 `norm` 默认情况下求 $1-$ 范数

B.13 矩阵求秩

```
qr(A)$rank # or qr.default(A)$rank
```

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

B.14 矩阵求迹

若

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

则矩阵 A 的迹 $\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}$

```
sum(diag(A))
```

```
## [1] 4
```

特殊矩阵的构造

B.15 单位矩阵

矩阵对角线上全是 1，其余位置都是 0

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
diag(rep(3))
```

©黄湘云


```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    1    0
## [3,]    0    0    1
```

而全 1 矩阵是所有元素都是 1 的矩阵，可以借助外积运算构造，如 3 阶全 1 矩阵

```
rep(1,3) %o% rep(1,3)
```

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

B.16 对角矩阵

```
diag(A)       # diagonal of a matrix
```

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

```
diag(diag(A)) # construct a diagonal matrix
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    0    3
```

B.17 上/下三角矩阵

矩阵下三角

row 和 col

```
row(A)
```

```
##      [,1] [,2]
## [1,]    1    1
## [2,]    2    2
```

```
col(A)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    2
```

```
A[row(A)]
```

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

```
upper.tri(A) # 矩阵上三角
```

```
##       [,1]  [,2]
## [1,] FALSE  TRUE
## [2,] FALSE FALSE
```

```
A[upper.tri(A)]
```

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

```
A[lower.tri(A)] <- 0 # 获得上三角矩阵
A
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    0    3
```

- 下三角矩阵

```
A <- matrix(c(1, 2, 2, 3), nrow = 2)
A
```

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

```
lower.tri(A)
```

```
##       [,1]  [,2]
## [1,] FALSE FALSE
## [2,]  TRUE FALSE
```

```
A[lower.tri(A)]
```

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

```
A[upper.tri(A)] <- 0 # 获得下三角矩阵
A
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    2    3
```

```
A <- matrix(c(1, 2, 2, 3), nrow = 2)
A[row(A) < col(A)] <- 0
A
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    2    3
```

© 王兼

B.18 稀疏矩阵

```
dn <- list(LETTERS[1:3], letters[1:5])
## pointer vectors can be used, and the (i,x) slots are sorted if necessary:
## 使用指针构造
m <- sparseMatrix(i = c(3,1, 3:2, 2:1), p= c(0:2, 4,4,6), x = 1:6, dimnames = dn)
m
```

```
## 3 x 5 sparse Matrix of class "dgCMatrix"
##   a b c d e
## A . 2 . . 6
## B . . 4 . 5
## C 1 . 3 . .
```

```
## 典型构造方式
i <- c(1,3:8); j <- c(2,9,6:10); x <- 7 * (1:7)
(AA <- sparseMatrix(i, j, x = x))                    ##  8 x 10 "dgCMatrix"
```

```
## 8 x 10 sparse Matrix of class "dgCMatrix"
##
## [1,] . 7 . . .  .  .  .  .  .
## [2,] . . . . .  .  .  .  .  .
## [3,] . . . . .  .  .  . 14  .
## [4,] . . . . . 21  .  .  .  .
## [5,] . . . . .  . 28  .  .  .
## [6,] . . . . .  .  . 35  .  .
## [7,] . . . . .  .  .  . 42  .
## [8,] . . . . .  .  .  .  . 49
```

B.19 三对角矩阵

B.20 LU 分解

B.21 Schur 分解

B.22 Cholesky 分解

实对称正定矩阵的 Choleski 分解

```
chol(A + diag(rep(1,2)))
```

```
##          [,1] [,2]
## [1,] 1.414214    0
## [2,] 0.000000    2
```

内部资料 ©

```
# Inverse from Choleski (or QR) Decomposition
Matrix::chol2inv(A + diag(rep(1,2)))
```

```
##      [,1]   [,2]
## [1,] 0.25 0.0000
## [2,] 0.00 0.0625
```

`Matrix::Cholesky` 实现稀疏 Cholesky 分解

B.23 特征值分解

特征值分解（Eigenvalues Decomposition）也叫谱分解（Spectral Decomposition）

```
eigen(A)
```

```
## eigen() decomposition
## $values
## [1] 3 1
##
## $vectors
##      [,1]       [,2]
## [1,]    0  0.7071068
## [2,]    1 -0.7071068
```

B.24 SVD 分解

Fast truncated singular value decompositions 奇异值分解是特征值分解的推广

```
svd(A)
```

```
## $d
## [1] 3.6502815 0.8218544
##
## $u
##            [,1]       [,2]
## [1,] -0.1601822 -0.9870875
## [2,] -0.9870875  0.1601822
##
## $v
##            [,1]       [,2]
## [1,] -0.5847103 -0.8112422
## [2,] -0.8112422  0.5847103
```

```
svd(A)$d
```

```
## [1] 3.6502815 0.8218544
```

© 草稿

邱怡轩将奇异值分解用于图像压缩 https://cosx.org/2014/02/svd-and-image-compression 并制作了 Shiny App 交互式演示

B.25 QR 分解

```
qr.default(A)
```

```
## $qr
##            [,1]      [,2]
## [1,] -2.2360680 -2.683282
## [2,]  0.8944272  1.341641
##
## $rank
## [1] 2
##
## $qraux
## [1] 1.447214 1.341641
##
## $pivot
## [1] 1 2
##
## attr(,"class")
## [1] "qr"
```

```
qr.X(qr.default(A))
```

```
##      [,1] [,2]
## [1,]    1    0
## [2,]    2    3
```

```
qr.Q(qr.default(A))
```

```
##            [,1]       [,2]
## [1,] -0.4472136 -0.8944272
## [2,] -0.8944272  0.4472136
```

```
qr.R(qr.default(A))
```

```
##           [,1]      [,2]
## [1,] -2.236068 -2.683282
## [2,]  0.000000  1.341641
```

```
qr.Q(qr.default(A)) %*% qr.R(qr.default(A))
```

```
##      [,1]          [,2]
## [1,]    1 -2.220446e-16
## [2,]    2  3.000000e+00
```

用 Householder 变换 做 QR 分解 [Bates and Watts, 1988] 及其 R 语言实现 https://rpubs.com/aaronsc32/

内部草稿 ©

qr-decomposition-householder

Householder 变换是平面反射的一般情况：要计算 $N \times P$ 维矩阵 X 的 QR 分解，我们采用 Householder 变换

$$\mathbf{H}_u = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$$

其中 I 是 $N \times N$ 维的单位矩阵，u 是 N 维单位向量，即 $\|\mathbf{u}\| = \sqrt{\mathbf{u}\mathbf{u}^\top} = 1$。则 H_u 是对称正交的，因为

$$\mathbf{H}_u^\top = \mathbf{I}^\top - 2\mathbf{u}\mathbf{u}^\top = \mathbf{H}_u$$

并且

$$\mathbf{H}_u^\top \mathbf{H}_u = \mathbf{I} - 4\mathbf{u}\mathbf{u}^\top + 4\mathbf{u}\mathbf{u}^\top\mathbf{u}\mathbf{u}^\top = \mathbf{I}$$

让 $\mathbf{H}_u$ 乘以向量 $\mathbf{y}$，即

$$\mathbf{H}_u\mathbf{y} = \mathbf{y} - 2\mathbf{u}\mathbf{u}^\top\mathbf{y}$$

它是 y 关于垂直于过原点的 u 的直线的反射，只要

$$\mathbf{u} = \frac{\mathbf{y} - \|\mathbf{y}\|\mathbf{e}_1}{\|\mathbf{y} - \|\mathbf{y}\|\mathbf{e}_1\|} \tag{B.1}$$

或者

$$\mathbf{u} = \frac{\mathbf{y} + \|\mathbf{y}\|\mathbf{e}_1}{\|\mathbf{y} + \|\mathbf{y}\|\mathbf{e}_1\|} \tag{B.2}$$

其中 $\mathbf{e}_1 = (1, 0, \ldots, 0)^\top$，Householder 变换使得向量 y 成为 x 轴，在新的坐标系统中，向量 $H_u y$ 的坐标为 $(\pm\|y\|, 0, \ldots, 0)^\top$

举个例子

借助 Householder 变换做 QR 分解的优势：

1. 更快、数值更稳定比直接构造 Q，特别当 N 大于 P 的时候
2. 相比于存储矩阵 Q 的 N^2 个元素，Householder 变换只存储 P 个向量 $u_1, \ldots, u_P$
3. QR 分解的真实实现，比如在 LINPACK 中，定义 u 的时候，公式 (B.1) 或 (B.2) 的选择基于 y 的第一个坐标的符号。如果坐标是负的，使用公式(B.1)，如果是正的，使用公式 (B.2)，这个做法可以使得数值计算更加稳定。

Stan 实现的 QR 分解在贝叶斯线性回归模型中的应用[1]

[1] https://mc-stan.org/users/documentation/case-studies/qr_regression.html

© 王建平

B.26 Jordan 分解

B.27 Givens 旋转

- Givens 旋转 https://www.wikiwand.com/en/Givens_rotation
- 帽子矩阵在统计中的应用回归与方差分析 [Hoaglin and Welsch, 1978]

B.28 特殊函数

B.28.1 阶乘

- 阶乘 $n! = 1 \times 2 \times 3 \cdots n$
- 双阶乘 $(2n+1)!! = 1 \times 3 \times 5 \times \cdots \times (2n+1), n = 0, 1, 2, \cdots$

```
factorial(5) # 阶乘
```

```
## [1] 120
```

```
seq(from = 1, to = 5, length.out = 3)
```

```
## [1] 1 3 5
```

```
prod(seq(from = 1, to = 5, length.out = 3)) # 连乘 双阶乘
```

```
## [1] 15
```

```
seq(5)
```

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

```
cumprod(seq(5)) # 累积
```

```
## [1]   1   2   6  24 120
```

```
cumsum(seq(5)) # 累和
```

```
## [1]  1  3  6 10 15
```

此外还有 cummax 和 cummin

- 组合数 $C_n^k = \frac{n(n-1)(n-k+1)}{k!}$

$C_5^3 = \frac{5\times4\times3}{3\times2\times1}$

```
choose(5,3)
```

```
## [1] 10
```

斯特林公式

内部资料 ©

B.28.2 伽马函数

$\Gamma(x) = \int_0^\infty t^{x-1}\exp(-t)dt\,\Gamma(n) = (n-1)!, n \in \mathbb{Z}^+$

```
gamma(2)
```

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

```
gamma(10)
```

```
## [1] 362880
```

```
gamma2 <- function(t,x){
  t^(x-1)*exp(-t)
}
integrate(gamma2, lower = 0, upper = + Inf, x = 10)
```

```
## 362880 with absolute error < 0.025
```

- psigamma(x, deriv) 表示 $\psi(x)$ 的 `deriv` 阶导数

$\text{digamma}(x) \triangleq \psi(x) = \frac{d}{dx}\ln\Gamma(x) = \Gamma'(x)/\Gamma(x)$

```
# 例1
x <- 2
eval(deriv(~ gamma(x), "x"))/gamma(x)
```

```
## [1] 1
## attr(,"gradient")
##               x
## [1,] 0.4227843
```

```
# 与此等价
psigamma(2, 0)
```

```
## [1] 0.4227843
```

```
digamma(x) # psi(x) 的一阶导数
```

```
## [1] 0.4227843
```

```
trigamma(x) # psi(x) 的二阶导数
```

```
## [1] 0.6449341
```

```
# 例2
eval(deriv(~ psigamma(x, 1), "x"))
```

```
## [1] 0.6449341
## attr(,"gradient")
##                x
## [1,] -0.4041138
```

```
# 与此等价
psigamma(2, 2)
```

© 科学出版社


```
## [1] -0.4041138
```

```
# 注意与下面这个例子比较
dx2x <- deriv(~ x^3, "x")
eval(dx2x)
```

```
## [1] 8
## attr(,"gradient")
##       x
## [1,] 12
```

B.28.3 贝塔函数

$B(a,b) = \Gamma(a)\Gamma(b)/\Gamma(a+b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt$

```
beta(1, 1)
```

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

```
beta(2, 3)
```

```
## [1] 0.08333333
```

```
beta2 <- function(t, a, b) {
  t^(a - 1) * (1 - t)^(b - 1)
}
integrate(beta2, lower = 0, upper = 1, a = 2, b = 3)
```

```
## 0.08333333 with absolute error < 9.3e-16
```

B.28.4 贝塞尔函数

```
besselI(x, nu, expon.scaled = FALSE) # 修正的第一类
besselK(x, nu, expon.scaled = FALSE) # 修正的第二类
besselJ(x, nu) # 第一类
besselY(x, nu) # 第二类
```

- ν 贝塞尔函数的阶，可以是分数
- expon.scaled 是否使用指数表示

```
nus <- c(0:5, 10, 20)
x <- seq(0, 4, length.out = 501)
plot(x, x,
  ylim = c(0, 6), ylab = "", type = "n",
  main = "Bessel Functions  I_nu(x)"
)
for (nu in nus) lines(x, besselI(x, nu = nu), col = nu + 2)
legend(0, 6, legend = paste("nu=", nus), col = nus + 2, lwd = 1)
```

© 黄湘云

介绍复数矩阵的计算，矩阵的指数计算，介绍一点分形

```
# 考虑用 gganimate 实现，去掉 caTools 依赖
library(caTools)
jet.colors <- colorRampPalette(c(
  "green", "blue", "red", "cyan", "#7FFF7F",
  "yellow", "#FF7F00", "red", "#7F0000"
))
m <- 1000 # define size
C <- complex(
  real = rep(seq(-1.8, 0.6, length.out = m), each = m),
  imag = rep(seq(-1.2, 1.2, length.out = m), m)
)
C <- matrix(C, m, m) # reshape as square matrix of complex numbers
Z <- 0 # initialize Z to zero
X <- array(0, c(m, m, 20)) # initialize output 3D array
for (k in 1:20) { # loop with 20 iterations
  Z <- Z^2 + C # the central difference equation
  X[, , k] <- exp(-abs(Z)) # capture results
}
write.gif(X, "Mandelbrot.gif", col = jet.colors, delay = 100)
```

© 草稿

附录 C　符号计算

相比于数值计算，符号计算可以无限精度，包括微分、积分运法，求解线性、非线性方程（组），常微分、偏微分方程（组）等，R 自带几个函数如 `deriv()`、`D()` 等可以做一些简单的微分运算，扩展包 Ryacas 提供 Yacas 核心计算引擎，symengine 引入 C++ 符号计算库SymEngine，相比于 **Ryacas**[Andersen and Højsgaard, 2019]，**symengine** 不会和 Base R 函数冲突。Python 的符号计算模块 sympy [Meurer et al., 2017] 不仅支持简单的四则运算，还支持微分、积分、解方程等，详见官方文档 https://sympy.org/。

16 年在统计之都灌水符号计算与 R 语言，相应的 Rmd 源文件放在Github上。

```
# 多元函数求偏导
ft <- deriv(expression(sin(x1) + sin(x2) + cos(3 * x1 * x2) + x1^2 + x2^2),
  namevec = c("x1", "x2"), function.arg = TRUE
)
# 隐函数求偏导
deriv(y ~ sin(cos(x) * y), namevec = c("x","y"), function.arg = TRUE)
```

```
## function (x, y)
## {
##     .expr1 <- cos(x)
##     .expr2 <- .expr1 * y
##     .expr4 <- cos(.expr2)
##     .value <- sin(.expr2)
##     .grad <- array(0, c(length(.value), 2L), list(NULL, c("x",
##         "y")))
##     .grad[, "x"] <- -(.expr4 * (sin(x) * y))
##     .grad[, "y"] <- .expr4 * .expr1
##     attr(.value, "gradient") <- .grad
##     .value
## }
```

下面以标准正态分布的密度函数为例，

```
NormDensity <- expression(1 / sqrt(2 * pi) * exp(-x^2 / 2))
# 递归的方法求高阶倒数
DD <- function(expr, name, order = 1) {
  if (order < 1) {
    stop("'order' must be >= 1")
  }
  if (order == 1) {
```

© 基础统计

```
    D(expr, name)
  } else {
    DD(D(expr, name), name, order - 1)
  }
}
# 计算三阶导数
DD(NormDensity, "x", 3)
```

```
## 1/sqrt(2 * pi) * (exp(-x^2/2) * (2 * x/2) * (2/2) + ((exp(-x^2/2) *
##     (2/2) - exp(-x^2/2) * (2 * x/2) * (2 * x/2)) * (2 * x/2) +
##     exp(-x^2/2) * (2 * x/2) * (2/2)))
```

Deriv 可以将 R 表达式简化

```
library(Deriv)
Simplify(DD(NormDensity, "x", 3))
```

```
## x * (3 - x^2) * exp(-(x^2/2))/sqrt(2 * pi)
```

即 $x(3-x^2)\mathrm{e}^{-x^2/2}/\sqrt{2\pi}$，eval() 将表达式转为函数，代入数值运算。

$$\tau(x) = \frac{(-1)^{j-1}}{\sqrt{j!}}\phi^{(j)}(x)$$

```
Tetrachoric <- function(x, j) {
  (-1)^(j - 1) / sqrt(factorial(j)) * eval(DD(NormDensity, "x", j))
}
Tetrachoric(2, 3)
```

```
## [1] -0.04408344
```

表达式转函数

```
ExpToFun<-function(x) eval(Simplify(DD(NormDensity, "x", 4)))
ExpToFun(2)
```

```
## [1] -0.2699548
```

函数求积分

```
integrate(ExpToFun, 0, pi)
```

```
## -0.06192048 with absolute error < 5.8e-12
```

对函数求微分

```
fun <- function(x) x * pi / sqrt(x)
# D(body(fun),'x')
Simplify(D(body(fun), "x"))
```

```
## 0.5 * pi/sqrt(x)
```

```
Dfun <- function(x) {
}
```

© 黄湘云

图 C.1: Tetrachoric 函数

统计出版社 ©

```
body(Dfun) <- Simplify(D(body(fun), "x"))
Dfun
```

```
## function (x)
## 0.5 * pi/sqrt(x)
```

```
Dfun(4)
```

```
## [1] 0.7853982
```

下面简单介绍 symengine 的符号计算能力

```
library(symengine)
# 声明几个符号变量
use_vars(x, y, z)
# 表达式展开
expr <- (x + y + z) ^ 2L - 42L
expand(expr)
```

```
## (Add)    -42 + 2*x*y + 2*x*z + 2*y*z + x^2 + y^2 + z^2
```

变量替换

```
a <- S("a")
# z 用 a 替换
expr <- subs(expr, z, a)
# y 用 x^2 替换
expr <- subs(expr, y, x^2L)
expr
```

```
## (Add)    -42 + (a + x + x^2)^2
```

表达式求 2 阶偏导

```
d1_expr <- DD(expr, "x", 2)
expand(d1_expr)
```

```
## (Add)    2 + 4*a + 12*x + 12*x^2
```

求解带参数 a 的一元二次方程

```
solutions <- solve(d1_expr, "x")
solutions
```

```
## VecBasic of length 2
## V( -1/2 + (-1/2)*sqrt(1 + (-1/3)*(2 + 4*a)), -1/2 + (1/2)*sqrt(1 + (-1/3)*(2 + 4*a)) )
```

©黄湘云

附录 D　混合编程

R 语言 [Ihaka and Gentleman, 1996] 是一个统计计算和绘图的环境，以下各个节不介绍具体 R 包函数用法和参数设置，重点在历史发展趋势脉络，详细介绍去见《现代统计图形》的相应章节。R 语言的目标在于统计计算和绘图，设计优势在数据结构、图形语法、动态文档和交互图形

D.1　函数源码

flow 包可以将函数调用的过程以流程图的方式呈现，代码结构一目了然，快速理清源代码

```
remotes::install_github('moodymudskipper/funflow')
funflow::view_flow('median.default')
```

```
methods(predict)
```

```
##  [1] predict.ar*                predict.Arima*
##  [3] predict.arima0*            predict.glm
##  [5] predict.HoltWinters*       predict.lm
##  [7] predict.loess*             predict.mlm*
##  [9] predict.nls*               predict.poly*
## [11] predict.ppr*               predict.prcomp*
## [13] predict.princomp*          predict.smooth.spline*
## [15] predict.smooth.spline.fit* predict.StructTS*
## see '?methods' for accessing help and source code
```

stats 包里找不到这个函数

```
ls("package:stats", all.names = TRUE, pattern = "predict.poly")
```

```
## character(0)
```

```
predict.poly
```

```
## Error in eval(expr, envir, enclos): object 'predict.poly' not found
```

可见函数 `predict.poly()` 默认没有导出

```
stats:::predict.poly
```

```
## function (object, newdata, ...)
## {
##     if (missing(newdata))
```

© 北京大学出版社

```
##         object
##     else if (is.null(attr(object, "coefs")))
##         poly(newdata, degree = max(attr(object, "degree")), raw = TRUE,
##             simple = TRUE)
##     else poly(newdata, degree = max(attr(object, "degree")),
##         coefs = attr(object, "coefs"), simple = TRUE)
## }
## <bytecode: 0x5642c74bc6f0>
## <environment: namespace:stats>
```

或者

```
getAnywhere(predict.poly)
```

```
## A single object matching 'predict.poly' was found
## It was found in the following places
##   registered S3 method for predict from namespace stats
##   namespace:stats
## with value
##
## function (object, newdata, ...)
## {
##     if (missing(newdata))
##         object
##     else if (is.null(attr(object, "coefs")))
##         poly(newdata, degree = max(attr(object, "degree")), raw = TRUE,
##             simple = TRUE)
##     else poly(newdata, degree = max(attr(object, "degree")),
##         coefs = attr(object, "coefs"), simple = TRUE)
## }
## <bytecode: 0x5642c74bc6f0>
## <environment: namespace:stats>
```

```
getAnywhere("predict.poly")$where
```

```
## [1] "registered S3 method for predict from namespace stats"
## [2] "namespace:stats"
```

函数参数个数

```
names(formals(read.table))
```

```
##  [1] "file"             "header"           "sep"              "quote"
##  [5] "dec"              "numerals"         "row.names"        "col.names"
##  [9] "as.is"            "na.strings"       "colClasses"       "nrows"
## [13] "skip"             "check.names"      "fill"             "strip.white"
## [17] "blank.lines.skip" "comment.char"     "allowEscapes"     "flush"
## [21] "stringsAsFactors" "fileEncoding"     "encoding"         "text"
## [25] "skipNul"
```

© 黄湘云

D.2 命名约定

R 语言当前的命名状态 https://journal.r-project.org/archive/2012-2/RJournal_2012-2_Baaaath.pdf 和 https://essentials.togaware.com/StyleO.pdf

R 与不同的编程语言如何交互

D.3 R 与 JavaScripts

```
library(htmlwidgets)
```

D.4 R 与 Python

R 包 knitr 和 reticulate 支持 R Markdown 文档中嵌入 Python 代码块，reticulate 包还支持 Python 和 R 之间的数据对象通信交流。

```
library(reticulate)
```

如图 D.1 所示，在 R Markdown 中执行 Python 绘图代码，并且将图形插入文档。

```
import matplotlib.pyplot as plt
plt.switch_backend('agg')

plt.plot([0, 2, 1, 4])
plt.show()
```

图 D.1: Python 图形

© 统计出版社

D.5 R 与 C

knitr 支持在 R Markdown 中嵌入 C 语言代码

```
void useC(int *i){
  i[0] = 11;
}
```

```
## make[1]: Entering directory '/home/runner/work/masr/masr'
## gcc -I"/opt/R/4.1.3/lib/R/include" -DNDEBUG  -I/usr/local/include  -fpic  -g -O2  -c c26140151d31ce.c -o c2
## gcc -shared -L/opt/R/4.1.3/lib/R/lib -L/usr/local/lib -o c26140151d31ce.so c26140151d31ce.o -L/opt/R/4.1.3
## make[1]: Leaving directory '/home/runner/work/masr/masr'
```

```
a <- rep(2,10)
out <- .C("useC", b = as.integer(a))
out
```

```
## $b
##  [1] 11  2  2  2  2  2  2  2  2  2
```

```
out$b
```

```
##  [1] 11  2  2  2  2  2  2  2  2  2
```

一步一步地命令行操作

```
R CMD SHLIB useC1.c
```

```
dyn.load("useC1.dll")
a <- rep(2,10)
out <- .C("useC", b = as.integer(a))
out$b
```

D.6 R 与 C++

Dirk Eddelbuettel 是 Rcpp 的核心开发者。

- Dirk Eddelbuettel celebRtion 2020, Copenhagen, Denmark Introduction to Rcpp: from simple examples to machine learning
- Online Tutorial for useR! 2020 Seamless R and C++ Introduction with Rcpp 视频 https://vimeo.com/438283959
- James Balamuta unofficial rcpp api documentation https://github.com/coatless/rcpp-api
- Rcpp for everyone https://github.com/teuder/rcpp4everyone_en
- 课程 Foundations of Data Science

```
library(Rcpp)
```

© 黄湘云

D.7 R 与 LaTeX

tikzDevice 包将 LaTeX 公式和绘图系统 TikZ 引入 R 语言生态，贡献在于提供更加漂亮的公式输出，对图形进行后期布局排版加工，达到设计师出品的质量水平。图 D.2 展示了复杂的 TeX 生态系统，R 语言只是取其精华，使用 TikZ 绘制。

```
\begin{tikzpicture}
  \path [
    mindmap,
    text = white,
    level 1 concept/.append style =
      {font=\Large\bfseries\sffamily, sibling angle=90, level distance=125},
    level 2 concept/.append style =
      {font=\normalsize\bfseries\sffamily},
    level 3 concept/.append style =
      {font=\small\bfseries\sffamily},
    tex/.style     = {concept, ball color=blue,
      font=\Huge\bfseries},
    engines/.style = {concept, ball color=green!50!black},
    formats/.style = {concept, ball color=purple!50!black},
    systems/.style = {concept, ball color=red!90!black},
    editors/.style = {concept, ball color=orange!90!black}
  ]
  node [tex] {\TeX} [clockwise from=0]
    child[concept color=green!50!black, nodes={engines}] {
      node {Engines} [clockwise from=90]
        child { node {\TeX} }
        child { node {pdf\TeX} }
        child { node {XeTeX} }
        child { node {Lua\TeX} }}
    child [concept color=purple, nodes={formats}] {
      node {Formats} [clockwise from=300]
        child { node {\LaTeX} }
        child { node {Con\TeX t} }}
    child [concept color=red, nodes={systems}] {
      node {Systems} [clockwise from=210]
        child { node {\TeX Live} [clockwise from=300]
          child { node {Mac \TeX} }}
        child { node {MiK\TeX} [clockwise from=60]
          child { node {Pro \TeX t} }}}
    child [concept color=orange, nodes={editors}] {
      node {Editors} [clockwise from=180]
        child { node {WinEdt} }
        child { node {\TeX works} }
        child { node {\TeX studio} }
```

©韦泽

```
        child { node {\TeX maker} }};
\end{tikzpicture}
```

图 D.2: TeX 系统

D.8 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
```

© 黄湘云


```
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] Rcpp_1.0.8.3      reticulate_1.24   htmlwidgets_1.5.4 shiny_1.7.1
## [5] magrittr_2.0.3
##
## loaded via a namespace (and not attached):
##  [1] knitr_1.38       sysfonts_0.8.8   lattice_0.20-45  xtable_1.8-4
##  [5] R6_2.5.1         rlang_1.0.2      fastmap_1.1.0    stringr_1.4.0
##  [9] tools_4.1.3      grid_4.1.3       xfun_0.30        png_0.1-7
## [13] tinytex_0.38     cli_3.2.0        htmltools_0.5.2  ellipsis_0.3.2
## [17] yaml_2.3.5       digest_0.6.29    lifecycle_1.0.1  bookdown_0.25
## [21] Matrix_1.4-1     later_1.3.0      promises_1.2.0.1 curl_4.3.2
## [25] evaluate_0.15    mime_0.12        rmarkdown_2.13   stringi_1.7.6
## [29] compiler_4.1.3   jsonlite_1.8.0   httpuv_1.6.5
```

© 黄湘云

附录 E　面向对象编程

进入这一章的读者都是对编程感兴趣的读者，希望在工程能力上有所提升的读者。那么最重要的是：

> Code should be written to minimize the time it would take for someone else to understand it.
>
> — The Art of Readable Code, Boswell, D. / Foucher, T.

代码可读性，代码复用性，代码维护性，代码扩展性，代码简洁性，代码高效性，代码容错性，我们共勉吧！如果读者已投身商业公司，应当以完成任务为第一，这自不必说！

E.1　环境

```
environment(fun = NULL)
environment(fun) <- value

is.environment(x)

.GlobalEnv
globalenv()
.BaseNamespaceEnv

emptyenv()
baseenv()

new.env(hash = TRUE, parent = parent.frame(), size = 29L)

parent.env(env)
parent.env(env) <- value

environmentName(env)

env.profile(env)
```

仅供试读

E.2 引用

```
get(x, pos = -1, envir = as.environment(pos), mode = "any",
    inherits = TRUE)

mget(x, envir = as.environment(-1), mode = "any", ifnotfound,
     inherits = FALSE)

dynGet(x, ifnotfound = , minframe = 1L, inherits = FALSE)
```

```
get Return the Value of a Named Object

exists Is an Object Defined?

exists(x, where = -1, envir = , frame, mode = "any",
       inherits = TRUE)

get0(x, envir = pos.to.env(-1L), mode = "any", inherits = TRUE,
     ifnotfound = NULL)
```

E.3 调用栈

Functions to Access the Function Call Stack

```
sys.call(which = 0)
sys.frame(which = 0)
sys.nframe()
sys.function(which = 0)
sys.parent(n = 1)

sys.calls()
sys.frames()
sys.parents()
sys.on.exit()
sys.status()
parent.frame(n = 1)

sys.source Parse and Evaluate Expressions from a File
```

E.4 闭包

An illustration of lexical scoping.

```
demo(scoping)
```

E.5 递归

Using recursion for adaptive integration

```
demo(recursion)
```

斐波那契数列

```
# 递归 Recall
fibonacci <- function(n) {
  if (n <= 2) {
    if (n >= 0) 1 else 0
  } else {
    Recall(n - 1) + Recall(n - 2)
  }
}
fibonacci(10) # 55
```

```
## [1] 55
```

E.6 异常

异常捕获和处理

```
demo(error.catching)
```

E.7 对象

判断对象类型

```
demo(is.things)
```

E.8 泛型

> I'd like to prefix all these solutions with 'Here's how to do it, but don't actually do it you crazy fool'. It's on a par with redefining pi, or redefining '+'. And then redefining '<-'. These techniques have their proper place, and that would be in the currently non-existent obfuscated R contest. No, the R-ish (iRish?) way is to index vectors from 1. That's what the R gods intended!
>
> — Barry Rowlingson [1]

[1] https://stat.ethz.ch/pipermail/r-help/2004-March/048688.html

© 黄湘云

如果要让下标从 0 开始的话，我们需要在现有的向量类型 vector 上定义新的向量类型 vector0，在其上并且实现索引运算 [和赋值修改元素的运算 [<-

```
# https://stat.ethz.ch/pipermail/r-help/2004-March/048682.html
as.vector0 <- function(x) structure(x, class = "vector0") # 创建一种新的数据结构 vector0
as.vector.vector0 <- function(x) unclass(x)
"[.vector0" <- function(x, i) as.vector0(as.vector.vector0(x)[i + 1]) # 索引操作
"[<-.vector0" <- function(x, i, value) { # 赋值操作
  x <- as.vector.vector0(x)
  x[i + 1] <- value
  as.vector0(x)
}
print.vector0 <- function(x) print(as.vector.vector0(x)) # 实现 print 方法
```

举个例子看看

```
1:10 # 是一个内置的现有向量类型 vector
```

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

```
x <- as.vector0(1:10) # 转化为新建的 vector0 类型
x[0:4] <- 100 * x[0:4] # 对 x 的元素替换修改
x
```

```
##  [1] 100 200 300 400 500   6   7   8   9  10
```

第三方 R 包大大扩展了 Base R 函数 `plot()` 的功能，比如 **mgcv** ，**nlme** 包和 **lattice** 包等，表 E.1 列出当前环境下，`plot()` 绘图方法。

E.9 除虫

Debugging with RStudio

E.10 性能

E.11 质量

Github Action 提供的测试环境支持单元测试 testthat、静态代码检查 lintr、覆盖测试 covr、集成测试 Travis-CI、集成部署 Netlify 等一系列代码检查，还有额外的辅助工具，见 Github Action 工具合集，相关学习材料见快速参考手册 https://github.com/github/actions-cheat-sheet PDF 版本，以创建 R 包为例，展示工程开发的流程 https://mdneuzerling.com/post/data-science-workflows/

标准计算和非标准计算 Standard and non-standard evaluation in R https://www.brodieg.com/2020/05/05/on-nse/

© 黄湘云

表 E.1: 泛型函数

A	B	C
plot.acf	plot.HoltWinters	plot.profile.nls
plot.ACF	plot.intervals.lmList	plot.ranef.lme
plot.augPred	plot.isoreg	plot.ranef.lmList
plot.compareFits	plot.jam	plot.raster
plot.data.frame	plot.lm	plot.shingle
plot.decomposed.ts	plot.lme	plot.simulate.lme
plot.default	plot.lmList	plot.spec
plot.dendrogram	plot.medpolish	plot.spline
plot.density	plot.mlm	plot.stepfun
plot.ecdf	plot.nffGroupedData	plot.stl
plot.factor	plot.nfnGroupedData	plot.table
plot.formula	plot.nls	plot.trellis
plot.function	plot.nmGroupedData	plot.ts
plot.gam	plot.pdMat	plot.tskernel
plot.gls	plot.ppr	plot.TukeyHSD
plot.hclust	plot.prcomp	plot.Variogram
plot.histogram	plot.princomp	plot.xyVector

©黄湘云

附录 F　文件操作

考虑添加 Shell 下的命令实现，参考 命令行的艺术

```
library(magrittr) # 提供管道命令 %>%
```

fs 由 Jim Hester 开发，提供文件系统操作的统一接口，相比于 R 默认的文件系统的操作函数有显而易见的优点，详情请看 https://fs.r-lib.org/

对于文件操作，Jim Hester 开发了 fs 包 目的是统一文件操作的命令，由于时间和历史原因，R 内置的文件操作函数的命名很不统一，如 `path.expand()` 和 `normalizePath()`，`Sys.chmod()` 和 `file.access()` 等

```
# 加载 R 包
library(fs)
```

F.1　查看文件

文件夹只包含文件，目录既包含文件又包含文件夹，`list.dirs` 列出目录或文件夹，`list.files` 列出文件或文件夹

- `list.dirs(path = ".", full.names = TRUE, recursive = TRUE)`
  - path: 指定完整路径名，默认使用当前路径 `getwd()`
  - full.names: TRUE 返回相对路径，FALSE 返回目录的名称
  - recursive: 是否递归的方式列出目录，如果是的话，目录下的子目录也会列出

```
# list.dirs(path = '.', full.names = TRUE, recursive = TRUE)
list.dirs(path = '.', full.names = TRUE, recursive = FALSE)
```

```
##  [1] "./_book"                          "./_bookdown_files"
##  [3] "./.git"                           "./.github"
##  [5] "./case-study_cache"               "./case-study_files"
##  [7] "./code"                           "./dashboard"
##  [9] "./data"                           "./data-manipulation_files"
## [11] "./data-transportation_files"      "./data-visualization_cache"
....
```

```
list.dirs(path = '.', full.names = FALSE, recursive = FALSE)
```

```
##  [1] "_book"                            "_bookdown_files"
##  [3] ".git"                             ".github"
##  [5] "case-study_cache"                 "case-study_files"
```

© 黄湘云

```
##  [7] "code"                          "dashboard"
##  [9] "data"                          "data-manipulation_files"
## [11] "data-transportation_files"     "data-visualization_cache"
....
```

- `list.files(path = ".", pattern = NULL, all.files = FALSE, full.names = FALSE, recursive = FALSE,ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)`

  是否递归的方式列出目录，如果是的话，目录下的子目录也会列出

  - path: 指定完整路径名，默认使用当前路径 `getwd()`
  - full.names: TRUE 返回相对路径，FALSE 返回目录的名称
  - recursive: 是否递归的方式列出目录，如果是的话，目录下的子目录也会列出

- `file.show(..., header = rep("", nfiles), title = "R Information", delete.file = FALSE, pager = getOption("pager"),encoding = "")`

  打开文件内容，`file.show` 会在 R 终端中新开一个窗口显示文件

```r
rinternals <- file.path(R.home("include"), "Rinternals.h")
# file.show(rinternals)
```

- `file.info(..., extra_cols = TRUE)`

  获取文件信息，此外 `file.mode(...)` 、`file.mtime(...)` 和 `file.size(...)` 分别表示文件的读写权限，修改时间和文件大小。

```r
file.info(rinternals)
```

```
##                                          size isdir mode               mtime
## /opt/R/4.1.3/lib/R/include/Rinternals.h 63180 FALSE  644 2022-03-23 00:00:20
##                                                        ctime               atime
## /opt/R/4.1.3/lib/R/include/Rinternals.h 2022-04-13 04:10:48 2022-04-13 04:19:32
##                                         uid gid uname grname
## /opt/R/4.1.3/lib/R/include/Rinternals.h   0   0  root   root
```

```r
file.mode(rinternals)
```

```
## [1] "644"
```

```r
file.mtime(rinternals)
```

```
## [1] "2022-03-23 00:00:20 UTC"
```

```r
file.size(rinternals)
```

```
## [1] 63180
```

```r
# 查看当前目录的权限
file.info(".")
```

```
##    size isdir mode               mtime               ctime               atime
## . 16384  TRUE  755 2022-04-13 07:33:35 2022-04-13 07:33:35 2022-04-13 07:33:36
##    uid gid  uname grname
## . 1001 121 runner docker
```

©黄湘云


```
# 查看指定目录权限
file.info("./_book/")
```

```
##           size isdir mode               mtime               ctime
## ./_book/ 12288  TRUE  755 2022-04-13 07:25:14 2022-04-13 07:25:14
##                        atime  uid gid  uname grname
## ./_book/ 2022-04-13 07:25:20 1001 121 runner docker
```

- `file.access(names, mode = 0)`

  文件是否可以被访问，第二个参数 mode 一共有四种取值 0，1，2，4，分别表示文件的存在性，可执行，可写和可读四种，返回值 0 表示成功，返回值 -1 表示失败。

```
file.access(rinternals,mode = 0)
```

```
## /opt/R/4.1.3/lib/R/include/Rinternals.h
##                                       0
```

```
file.access(rinternals,mode = 1)
```

```
## /opt/R/4.1.3/lib/R/include/Rinternals.h
##                                      -1
```

```
file.access(rinternals,mode = 2)
```

```
## /opt/R/4.1.3/lib/R/include/Rinternals.h
##                                      -1
```

```
file.access(rinternals,mode = 4)
```

```
## /opt/R/4.1.3/lib/R/include/Rinternals.h
##                                       0
```

- `dir(path = ".", pattern = NULL, all.files = FALSE, full.names = FALSE, recursive = FALSE, ignore.case = FALSE, include.dirs = FALSE, no.. = FALSE)`

  查看目录，首先看看和目录操作有关的函数列表

```
apropos("^dir.")
```

```
##  [1] "dir_copy"   "dir_create" "dir_delete" "dir_exists" "dir_info"
##  [6] "dir_ls"     "dir_map"    "dir_tree"   "dir_walk"   "dir.create"
## [11] "dir.exists" "dirname"
```

显而易见，`dir.create` 和 `dir.exists` 分别是创建目录和查看目录的存在性。`dirname` 和 `basename` 是一对函数用来操作文件路径。以当前目录/home/runner/work/masr/masr 为例，`dirname(getwd())` 返回 /home/runner/work/masr 而 `basename(getwd())` 返回 masr。对于文件路径而言，`dirname(rinternals)` 返回文件所在的目录/opt/R/4.1.3/lib/R/include，`basename(rinternals)` 返回文件名 Rinternals.h。`dir` 函数查看指定路径或目录下的文件，支持以模式匹配和递归的方式查找目录下的文件

```
# 当前目录下的子目录和文件
dir()
```

```
##   [1] "_book"
```

© 黄湘云


```
##   [2] "_bookdown_files"
##   [3] "_bookdown.yml"
##   [4] "_build.sh"
##   [5] "_common.R"
##   [6] "_deploy-book.R"
....
```

```
# 查看指定目录的子目录和文件
dir(path = "./")
```

```
##   [1] "_book"
##   [2] "_bookdown_files"
##   [3] "_bookdown.yml"
##   [4] "_build.sh"
##   [5] "_common.R"
##   [6] "_deploy-book.R"
....
```

```
# 只列出以字母R开头的子目录和文件
dir(path = "./", pattern = "^R")
```

```
## [1] "README.md"
```

```
# 列出目录下所有目录和文件，包括隐藏文件
dir(path = "./", all.files = TRUE)
```

```
##   [1] "_book"
##   [2] "_bookdown_files"
##   [3] "_bookdown.yml"
##   [4] "_build.sh"
##   [5] "_common.R"
##   [6] "_deploy-book.R"
....
```

```
# 支持正则表达式
dir(pattern = '^[A-Z]+[.]txt$', full.names=TRUE, system.file('doc', 'SuiteSparse', package='Matrix'))
```

```
## [1] "/home/runner/work/_temp/Library/Matrix/doc/SuiteSparse/AMD.txt"
## [2] "/home/runner/work/_temp/Library/Matrix/doc/SuiteSparse/CHOLMOD.txt"
## [3] "/home/runner/work/_temp/Library/Matrix/doc/SuiteSparse/COLAMD.txt"
## [4] "/home/runner/work/_temp/Library/Matrix/doc/SuiteSparse/SPQR.txt"
```

```
# 在临时目录下递归创建一个目录
dir.create(paste0(tempdir(), "/_book/tmp"), recursive = TRUE)
```

查看当前目录下的文件和文件夹 `tree -L 2 .` 或者 `ls -l .`

© 王菲菲

F.2 操作文件

实现文件增删改查的函数如下

```
apropos("^file.")
```

```
##  [1] "file_access"    "file_chmod"     "file_chown"     "file_copy"
##  [5] "file_create"    "file_delete"    "file_exists"    "file_info"
##  [9] "file_move"      "file_show"      "file_size"      "file_temp"
## [13] "file_temp_pop"  "file_temp_push" "file_test"      "file_touch"
## [17] "file.access"    "file.append"    "file.choose"    "file.copy"
## [21] "file.create"    "file.edit"      "file.exists"    "file.info"
## [25] "file.link"      "file.mode"      "file.mtime"     "file.path"
## [29] "file.remove"    "file.rename"    "file.show"      "file.size"
## [33] "file.symlink"   "fileSnapshot"
```

1. `file.create(..., showWarnings = TRUE)`

   创建/删除文件，检查文件的存在性

   ```
   file.create('demo.txt')
   ```

   ```
   ## [1] TRUE
   ```

   ```
   file.exists('demo.txt')
   ```

   ```
   ## [1] TRUE
   ```

   ```
   file.remove('demo.txt')
   ```

   ```
   ## [1] TRUE
   ```

   ```
   file.exists('demo.txt')
   ```

   ```
   ## [1] FALSE
   ```

2. `file.rename(from, to)` 文件重命名

   ```
   file.create('demo.txt')
   ```

   ```
   ## [1] TRUE
   ```

   ```
   file.rename(from = 'demo.txt', to = 'tmp.txt')
   ```

   ```
   ## [1] TRUE
   ```

   ```
   file.exists('tmp.txt')
   ```

   ```
   ## [1] TRUE
   ```

3. `file.append(file1, file2)` 追加文件 `file2` 的内容到文件 `file1` 上

   ```
   if(!dir.exists(paths = 'data/')) dir.create(path = 'data/')
   # 创建两个临时文件
   # file.create(c('data/tmp1.md','data/tmp2.md'))
   # 写入内容
   cat("AAA\n", file = 'data/tmp1.md')
   ```

© 黄湘云


```
cat("BBB\n", file = 'data/tmp2.md')
# 追加文件
file.append(file1 = 'data/tmp1.md', file2 = 'data/tmp2.md')
```

```
## [1] TRUE
```

```
# 展示文件内容
readLines('data/tmp1.md')
```

```
## [1] "AAA" "BBB"
```

4. `file.copy(from, to, overwrite = recursive, recursive = FALSE,copy.mode = TRUE, copy.date = FALSE)` 复制文件，参考 https://blog.csdn.net/wzj_110/article/details/86497860

```
file.copy(from = 'Makefile', to = 'data/Makefile')
```

```
## [1] FALSE
```

5. `file.symlink(from, to)` 创建符号链接 `file.link(from, to)` 创建硬链接
6. `Sys.junction(from, to)` windows 平台上的函数，提供类似符号链接的功能
7. `Sys.readlink(paths)` 读取文件的符号链接（软链接）
8. `choose.files` 在 Windows 环境下交互式地选择一个或多个文件，所以该函数只运行于 Windows 环境

```
# 选择 zip 格式的压缩文件或其它
if (interactive())
    choose.files(filters = Filters[c("zip", "All"),])
```

`Filters` 参数传递一个矩阵，用来描述或标记 R 识别的文件类型，上面这个例子就能筛选出 zip 格式的文件

9. `download.file` 文件下载

```
download.file(url = 'https://mirrors.tuna.tsinghua.edu.cn/CRAN/src/base/R-latest.tar.gz',
              destfile = 'data/R-latest.tar.gz', method = 'auto')
```

F.3 压缩文件

tar 和 zip 是两种常见的压缩文件工具，具有免费和跨平台的特点，因此应用范围广[1]。R 内对应的压缩与解压缩命令是 `tar/untar`

```
tar(tarfile, files = NULL,
    compression = c("none", "gzip", "bzip2", "xz"),
    compression_level = 6, tar = Sys.getenv("tar"),
    extra_flags = "")
```

比较常用的压缩文件格式是 `.tar.gz` 和 `.tar.bz2`，将目录 `_book/`及其文件分别压缩成 `_book.tar.gz` 和 `_book.tar.bz2` 压缩包的名字可以任意取，后者压缩比率高。`.tar.xz` 的压缩比率最高，需要确保系统中

[1] https://github.com/libarchive/libarchive/wiki/FormatTar

© 黄湘云

安装了 gzip，bzip2 和 xz-utils 软件，R 软件自带的 tar 软件来自 Rtools[2]，我们可以通过设置系统环境变量 `Sys.setenv(tar="path/to/tar")` 指定外部 tar。`tar` 实际支持的压缩类型只有 `.tar.gz`[3]。`zip/unzip` 压缩与解压缩就不赘述了。

```
# 打包目录 _book
tar(tarfile = 'data/_book.tar', files = '_book', compression = 'none')
# 文件压缩成 _book.xz 格式
tar(tarfile = 'data/_book.tar.xz', files = 'data/_book', compression = 'xz')
# tar -cf data/_book.tar _book 然后 xz -z data/_book.tar.xz data/_book.tar
# 或者一次压缩到位 tar -Jcf data/_book.tar.xz _book/

# 解压 xz -d data/_book.tar.xz 再次解压 tar -xf data/_book.tar
# 或者一次解压 tar -Jxf data/_book.tar.xz

# 文件压缩成 _book.tar.gz 格式
# tar -czf data/_book.tar.gz _book
tar(tarfile = 'data/_book.tar.gz', files = '_book', compression = 'gzip')
# 解压 tar -xzf data/_book.tar.gz

# 文件压缩成 .tar.bz2 格式
# tar -cjf data/book2.tar.bz2 _book
tar(tarfile = 'data/_book.tar.bz2', files = '_book', compression = 'bzip2')
# 解压 tar -xjf data/book2.tar.bz2
```

```
untar(tarfile, files = NULL, list = FALSE, exdir = ".",
      compressed = NA, extras = NULL, verbose = FALSE,
      restore_times =  TRUE, tar = Sys.getenv("TAR"))
```

F.4 路径操作

环境变量算是路径操作

```
# 获取环境变量
Sys.getenv("PATH")
```

```
## [1] "/home/runner/.TinyTeX/bin/x86_64-linux:/home/linuxbrew/.linuxbrew/bin:/home/linuxbrew/.linuxb
```

```
# 设置环境变量 Windows
# Sys.setenv(R_GSCMD = "C:/Program Files/gs/gs9.26/bin/gswin64c.exe")
# 设置 pandoc 环境变量
pandoc_path <- Sys.getenv("RSTUDIO_PANDOC", NA)
if (Sys.which("pandoc") == "" && !is.na(pandoc_path)) {
  Sys.setenv(PATH = paste(
    Sys.getenv("PATH"), pandoc_path,
```

[2]继 Rtools35 之后，RTools40 主要为 R 3.6.0 准备的，包含有 GCC 8 及其它编译 R 包需要的工具包，详情请看的幻灯片

[3]https://github.com/rwinlib/utils

非洲出版社 ©

```
    sep = if (.Platform$OS.type == "unix") ":" else ";"
  ))
}
```

操作文件路径

1. `file.path` Construct Path to File

```
file.path('./_book')
```

```
## [1] "./_book"
```

2. `path.expand(path)` Expand File Paths

```
path.expand('./_book')
```

```
## [1] "./_book"
```

```
path.expand('~')
```

```
## [1] "/home/runner"
```

3. `normalizePath()` Express File Paths in Canonical Form

```
normalizePath('~')
```

```
## [1] "/home/runner"
```

```
normalizePath('./_book')
```

```
## [1] "/home/runner/work/masr/masr/_book"
```

4. `shortPathName(path)` 只在 Windows 下可用，Express File Paths in Short Form

```
cat(shortPathName(c(R.home(), tempdir())), sep = "\n")
```

5. `Sys.glob` Wildcard Expansion on File Paths

```
Sys.glob(file.path(R.home(), "library", "compiler", "R", "*.rdx"))
```

```
## [1] "/opt/R/4.1.3/lib/R/library/compiler/R/compiler.rdx"
```

F.5 查找文件

here 包用来查找你的文件，查找文件、可执行文件的完整路径、R 包

1. `Sys.which` Find Full Paths to Executables

```
Sys.which('pandoc')
```

```
##             pandoc
## "/usr/bin/pandoc"
```

2. `system.file` Find Names of R System Files

```
system.file('CITATION',package = 'base')
```

```
## [1] "/opt/R/4.1.3/lib/R/library/base/CITATION"
```

© 黄湘云

3. R.home

```
# R 安装目录
R.home()
```

```
## [1] "/opt/R/4.1.3/lib/R"
```

```
# R执行文件目录
R.home('bin')
```

```
## [1] "/opt/R/4.1.3/lib/R/bin"
```

```
# 配置文件目录
R.home('etc')
```

```
## [1] "/opt/R/4.1.3/lib/R/etc"
```

```
# R 基础扩展包存放目录
R.home('library')
```

```
## [1] "/opt/R/4.1.3/lib/R/library"
```

4. .libPaths() R 包存放的路径有哪些

```
.libPaths()
```

```
## [1] "/home/runner/work/_temp/Library" "/opt/R/4.1.3/lib/R/library"
```

5. find.package 查找 R 包所在目录

```
find.package(package = 'MASS')
```

```
## [1] "/home/runner/work/_temp/Library/MASS"
```

6. file.exist 检查文件是否存在

```
file.exists(paste(R.home('etc'),"Rprofile.site",sep = .Platform$file.sep))
```

```
## [1] FALSE
```

7. apropos 和 find 查找对象

```
apropos(what, where = FALSE, ignore.case = TRUE, mode = "any")
find(what, mode = "any", numeric = FALSE, simple.words = TRUE)
```

匹配含有 find 的函数

```
apropos("find")
```

```
##  [1] "find"                 "Find"                 "find.package"
##  [4] "findClass"            "findFunction"         "findInterval"
##  [7] "findLineNum"          "findMethod"           "findMethods"
## [10] "findMethodSignatures" "findPackageEnv"       "findRestart"
## [13] "findUnique"
```

问号 ? 加函数名搜索 R 软件内置函数的帮助文档，如 ?regrex。如果不知道具体的函数名，可采用关键词搜索，如

©黄湘云

```
help.search(keyword = "character", package = "base")
```

browseEnv 函数用来在浏览器中查看当前环境下，对象的列表，默认环境是 sys.frame()

F.6 文件权限

操作目录和文件的权限 Manipulation of Directories and File Permissions

1. dir.exists(paths) 检查目录是否存在

```
dir.exists(c('./_book','./book'))
```

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

2. dir.create(path, showWarnings = TRUE, recursive = FALSE, mode = "0777") 创建目录

```
dir.create('./_book/tmp')
```

```
## Warning in dir.create("./_book/tmp"): './_book/tmp' already exists
```

3. Sys.chmod(paths, mode = "0777", use_umask = TRUE) 修改权限

```
Sys.chmod('./_book/tmp')
```

4. Sys.umask(mode = NA)

F.7 区域设置

1. Sys.getlocale(category = "LC_ALL") 查看当前区域设置

```
Sys.getlocale(category = "LC_ALL")
```

```
## [1] "LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_COLLATE=en_US.UTF-8;LC_MONETARY=en_US
```

2. Sys.setlocale(category = "LC_ALL", locale = "") 设置区域

```
# 默认设置
Sys.setlocale(category = "LC_ALL", locale = "")
```

```
## [1] "LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_COLLATE=en_US.UTF-8;LC_MONETARY=en_US
```

```
# 保存当前区域设置
old <- Sys.getlocale()

Sys.setlocale("LC_MONETARY", locale = "")
```

```
## [1] "en_US.UTF-8"
```

```
Sys.localeconv()
```

```
##     decimal_point     thousands_sep          grouping   int_curr_symbol
##               "."                ""                ""            "USD "
##   currency_symbol mon_decimal_point mon_thousands_sep      mon_grouping
```

© 高等教育出版社


```
##              "$"              "."              ","       "\003\003"
##    positive_sign    negative_sign  int_frac_digits      frac_digits
##               ""              "-"              "2"              "2"
##    p_cs_precedes   p_sep_by_space    n_cs_precedes   n_sep_by_space
##              "1"              "0"              "1"              "0"
##      p_sign_posn      n_sign_posn
##              "1"              "1"
```

```
Sys.setlocale("LC_MONETARY", "de_AT")
```

```
## Warning in Sys.setlocale("LC_MONETARY", "de_AT"): OS reports request to set
## locale to "de_AT" cannot be honored
```

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

```
Sys.localeconv()
```

```
##    decimal_point    thousands_sep         grouping  int_curr_symbol
##              "."               ""               ""          "USD "
##  currency_symbol mon_decimal_point mon_thousands_sep     mon_grouping
##              "$"              "."              ","       "\003\003"
##    positive_sign    negative_sign  int_frac_digits      frac_digits
##               ""              "-"              "2"              "2"
##    p_cs_precedes   p_sep_by_space    n_cs_precedes   n_sep_by_space
##              "1"              "0"              "1"              "0"
##      p_sign_posn      n_sign_posn
##              "1"              "1"
```

```
# 恢复区域设置
Sys.setlocale(locale = old)
```

```
## Warning in Sys.setlocale(locale = old): OS reports request to set locale to
## "LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_COLLATE=en_US.UTF-8;LC_MONETARY=en_
## cannot be honored
```

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

3. `Sys.localeconv()` 当前区域设置下，数字和货币的表示

```
Sys.localeconv()
```

```
##    decimal_point    thousands_sep         grouping  int_curr_symbol
##              "."               ""               ""          "USD "
##  currency_symbol mon_decimal_point mon_thousands_sep     mon_grouping
##              "$"              "."              ","       "\003\003"
##    positive_sign    negative_sign  int_frac_digits      frac_digits
##               ""              "-"              "2"              "2"
##    p_cs_precedes   p_sep_by_space    n_cs_precedes   n_sep_by_space
##              "1"              "0"              "1"              "0"
##      p_sign_posn      n_sign_posn
##              "1"              "1"
```

© 基础

本地化信息

```
l10n_info()
```

```
## $MBCS
## [1] TRUE
##
## $`UTF-8`
## [1] TRUE
##
## $`Latin-1`
## [1] FALSE
##
## $codeset
## [1] "UTF-8"
```

F.8 进程管理

ps 包用来查询进程信息

- `Sys.getpid` 获取当前运行中的 R 控制台（会话）的进程 ID

```
Sys.getpid()
```

```
## [1] 156079
```

- `proc.time()` R 会话运行时间，常用于计算 R 程序在当前 R 控制台的运行时间

```
t1 <- proc.time()
tmp <- rnorm(1e6)
proc.time() - t1
```

```
##    user  system elapsed
##   0.057   0.008   0.065
```

- `system.time` 计算 R 表达式/程序块运行耗费的 CPU 时间

```
system.time({
  rnorm(1e6)
}, gcFirst = TRUE)
```

```
##    user  system elapsed
##   0.065   0.004   0.068
```

- `gc.time` 报告垃圾回收耗费的时间

```
gc.time()
```

```
## [1] 0 0 0 0 0
```

© 草稿

F.9 系统命令

system 和 system2 调用系统命令，推荐使用后者，它更灵活更便携。此外，Jeroen Ooms 开发的 sys 包可看作 base::system2 的替代品

```
system <- function(...) cat(base::system(..., intern = TRUE), sep = '\n')
system2 <- function(...) cat(base::system2(..., stdout = TRUE), sep = "\n")
```

```
system(command = "xelatex --version")
```

```
## XeTeX 3.141592653-2.6-0.999994 (TeX Live 2022)
## kpathsea version 6.3.4
## Copyright 2022 SIL International, Jonathan Kew and Khaled Hosny.
## There is NO warranty.  Redistribution of this software is
## covered by the terms of both the XeTeX copyright and
## the Lesser GNU General Public License.
## For more information about these matters, see the file
## named COPYING and the XeTeX source.
## Primary author of XeTeX: Jonathan Kew.
## Compiled with ICU version 70.1; using 70.1
## Compiled with zlib version 1.2.11; using 1.2.11
## Compiled with FreeType2 version 2.11.1; using 2.11.1
## Compiled with Graphite2 version 1.3.14; using 1.3.14
## Compiled with HarfBuzz version 3.4.0; using 3.4.0
## Compiled with libpng version 1.6.37; using 1.6.37
## Compiled with pplib version v2.05 less toxic i hope
## Compiled with fontconfig version 2.13.0; using 2.13.1
```

```
system2(command = 'pdflatex', args = '--version')
```

```
## pdfTeX 3.141592653-2.6-1.40.24 (TeX Live 2022)
## kpathsea version 6.3.4
## Copyright 2022 Han The Thanh (pdfTeX) et al.
## There is NO warranty.  Redistribution of this software is
## covered by the terms of both the pdfTeX copyright and
## the Lesser GNU General Public License.
## For more information about these matters, see the file
## named COPYING and the pdfTeX source.
## Primary author of pdfTeX: Han The Thanh (pdfTeX) et al.
## Compiled with libpng 1.6.37; using libpng 1.6.37
## Compiled with zlib 1.2.11; using zlib 1.2.11
## Compiled with xpdf version 4.03
```

F.10 时间管理

1. Sys.timezone 获取时区信息

©黄海广

```
Sys.timezone(location = TRUE)
```

```
## [1] "UTC"
```

2. Sys.time 系统时间，可以给定时区下，显示当前时间，精确到秒，返回数据类型为 POSIXct

```
# 此时美国洛杉矶时间
format(Sys.time(), tz = 'America/Los_Angeles', usetz = TRUE)
```

```
## [1] "2022-04-13 00:33:36 PDT"
```

```
# 此时加拿大东部时间
format(Sys.time(), tz = 'Canada/Eastern', usetz = TRUE)
```

```
## [1] "2022-04-13 03:33:36 EDT"
```

3. Sys.Date 显示当前时区下的日期，精确到日，返回数据类型为 date

```
Sys.Date()
```

```
## [1] "2022-04-13"
```

4. date 返回当前系统日期和时间，数据类型是字符串

```
date()
```

```
## [1] "Wed Apr 13 07:33:36 2022"
```

```
## 也可以这样表示
format(Sys.time(), "%a %b %d %H:%M:%S %Y")
```

```
## [1] "Wed Apr 13 07:33:36 2022"
```

5. as.POSIX* 是一个 Date-time 转换函数

```
as.POSIXlt(Sys.time(), "GMT") # the current time in GMT
```

```
## [1] "2022-04-13 07:33:36 GMT"
```

6. 时间计算

```
(z <- Sys.time())             # the current date, as class "POSIXct"
```

```
## [1] "2022-04-13 07:33:36 UTC"
```

```
Sys.time() - 3600             # an hour ago
```

```
## [1] "2022-04-13 06:33:36 UTC"
```

7. .leap.seconds 是内置的日期序列

```
.leap.seconds
```

```
##  [1] "1972-07-01 GMT" "1973-01-01 GMT" "1974-01-01 GMT" "1975-01-01 GMT"
##  [5] "1976-01-01 GMT" "1977-01-01 GMT" "1978-01-01 GMT" "1979-01-01 GMT"
##  [9] "1980-01-01 GMT" "1981-07-01 GMT" "1982-07-01 GMT" "1983-07-01 GMT"
## [13] "1985-07-01 GMT" "1988-01-01 GMT" "1990-01-01 GMT" "1991-01-01 GMT"
## [17] "1992-07-01 GMT" "1993-07-01 GMT" "1994-07-01 GMT" "1996-01-01 GMT"
```

© 黄湘云


```
## [21] "1997-07-01 GMT" "1999-01-01 GMT" "2006-01-01 GMT" "2009-01-01 GMT"
## [25] "2012-07-01 GMT" "2015-07-01 GMT" "2017-01-01 GMT"
```

计算日期对应的星期 weekdays，月 months 和季度 quarters

```
weekdays(.leap.seconds)
```

```
##  [1] "Saturday"  "Monday"    "Tuesday"   "Wednesday" "Thursday"  "Saturday"
##  [7] "Sunday"    "Monday"    "Tuesday"   "Wednesday" "Thursday"  "Friday"
## [13] "Monday"    "Friday"    "Monday"    "Tuesday"   "Wednesday" "Thursday"
## [19] "Friday"    "Monday"    "Tuesday"   "Friday"    "Sunday"    "Thursday"
## [25] "Sunday"    "Wednesday" "Sunday"
```

```
months(.leap.seconds)
```

```
##  [1] "July"    "January" "January" "January" "January" "January" "January"
##  [8] "January" "January" "July"    "July"    "July"    "July"    "January"
## [15] "January" "January" "July"    "July"    "July"    "January" "July"
## [22] "January" "January" "January" "July"    "July"    "January"
```

```
quarters(.leap.seconds)
```

```
##  [1] "Q3" "Q1" "Q1" "Q1" "Q1" "Q1" "Q1" "Q1" "Q1" "Q3" "Q3" "Q3" "Q3" "Q1" "Q1"
## [16] "Q1" "Q3" "Q3" "Q3" "Q1" "Q3" "Q1" "Q1" "Q1" "Q3" "Q3" "Q1"
```

8. Sys.setFileTime() 使用系统调用 system call 设置文件或目录的时间

```
# 修改时间前
file.info('./_common.R')
```

```
##             size isdir mode               mtime               ctime
## ./_common.R 3290 FALSE  644 2022-04-13 07:04:03 2022-04-13 07:04:03
##                           atime  uid gid  uname grname
## ./_common.R 2022-04-13 07:04:13 1001 121 runner docker
```

```
# 修改时间后，对比一下
Sys.setFileTime(path = './_common.R', time = Sys.time())
file.info('./_common.R')
```

```
##             size isdir mode               mtime               ctime
## ./_common.R 3290 FALSE  644 2022-04-13 07:33:36 2022-04-13 07:33:36
##                           atime  uid gid  uname grname
## ./_common.R 2022-04-13 07:33:36 1001 121 runner docker
```

9. strptime 用于字符串与 POSIXlt、POSIXct 类对象之间的转化，format 默认 tz = "" 且 usetz = TRUE

```
# 存放时区信息的数据库所在目录
list.files(file.path(R.home("share"), "zoneinfo"))
```

```
## character(0)
```

© 黄湘云


```
# 比较不同的打印方式
strptime(Sys.time(), format ="%Y-%m-%d %H:%M:%S", tz = "Asia/Taipei")
```

```
## [1] "2022-04-13 07:33:36 CST"
```

```
format(Sys.time(), format = "%Y-%m-%d %H:%M:%S") # 默认情形
```

```
## [1] "2022-04-13 07:33:36"
```

```
format(Sys.time(), format = "%Y-%m-%d %H:%M:%S", tz = "Asia/Taipei", usetz = TRUE)
```

```
## [1] "2022-04-13 15:33:36 CST"
```

10. 设置时区

```
Sys.timezone()
```

```
## [1] "UTC"
```

```
Sys.setenv(TZ = "Asia/Shanghai")
Sys.timezone()
```

```
## [1] "Asia/Shanghai"
```

全局修改，在文件 /opt/R/4.1.3/lib/R/etc/Rprofile.site 内添加 `Sys.setenv(TZ="Asia/Shanghai")`。局部修改，就是在本地 R 项目下，创建 `.Rprofile`，然后同样添加 `Sys.setenv(TZ="Asia/Shanghai")`。

F.11 R 包管理

相关的函数大致有

```
apropos('package')
```

```
##  [1] ".packages"                     ".packageStartupMessage"
##  [3] "$.package_version"             "as.package_version"
##  [5] "aspell_package_C_files"        "aspell_package_R_files"
##  [7] "aspell_package_Rd_files"       "aspell_package_vignettes"
##  [9] "available.packages"            "download.packages"
## [11] "find.package"                  "findPackageEnv"
## [13] "format.packageInfo"            "getPackageName"
## [15] "install.packages"              "installed.packages"
## [17] "is.package_version"            "make.packages.html"
## [19] "methodsPackageMetaName"        "new.packages"
....
```

1. `.packages(T)` 已安装的 R 包

```
.packages(T) %>% length()
```

```
## [1] 447
```

2. `available.packages` 查询可用的 R 包

©清华大学出版社


```
available.packages()[,"Package"] %>% head()
```

```
##           A3        aaSEA     AATtools       ABACUS   abbreviate       abbyyR
##         "A3"      "aaSEA"   "AATtools"     "ABACUS" "abbreviate"     "abbyyR"
```

查询 repos 的 R 包

```
rforge <- available.packages(repos = "https://r-forge.r-project.org/")
cran <- available.packages(repos = "https://mirrors.tuna.tsinghua.edu.cn/CRAN/")
setdiff(rforge[, "Package"], cran[, "Package"])
```

3. download.packages 下载 R 包

```
download.packages("Rbooks", destdir = "~/", repos = "https://r-forge.r-project.org/")
```

4. install.packages 安装 R 包

```
install.packages("rmarkdown")
```

5. installed.packages 已安装的 R 包

```
installed.packages(fields = c("Package","Version")) %>% head()
```

6. remove.packages 卸载/删除/移除已安装的 R 包

```
remove.packages('rmarkdown')
```

7. update.packages 更新已安装的 R 包

```
update.packages(ask = FALSE)
```

8. old.packages 查看过时/可更新的 R 包

```
old.packages() %>% head()
```

```
##           Package     LibPath                           Installed Built
## beanplot  "beanplot"  "/home/runner/work/_temp/Library" "1.2"     "4.1.3"
## blob      "blob"      "/home/runner/work/_temp/Library" "1.2.2"   "4.1.3"
## extrafont "extrafont" "/home/runner/work/_temp/Library" "0.17"    "4.1.3"
## formatR   "formatR"   "/home/runner/work/_temp/Library" "1.11"    "4.1.3"
## gifski    "gifski"    "/home/runner/work/_temp/Library" "1.4.3-1" "4.1.3"
## igraph    "igraph"    "/home/runner/work/_temp/Library" "1.2.11"  "4.1.3"
##           ReposVer  Repository
## beanplot  "1.3.1"   "https://cloud.r-project.org/src/contrib"
## blob      "1.2.3"   "https://cloud.r-project.org/src/contrib"
## extrafont "0.18"    "https://cloud.r-project.org/src/contrib"
## formatR   "1.12"    "https://cloud.r-project.org/src/contrib"
## gifski    "1.6.6-1" "https://cloud.r-project.org/src/contrib"
## igraph    "1.3.0"   "https://cloud.r-project.org/src/contrib"
```

9. new.packages 还没有安装的 R 包

```
new.packages() %>% head()
```

```
## [1] "A3"         "aaSEA"      "AATtools"   "ABACUS"     "abbreviate"
```

© 高等教育出版社

```
## [6] "abbyyR"
```

10. packageStatus 查看已安装的 R 包状态，可更新、可下载等

```
packageStatus()
```

```
## Number of installed packages:
##
##                                    ok upgrade unavailable
##   /home/runner/work/_temp/Library 395      17          12
##   /opt/R/4.1.3/lib/R/library       23       6           0
##
## Number of available packages (each package counted only once):
##
##                                            installed not installed
##   https://cloud.r-project.org/src/contrib       421         18580
```

11. packageDescription 查询 R 包描述信息

```
packageDescription('graphics')
```

```
## Package: graphics
## Version: 4.1.3
## Priority: base
## Title: The R Graphics Package
## Author: R Core Team and contributors worldwide
## Maintainer: R Core Team <do-use-Contact-address@r-project.org>
....
```

12. 查询 R 包的依赖关系

```
# rmarkdown 依赖的 R 包
tools::package_dependencies('rmarkdown', recursive = TRUE)
```

```
## $rmarkdown
##  [1] "bslib"     "evaluate"  "htmltools" "jquerylib" "jsonlite"  "knitr"
##  [7] "methods"   "stringr"   "tinytex"   "tools"     "utils"     "xfun"
## [13] "yaml"      "grDevices" "sass"      "rlang"     "digest"    "base64enc"
## [19] "fastmap"   "highr"     "glue"      "magrittr"  "stringi"   "stats"
## [25] "fs"        "R6"        "rappdirs"
```

```
# 依赖 rmarkdown 的 R 包
tools::dependsOnPkgs('rmarkdown', recursive = TRUE)
```

```
##  [1] "bookdown"       "flexdashboard"  "formattable"    "hrbrthemes"
##  [5] "kableExtra"     "prettydoc"      "reprex"         "tint"
##  [9] "packagemetrics" "tidyverse"
```

ggplot2 生态，仅列出以 gg 开头的 R 包

```
pdb <- available.packages()
gg <- tools::dependsOnPkgs("ggplot2", recursive = FALSE, installed = pdb)
```

©黄湘云

```r
grep("^gg", gg, value = TRUE)
```

```
##   [1] "gg.gap"            "ggalignment"       "ggallin"
##   [4] "ggalluvial"        "ggalt"             "gganimate"
##   [7] "ggarchery"         "ggasym"            "ggbeeswarm"
##  [10] "ggborderline"      "ggbreak"           "ggBubbles"
##  [13] "ggbuildr"          "ggbump"            "ggchangepoint"
##  [16] "ggcharts"          "ggChernoff"        "ggcleveland"
##  [19] "ggconf"            "ggcorrplot"        "ggcorset"
##  [22] "ggdag"             "ggdark"            "ggDCA"
##  [25] "ggdemetra"         "ggdendro"          "ggdensity"
##  [28] "ggdist"            "ggdmc"             "ggDoubleHeat"
##  [31] "ggeasy"            "ggedit"            "ggenealogy"
##  [34] "ggESDA"            "ggetho"            "ggExtra"
##  [37] "ggfacto"           "ggfan"             "ggfittext"
##  [40] "ggfocus"           "ggforce"           "ggformula"
##  [43] "ggfortify"         "ggfun"             "ggfx"
##  [46] "gggap"             "gggenes"           "gggghost"
##  [49] "gggibbous"         "gggrid"            "ggh4x"
##  [52] "gghalfnorm"        "gghalves"          "gghdr"
##  [55] "ggheatmap"         "gghighlight"       "gghilbertstrings"
##  [58] "ggHoriPlot"        "ggimage"           "ggimg"
##  [61] "gginference"       "gginnards"         "ggip"
##  [64] "ggiraph"           "ggiraphExtra"      "ggjoy"
##  [67] "gglm"              "ggloop"            "gglorenz"
##  [70] "ggmap"             "ggmatplot"         "ggmcmc"
##  [73] "ggmice"            "ggmosaic"          "ggmotif"
##  [76] "ggmuller"          "ggmulti"           "ggnetwork"
##  [79] "ggnewscale"        "ggnormalviolin"    "ggnuplot"
##  [82] "ggOceanMaps"       "ggokabeito"        "ggpacman"
##  [85] "ggpage"            "ggparallel"        "ggparliament"
##  [88] "ggparty"           "ggpattern"         "ggperiodic"
##  [91] "ggplot.multistats" "ggplotAssist"      "ggplotgui"
##  [94] "ggplotify"         "ggplotlyExtra"     "ggpmisc"
##  [97] "ggPMX"             "ggpointdensity"    "ggpointless"
## [100] "ggpol"             "ggpolar"           "ggpolypath"
## [103] "ggpp"              "ggprism"           "ggpubr"
## [106] "ggpval"            "ggQC"              "ggQQunif"
## [109] "ggquickeda"        "ggquiver"          "ggraph"
## [112] "ggraptR"           "ggrasp"            "ggrastr"
## [115] "ggrepel"           "ggResidpanel"      "ggridges"
## [118] "ggrisk"            "ggROC"             "ggsci"
## [121] "ggseas"            "ggseg"             "ggseqlogo"
## [124] "ggshadow"          "ggside"            "ggsignif"
## [127] "ggsn"              "ggsoccer"          "ggsolvencyii"
```

©黄湘云

```
## [130] "ggsom"            "ggspatial"        "ggspectra"
## [133] "ggstance"         "ggstar"           "ggstatsplot"
## [136] "ggstream"         "ggstudent"        "ggswissmaps"
## [139] "ggtea"            "ggtern"           "ggtext"
## [142] "ggThemeAssist"    "ggthemes"         "ggtikz"
## [145] "ggTimeSeries"     "ggupset"          "ggvenn"
## [148] "ggVennDiagram"    "ggvoronoi"        "ggwordcloud"
## [151] "ggx"
```

13. 重装 R 包，与 R 版本号保持一致

```r
db <- installed.packages()
db <- as.data.frame(db, stringsAsFactors = FALSE)
pkgs <- db[db$Built < getRversion(), "Package"]
install.packages(pkgs)
```

F.12 查找函数

lookup R 函数完整定义，包括编译的代码，S3 和 S4 方法。目前 lookup 包处于开发版，我们可以用 `remotes::install_github` 函数来安装它

```r
# install.packages("remotes")
remotes::install_github("jimhester/lookup")
```

R-level 的源代码都可以直接看

```r
body
```

```
## function (fun = sys.function(sys.parent()))
## {
##     if (is.character(fun))
##         fun <- get(fun, mode = "function", envir = parent.frame())
##     .Internal(body(fun))
## }
## <bytecode: 0x5595a3a2d9e8>
## <environment: namespace:base>
```

此外，`lookup` 可以定位到 C-level 的源代码，需要联网才能查看，lookup 基于 Winston Chang 在 Github 上维护的 R 源码镜像

```r
lookup(body)
```

```
base::body [closure]
function (fun = sys.function(sys.parent()))
{
    if (is.character(fun))
        fun <- get(fun, mode = "function", envir = parent.frame())
    .Internal(body(fun))
}
```

©黄湘云


```
<bytecode: 0x00000000140d6158>
<environment: namespace:base>
// c source: src/main/builtin.c#L264-L277
SEXP attribute_hidden do_body(SEXP call, SEXP op, SEXP args, SEXP rho)
{
    checkArity(op, args);
    if (TYPEOF(CAR(args)) == CLOSXP) {
        SEXP b = BODY_EXPR(CAR(args));
        RAISE_NAMED(b, NAMED(CAR(args)));
        return b;
    } else {
        if(!(TYPEOF(CAR(args)) == BUILTINSXP ||
             TYPEOF(CAR(args)) == SPECIALSXP))
            warningcall(call, _("argument is not a function"));
        return R_NilValue;
    }
}
```

F.13 运行环境

```
sessionInfo()
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] rmarkdown_2.13 fs_1.5.2       magrittr_2.0.3
##
```

```
## loaded via a namespace (and not attached):
##  [1] bookdown_0.25   sysfonts_0.8.8  digest_0.6.29   evaluate_0.15
##  [5] rlang_1.0.2     stringi_1.7.6   cli_3.2.0       curl_4.3.2
##  [9] tools_4.1.3     stringr_1.4.0   xfun_0.30       yaml_2.3.5
## [13] fastmap_1.1.0   compiler_4.1.3  htmltools_0.5.2 knitr_1.38
```

©清华大学出版社

© 基础统计

附录 G　其它软件

I think, therefore I R.

— William B. King [1]

G.1　文本编辑器

代码文件也是纯文本，RStudio 集成了编辑器，支持语法高亮。Windows 系统上优秀的代码编辑器有 Notepad++ 非常轻量。Markdown 文本编辑器我们推荐 Typora 编辑器，它是跨平台的，下面以 Ubuntu 环境为例，介绍安装和使用过程：

```
# or run:
# sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys BA300B7755AFCFAE
wget -qO - https://typora.io/linux/public-key.asc | sudo apt-key add -

# add Typora's repository
sudo add-apt-repository 'deb https://typora.io/linux ./'
sudo apt-get update

# install typora
sudo apt-get install typora
```

设置中文环境，并且将主题风格样式配置为 Vue，见图G.1（右），Vue 主题可从 Typora 官网下载 https://theme.typora.io/theme/Vue/。

1. Atom 编辑器 https://atom.io/

```
sudo add-apt-repository ppa:webupd8team/atom
sudo apt-get update
sudo apt-get install atom
```

1. Code 编辑器微软出品 https://code.visualstudio.com/
2. Notepad++ 开源的 Windows 平台上的编辑器 https://notepad-plus-plus.org/
3. VI & VIM 开源的跨平台编辑器
4. Atom 和 Code 有商业公司支持的开源免费的跨平台的编辑器
5. VI/VIM 和 Emacs 是跨平台的编辑器

[1] https://ww2.coastal.edu/kingw/statistics/R-tutorials/

(a) 默认的主题 (b) Vue 主题

图 G.1: Typora 主题

6. Markdown 编辑器 + blogdown 记笔记
7. Typora Markdown 编辑器，支持自定义 CSS 样式

G.2 代码编辑器

VS Code, Sublime Text 和 Atom

G.3 集成开发环境

RStudio 公司的愿景，介绍 RStudio 开发环境提供的效率提升工具或功能

G.3.1 RStudio 桌面版

```
# mongolite
sudo dnf install -y  openssl-devel cyrus-sasl-devel
# sodium
sudo dnf install -y  libsodium-devel
# rJava
R CMD javareconf
```

```
# https://github.com/s-u/rJava
# shinytest::installDependencies()
db <- rstudioapi::getRStudioPackageDependencies()

invisible(lapply(db$name, function(pkg) {
  if (system.file(package = pkg) == "") {
    install.packages(pkg)
```

© 草稿

图 G.2: 开源桌面版 RStudio 集成开发环境

```
  }
}))
```

rsthemes 主题

G.3.2 RStudio 服务器版

RStudio Server 开源服务器版可以放在虚拟机里或者容器里，RStudio 桌面版装在服务器上，服务器为 Ubuntu/CentOS/Windows 系统，然后本地是 Windows 系统，可以通过远程桌面连接服务器，使用 RStudio；服务器上启动 Docker ，运行 RStudio 镜像，本地通过桌面浏览器，如谷歌浏览器登陆连接。

1. 下载 RStudio IDE

   我们从 RStudio 官网下载开源桌面或服务器版本，服务器版本的使用介绍见文档，最常见的就是设置端口

   ```
   wget https://download2.rstudio.org/rstudio-server-1.1.456-amd64.deb
   sudo apt-get install gdebi
   sudo gdebi rstudio-server-1.1.456-amd64.deb
   ```

2. 设置端口

   在文件 `/etc/rstudio/rserver.conf` 下，设置

   ```
   www-port=8181
   ```

   注意：修改 `rserver.conf` 文件后需要重启才会生效

   ```
   sudo rstudio-server stop
   sudo rstudio-server start
   ```

   接着获取机器的 IP 地址，如 192.168.141.3

© 清华出版社

图 G.3: 虚拟机里的 RStudio

图 G.4: 容器里的 RStudio

© 王沛


```
ip addr
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
 inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
 inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
    link/ether 08:00:27:59:c0:fb brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic enp0s3
       valid_lft 83652sec preferred_lft 83652sec
    inet6 fe80::a00:27ff:fe59:c0fb/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 100
    link/ether 08:00:27:09:33:0d brd ff:ff:ff:ff:ff:ff
    inet 192.168.141.3/24 brd 192.168.141.255 scope global dynamic enp0s8
       valid_lft 547sec preferred_lft 547sec
    inet6 fe80::a00:27ff:fe09:330d/64 scope link
       valid_lft forever preferred_lft forever
```

然后，就可以从本地浏览器登陆 RStudio 服务器版本，如 http://192.168.141.3:8181/

提示

rstudio-server 已经收录在 Fedora 33+ 仓库中了，详情见 https://cran.r-project.org/bin/linux/fedora/

授权问题 ERROR system error 13 (Permission denied) How to Disable SELinux Temporarily or Permanently

G.3.3 Shiny 服务器版

shiny 开源服务器版

G.3.4 Eclipse + StatET

Eclipse 配合 StatET 插件 http://www.walware.de/goto/statet 提供 R 语言的集成开发环境 https://projects.eclipse.org/projects/science.statet

StatET 基于 Eclipse 首次建立索引很慢，估计半小时到一个小时，添加新的 R 包后，每次启动 StatET 也会建立索引缓存，此外，Eclipse 开发环境占用内存比较多，配置 StatET 的过程如下

G.3.5 Emacs + ESS

Emacs 配合 ESS 插件 https://ess.r-project.org/

©黄湘云

图 G.5: 基于 Eclipse 的 R 集成开发环境 StatET

G.3.6 Nvim-R

Nvim-R 是一个基于 Vim 的集成开发环境 https://github.com/jalvesaq/Nvim-R

G.4 Git 版本控制

Git 操作

MacOS 上用 Homebrew 安装 git-delta

```
brew install git-delta
```

gitdown

只考虑 Ubuntu 18.04 环境下的三剑客 Git & Github & Gitlab

```
summary(git2r::repository())
```

```
## Local:    devel /home/runner/work/masr/masr
## Remote:   devel @ origin (https://github.com/XiangyunHuang/masr)
## Head:   [bcc93b4] 2022-04-13: 点出 maps 包绘制地图最关键的地方：地图数据如何与观测数据建立一一映射
##
## Branches:         1
## Tags:             0
## Commits:          5
## Contributors:     1
## Stashes:          0
## Ignored files:   12
```

© 谢湘


```
## Untracked files: 88
## Unstaged files:   0
## Staged files:     0
##
## Latest commits:
## [bcc93b4] 2022-04-13: 点出 maps 包绘制地图最关键的地方：地图数据如何与观测数据建立一一映射
## [322d23f] 2022-04-04: 去掉限制 out.width="75%"
## [982d110] 2022-04-03: Polish
## [3550750] 2022-04-01: use system fonts
## [430c26c] 2022-03-31: upgrade to R 4.1.3
```

仓库 masr 哪些人给我点赞加星了

```
library(gh)
my_repos <- gh("GET /repos/:owner/:repo/stargazers", owner = "xiangyunhuang", repo = "masr", page = 1
vapply(my_repos, "[[", "", "login")
```

```
 [1] "dddd1007"        "boltomli"        "JackieMium"      "AXGL"            "fyemath"
 [6] "rogerclarkgc"    "swsoyee"         "joegaotao"       "YTLogos"         "Accelerator086"
[11] "yimingfish"      "gaospecial"      "shenxiangzhuang" "shuaiwang88"     "LusiXie"
[16] "llxlr"           "TingjieGuo"      "oiatz"           "XiaogangHe"      "xwydq"
[21] "guohongwang1"    "yinandong"       "algony-tony"     "XiangyunHuang"   "perlatex"
[26] "talegari"        "hao-shefer"      "zhouyisu"        "tsitong"         "liuyadong"
```

图 G.6: Git 代码版本管理

Jeroen Ooms 开发的 gert 包实现在 R 环境中操作 Git，我们可以从幻灯片 — Gert: A minimal git client for R 学习重点内容。

```
library(gert)
library(magrittr)
git_log(max = 10) %>%
  subset(subset = grepl("Yihui Xie", x = author), select = c("author", "message"))
```

提供了 `git_rm()`、`git_status()`、`git_add()` 和 `git_commit()` 等函数，其中包含 `git_reset()` 高级的 Git 操作。此外，还有 `git_branch_*()` 系列分支操作函数

G.4.1 安装配置

Ubuntu 16.04.5 默认安装的 Git 版本是 2.7.4，下面安装最新版本 Git 和配置自己的 GitHub 账户

© 草稿

1. 根据官网安装指导 https://git-scm.com/download/linux，在 Ubuntu 14.04.5 和 Ubuntu 16.04.5 安装最新版 GIT

```
sudo add-apt-repository -y ppa:git-core/ppa
sudo apt update && sudo apt install git
```

2. 配置账户

```
git config --global user.name "你的名字"
git config --global user.email "你的邮件地址"
touch .git-credentials
# 记住密码
echo "https://username:password@github.com" >> .git-credentials
git config --global credential.helper store
```

以 Fedora 为例 安装 tig，首先安装必要的依赖，然后从官网下载源码，编译安装，之后切到任意本地 Git 仓库下，输入 `tig` 就可以看到如图 G.7 所示的样子了

```
sudo yum install readline-devel ncurses-devel asciidoc docbook-utils xmlto
```

tig 主要用于查看 git 提交的历史日志

图 G.7: Git 日志查看器

G.4.2 追踪文件

```
git add .
```

提交新文件 (new) 和被修改 (modified) 文件，不包括被删除 (deleted) 文件

```
git add -u
```

提交被修改 (modified) 和被删除 (deleted) 文件，不包括新文件 (new)，`git add --update` 的缩写

```
git add -A
```

提交所有变化，`git add --all` 的缩写

© 黄湘云


```
git init
git remote add origin https://github.com/XiangyunHuang/masr.git
git add -A
git commit -m "添加提交说明"
git push -u origin master
```

往远程的空的 Github 仓库添加本地文件

G.4.3 合并上流

```
git clone --depth=5 https://github.com/XiangyunHuang/cosx.org.git
git submodule update --init --recursive
```

查看远程分支

```
cd cosx.org
git remote -v
```

```
origin  https://github.com/XiangyunHuang/cosx.org.git (fetch)
origin  https://github.com/XiangyunHuang/cosx.org.git (push)
```

```
# 添加上流分支
git remote add upstream https://github.com/cosname/cosx.org.git
# 查看远程分支
git remote -v
```

```
origin  https://github.com/XiangyunHuang/cosx.org.git (fetch)
origin  https://github.com/XiangyunHuang/cosx.org.git (push)
upstream        https://github.com/cosname/cosx.org.git (fetch)
upstream        https://github.com/cosname/cosx.org.git (push)
```

```
# 获取上流 commit 并且合并到我的 master 分支
git fetch upstream
git merge upstream/master master
git push origin master
```

G.4.4 大文件支持

```
sudo apt install git-lfs
git lfs install
git lfs track "*.psd"
git add .gitattributes
git commit -m "track *.psd files using Git LFS"
git push origin master
```

这玩意迟早需要你购买存储空间，慎用

© 黄海广

G.4.5 新建分支

```
git checkout -b stan     # 新建 stan 分支
git branch -v            # 查看本地分支 stan 前有个星号标记
git pull --rebase git@github.com:XiangyunHuang/cosx.org.git master
# 同步到远程分支 stan
git push --set-upstream origin stan
git push origin master:stan

git add .
git commit -m "balabala"
git push --set-upstream origin stan
```

本地新建仓库推送至远程分支

```
git remote add origin https://github.com/XiangyunHuang/notesdown.git
git add .
git commit -m "init cos-art"
# 此时远程仓库 notesdown 还没有 cos-art 分支
git push origin master:cos-art
```

位于 Github Git Community Book 中译本

G.4.6 创建 Github Pages 站点

基于 GitHub Pages 创建站点用于存放图片和数据

1. 在 Github 上创建一个空的仓库，命名为 uploads，没有 readme.md 和 LICENSE
2. 在本地创建目录 uploads
3. 切换到 uploads 目录下

```
git init
git checkout -b gh-pages
git remote add origin https://github.com/XiangyunHuang/uploads.git
```

添加图片或者数据，并且 git add 和 commit 后

```
git push --set-upstream origin gh-pages
```

这样仓库 uploads 只包含 gh-pages 分支，数据地址即为以日期为分割线

https://xiangyunhuang.github.io/uploads/data/eqList2018_05_18.xls

G.4.7 博客主题

初始化博客网站

```
git subtree add --squash --prefix=themes/hugo-lithium \
  git@github.com:yihui/hugo-lithium.git master
```

© 黄湘云

在 Github 创建新的空仓库，本地创建空的目录 xiangyun

```
cd xiangyun
git init
git remote add origin https://github.com/XiangyunHuang/xiangyun.git

git add .gitignore
git commit -m 'upload'
git push --set-upstream origin master
```

git subtree 将另外一个仓库收缩为当前仓库的一个目录，且只产生一条提交记录

```
# 子库分支
git subtree add --squash --prefix=themes/hugo-xmag \
  -m "add hugo-xmag" git@github.com:yihui/hugo-xmag.git master
# 或者子库分支
git subtree add --squash --prefix=themes/hugo-xmag \
  -m "add hugo-xmag" https://github.com/yihui/hugo-xmag.git master

# 移除 git subtree 添加的 hugo 主题
git filter-branch --index-filter 'git rm --cached --ignore-unmatch -rf themes/hugo-xmag' --prune-empt
```

G.4.8 修改远程仓库的位置

有时候我们将自己的仓库转移给别人/组织，或者我们将远程仓库的名字改变了，这时候需要修改远程仓库的位置。比如最近我将博客仓库从 https://github.com/XiangyunHuang/xiangyun 转移到 https://github.com/rbind/xiangyun

转移前

```
git remote -v
```

```
origin  https://github.com/XiangyunHuang/xiangyun.git (fetch)
origin  https://github.com/XiangyunHuang/xiangyun.git (push)
```

转移命令

```
git remote set-url origin https://github.com/rbind/xiangyun.git
```

转移后

```
git remote -v
```

```
origin  https://github.com/rbind/xiangyun.git (fetch)
origin  https://github.com/rbind/xiangyun.git (push)
```

G.4.9 统计代码仓库的提交量

比如统计之都的主站仓库，提交量最大的 20 个人

©黄海广

```
git shortlog -sn | head -n 20
```

```
  153   Dawei Lang
  106   Yihui Xie
   89   Beilei Bian
   46   王佳
   42   雷博文
   39   Ryan Feng Lin
   35   Xiangyun Huang
   32   fanchao
   32   闫晗
   30   Lin Feng
   28   Jiaao Yu
   25   fyears
   24   Yixuan Qiu
   24   Miao YU
   22   Yuxuan Li
   22   qinwf
   20   Alice敏
   19   yanshi
   18   Shuyi.Yang
   13   黄湘云
```

G.4.10 账户共存

本节介绍如何使 Gitlab/Github 账户共存在一台机器上

如何生成 SSH 密钥见 Github 文档 — 使用 SSH 连接到 GitHub。有了密钥之后只需在目录 `~/.ssh` 下创建一个配置文件 config

生成 SSH Key

```
ssh-keygen -t rsa -f ~/.ssh/id_rsa_github -C "name1@xxx1.com"
ssh-keygen -t rsa -f ~/.ssh/id_rsa_gitlab -C "name2@xxx2.com"
```

将 GitHub/GitLab 公钥分别上传至服务器，然后创建配置文件

```
touch ~/.ssh/config
```

配置文件内容如下

```
#
# Github
#
Host github.com // 个人的代码仓库服务器地址
HostName github.com
User XiangyunHuang
IdentityFile ~/.ssh/id_rsa_github
```

© 黄湘云


```
#
# company
#
Host xx.xx.xx.xx //
IdentityFile ~/.ssh/id_rsa_gitlab
```

配置成功，你会看到

```
ssh -T git@xx.xx.xx.xx
```

```
Welcome to GitLab, xiangyunhuang!
```

和

```
ssh -T git@github.com
```

```
Hi XiangyunHuang! You've successfully authenticated, but GitHub does not provide shell access.
```

G.4.11 回车换行

CR (Carriage Return) 表示回车，LF (Line Feed) 表示换行，Windows 下用回车加换行表示下一行，UNIX/Linux 采用换行符 (LF) 表示下一行，MAC OS 则采用回车符 (CR) 表示下一行

```
git config --global core.autocrlf false
```

G.4.12 子模块

- 添加子模块到目录 `templates/` 下

```
git submodule add git://github.com/jgm/pandoc-templates.git templates
```

- 移除子模块

https://stackoverflow.com/questions/1260748/how-do-i-remove-a-submodule/

G.4.13 克隆项目

```
git clone --depth=10 --branch=master --recursive \
    git@github.com:XiangyunHuang/pandoc4everything.git
```

G.4.14 创建 PR

```
git pull --rebase git@github.com:yihui/xaringan.git master
# then force push to your master branch
```

参考 https://github.com/yihui/xaringan/pull/107

> I don't recommend you to use your master branch for pull requests, because all commits will be squashed before merging, e.g. c2c2055 Then you will have some trouble with syncing

© 黄湘云

your master branch with the master branch here (your choices are (1) delete your repo and fork again; or (2) force push; either option is not good). For pull requests, I recommend that you always use different branches for different pull requests.

G.4.15 修改 PR

之前一直有一个思想在阻止自己，就是别人的 repo 我是不能修改的，但是在这里，我拥有修改原始仓的权限，那么别人的复制品衍生的分支，我也有修改权限

```
git fetch origin refs/pull/771/head:patch-2
# 771 是 PR 对应的编号
git checkout patch-2

# 你的修改

git add -u
git commit -m "描述你的修改"

git remote add LalZzy https://github.com/LalZzy/cosx.org.git

git push --set-upstream LalZzy patch-2
```

整理自统计之都论坛的讨论 https://d.cosx.org/d/420363

1. GitHub/Git 小抄英文版 https://www.runoob.com/manual/github-git-cheat-sheet.pdf
2. GitHub/Git 小抄中文版 https://github.github.com/training-kit/downloads/zh_CN/github-git-cheat-sheet/
3. Github 秘籍 https://github.com/tiimgreen/github-cheat-sheet/blob/master/README.zh-cn.md
4. Git 简明指南 https://rogerdudler.github.io/git-guide/index.zh.html
5. Git 奇技淫巧 https://github.com/521xueweihan/git-tips
6. Git 官方书籍 https://git-scm.com/book/zh/v2
7. Git 时代的 VIM 不完全使用教程 http://beiyuu.com/git-vim-tutorial
8. 最佳搭档：利用 SSH 及其配置文件节省你的生命 https://liam.page/2017/09/12/rescue-your-life-with-SSH-config-file/

G.5 Pandoc 文档处理

Pandoc 是一个万能文档转化器，安装 pandoc，下载网址 https://github.com/jgm/pandoc/releases/latest

```
sudo apt-get install gdebi-core
wget https://github.com/jgm/pandoc/releases/download/2.9.2/pandoc-2.9.2-1-amd64.deb
sudo chmod +x pandoc-2.9.2-1-amd64.deb
sudo gdebi pandoc-2.9.2-1-amd64.deb
```

rmarkdown 包裹了 Pandoc 工具，使用 `rmarkdown::render()` 函数即可将 R Markdown 文档转化为 HTML、LaTeX 和 Markdown 等格式。

© 黄湘云

G.6 Calibre 书籍管理

Calibre 是一款电子书转化和管理软件，首先安装 calibre

```
sudo -v && wget -nv -O- https://download.calibre-ebook.com/linux-installer.sh | sudo sh /dev/stdin
```

calibre 可以将 epub 格式电子书文档转化为 mobi 格式，bookdown 已经给这个工具穿上了一件马甲，用户只需调用 `bookdown::calibre()` 函数即可实现电子书格式的转换。

G.7 ImageMagick 图像处理

图像的各种操作，包括合成、转换、旋转等等

首先安装 ImageMagick 软件包中的 convert 程序

```
asy -f jpg test.asy
```

指定分辨率

```
convert -geometry 1000x3000 -density 300 -units PixelsPerInch INPUT.eps OUTPUT.png
```

这样不改变图像的像素数，只是给出一个每个像素应该显示多大的提示。

```
convert -quality 100 -density 300x300 INPUT.pdf OUTPUT.png
```

高质量大图，给定像素，转化 eps 格式图片，需要先安装 Ghostscript

```
convert -geometry 1000x3000 INPUT.eps OUTPUT.png
```

```
convert -quality 100 -antialias -density 96 -transparent white -trim INPUT.pdf OUTPUT.png
```

选项	作用
trim	裁剪图像四周空白区域
transparent color	去除图像中指定的颜色
density geometry	设定图像的 DPI 值
antialias	让图像具有抗锯齿的效果
quality	图像压缩等级

像素、点等常见术语

符号	含义
px	pixel 像素，电子屏幕上组成一幅图画或照片的最基本单元
pt	point，点，印刷行业常用单位，等于 1/72 英寸
ppi	pixel per inch，每英寸像素数，该值越高，则屏幕越细腻
dpi	dot per inch，每英寸多少点，该值越高，则图片越细腻

多页的 PDF 文件转化为多张 PNG 图片

```
convert -quality 100 -density 300x300  INPUT.pdf OUTPUT.png
```

将多页 PDF 文件合成为 GIF 动图

```
convert -delay 60 -density 300x300 -background white -alpha remove \
  -dispose previous INPUT.pdf -layers coalesce OUTPUT.gif
```

见益辉博客一些 ImageMagick 命令

G.8 OptiPNG 图片优化

OptiPNG 是一个非常好的图片压缩、优化工具

现在，我们设置 chunk 选项 `optipng` 为非空 (non-`NULL`) 的值，例如，`''` 去激活这个 hook （益辉称之为钩子，这里勾的是 optipng 这个图片优化工具）

```
knitr::knit_hooks$set(optipng = knitr::hook_optipng)
```

```
library(ggplot2)
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point()
```

图 G.8: 没有优化

```
library(ggplot2)
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +
```

© 黄湘云


```
geom_point()
```

图 G.9: 优化

```
optipng -o5 filename.png
```

G.9 PDFCrop 裁剪边空

PDFCrop 可将 PDF 图片中留白的部分裁去，再也不用纠结 par 了

G.10 PhantomJS 网页截图

Winston Chang 开发了 webshot 包网页截图，它依赖 PhantomJS，所以首先需要安装

```
install.packages("webshot")
webshot::install_phantomjs()
```

以截取网页 https://www.r-project.org/ 为例，

```
library(webshot)
webshot("https://www.r-project.org/", "r.png")
webshot("https://www.r-project.org/", "r.pdf") # Can also output to PDF
```

还可以截取 R Markdown 文档内容，注意是先编译 R Markdown 文档为 HTML 文档，然后截取网页

统计出版社 ©

```
rmdshot(system.file("examples/knitr-minimal.Rmd", package = "knitr"), file = "screenshots/knitr-minimal.pn
```

裁剪出特定大小的图片，需要额外的系统依赖 GraphicsMagick (recommended) or ImageMagick installed

```
# Can specify pixel dimensions for resize()
webshot("https://www.r-project.org/", "r-small.png") %>%
  resize("400x") %>%
  shrink()
```

```
** Processing: r-small.png
400x442 pixels, 4x8 bits/pixel, RGB+alpha
Reducing image to 3x8 bits/pixel, RGB
Input IDAT size = 70570 bytes
Input file size = 70867 bytes

Trying:
  zc = 9  zm = 8  zs = 0  f = 0         IDAT size = 59441
  zc = 9  zm = 8  zs = 1  f = 0
  zc = 1  zm = 8  zs = 2  f = 0
  zc = 9  zm = 8  zs = 3  f = 0
  zc = 9  zm = 8  zs = 0  f = 5
  zc = 9  zm = 8  zs = 1  f = 5
  zc = 1  zm = 8  zs = 2  f = 5
  zc = 9  zm = 8  zs = 3  f = 5

Selecting parameters:
  zc = 9  zm = 8  zs = 0  f = 0         IDAT size = 59441

Output IDAT size = 59441 bytes (11129 bytes decrease)
Output file size = 59714 bytes (11153 bytes = 15.74% decrease)
```

G.11 Inkscape 矢量绘图

Inkscape 是一款开源、免费、跨平台的矢量绘图软件。是替代 Adobe Illustrator（简称 AI）最佳工具，没有之一

```
# Ubuntu 20.04 及之前版本
sudo add-apt-repository ppa:inkscape.dev/stable
sudo apt update
sudo apt install inkscape
```

PDF 图片格式转化为 SVG 格式

```
inkscape -l output-filename.svg input-filename.pdf
```

SVG 转 PDF 格式

© 基础


```
inkscape -f input-filename.svg -A output-filename.pdf
```

```
inkscape --export-type=png in1.svg in2.svg
```

Jeroen Ooms 开发的 rsvg 包支持将 SVG 格式图片导出为 PNG、PDF、PS 等格式。使用它可以批量将 SVG 格式文件转化为其它格式文件，比如 PDF（`rsvg::rsvg_pdf`），PS（`rsvg::rsvg_ps`）和 PNG（`rsvg::rsvg_png`）

```
svg_paths = list.files(path = "images", pattern = "*.svg", full.names = T)
for (svg in svg_paths) {
  rsvg::rsvg_pdf(svg, file = gsub(pattern  = "\\.svg", replacement=  "\\.pdf", svg))
}
```

G.12 QPDF PDF 文件操作

Jeroen Ooms 开发的另一个 qpdf 包将 C++ 库 qpdf 搬运到 R 环境中，用于 PDF 文件的拆分 `pdf_split()`，组合 `pdf_combine()`，加密（传递 `password` 参数值即可加密），提取 `pdf_subset()` 和压缩 `pdf_compress()` 等。下面以组合为例，就是将多个 PDF 文件合成一个 PDF 文件。

```
library(qpdf)
pdf_paths = list.files(path = "images", pattern = "*.pdf", full.names = T)
pdf_combine(input = pdf_paths, output = "images/all.pdf", password = "")
```

PDF 操作：价值数百美元的开源替代方案，参考 Adobe Acrobat 的功能

G.13 UML 标准建模图

UML (Unified Modeling Language) 表示统一建模语言

图 G.10: 图片制作、合成、优化、转换等常用工具

© 基础版

Javier Luraschi 将 UML 绘图库 nomnoml 引入 R 社区，开发了 nomnoml 包，相比于 DiagrammeR 包，它显得非常轻量，网站 https://www.nomnoml.com/ 还可以在线编辑、预览、下载 UML 图。**webshot** 包可以将网页截图并插入 PDF 文档中。其它制作图形的工具见 G.10。

nomnoml 调 webshot 包对网页截图生成 PNG 格式的图片，其中 webshot 调 phantomjs 软件。nomnoml 制作 R Markdown 生态图，导出为 PNG 格式

安装 nomnoml

```
install.packages("nomnoml")
```

安装 PhantomJS

```
brew install --cask phantomjs
```

```
nomnoml::nomnoml("
#stroke: #26A63A
#.box: dashed visual=ellipse
#direction: down

[<box>HTML]       -> [网页三剑客]
[<box>JavaScript] -> [网页三剑客]
[<box>CSS]        -> [<table>网页三剑客|htmlwidgets|htmltools||sass|bslib||thematic|jquerylib]

[设计布局|bs4Dash|flexdashboard|shinydashboard] -> [<actor>开发应用|R Shiny]
[设计交互|waiter|shinyFeedback|shinyToastify] -> [<actor>开发应用|R Shiny]
[权限代理|shinyproxy|shinyauthr|shinymanager] -> [<actor>开发应用|R Shiny]

[网页三剑客]  -> [<actor>开发应用|R Shiny]
[网页三剑客]  -> [<actor>开发应用|R Shiny]
[网页三剑客]  -> [<actor>开发应用|R Shiny]

[开发应用] <- [<table>处理数据|Base R|SQL||data.table|dplyr||tidyr|purrr]
[开发应用] <- [<table>制作表格|DT|gt||reactable|formattable||kableExtra|sparkline]
[开发应用] <- [<table>制作图形|ggplot2|plotly||echarts4r|leaflet||dygraphs|visNetwork]
", png = "shiny-app.png")
```

G.14 Graphviz 流程图

Graphviz 官网 http://www.graphviz.org/，常用于绘制流程图，广泛用于 tensorflow 和 mxnet 的模型描述中

DiagrammeR 包将 Graphviz 引入 R 语言

```
library(DiagrammeR)
library(DiagrammeRsvg)
library(magrittr)
library(rsvg)
```

© 黄湘云

图 G.11: 数据分析流程图

```
graph <-
  "graph {
        rankdir=LR; // Left to Right, instead of Top to Bottom
        a -- { b c d };
        b -- { c e };
        c -- { e f };
        d -- { f g };
        e -- h;
        f -- { h i j g };
        g -- k;
        h -- { o l };
        i -- { l m j };
        j -- { m n k };
        k -- { n r };
        l -- { o m };
        m -- { o p n };
        n -- { q r };
        o -- { s p };
        p -- { s t q };
        q -- { t r };
        r -- t;
        s -- z;
        t -- z;
    }
"
# 导出图形
grViz(graph) %>%
  export_svg %>% charToRaw %>% rsvg_pdf("graph.pdf")
```

© 基础统计

```
grViz(graph) %>%
  export_svg %>% charToRaw %>% rsvg_png("graph.png")
grViz(graph) %>%
  export_svg %>% charToRaw %>% rsvg_svg("graph.svg")
```

G.15 LaTeX 排版工具

另外值得一提的是 TikZ 和 PGF（Portable Graphic Format）宏包，支持强大的绘图功能，图形质量达到出版级别，详细的使用说明见宏包手册 https://pgf-tikz.github.io/pgf/pgfmanual.pdf。

G.15.1 TinyTeX 发行版

```
library(tinytex)
# 升级 TinyTeX 发行版
upgrade_tinytex <- function(repos = NULL) {
  # 此处还要考虑用户输错的情况和选择离用户最近（快）的站点
  if(is.null(repos)) repos = "https://mirrors.tuna.tsinghua.edu.cn/CTAN/"

  file_ext <- if (.Platform$OS.type == "windows") ".exe" else ".sh"
  tlmgr_url <- paste(repos, "/systems/texlive/tlnet/update-tlmgr-latest", file_ext, sep = "")
  file_name <- paste0("update-tlmgr-latest", file_ext)
  download.file(url = tlmgr_url, destfile = file_name,
                mode = if (.Platform$OS.type == "windows") "wb" else "w")

  # window下 命令行窗口下 如何执行 exe 文件
  if(.Platform$OS.type == "windows"){
    shell.exec(file = file_name)
    file.remove("update-tlmgr-latest.exe")
  }
  else{
    system("sudo sh update-tlmgr-latest.sh  -- --upgrade")

    file.remove("update-tlmgr-latest.sh")
  }

  # 类似地 Linux 下执行 sh
  # 升级完了 删除 update-tlmgr-latest.exe
}
```

Winston Chang 整理了一份 LaTeX 常用命令速查小抄 https://wch.github.io/latexsheet/latexsheet.pdf

© 黄湘云

G.15.2 安装和更新

tlmgr (TeXLive Manager) 是 LaTeX 包管理器

```
# 就近选择 CTAN 镜像站点
tlmgr option repository https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/tlnet
tlmgr option repository http://mirror.ctan.org/systems/texlive/tlnet
# 可更新的 TeX 包列表
tlmgr update --list
# 更新所有已经安装的 TeX 包
tlmgr update --all
# 更新 tlmgr 管理器本身
tlmgr update --self
# 安装
tlmgr install ctex fandol
# 列出套装
tlmgr list schemes
tlmgr list collections
# 列出已经安装的 TeX 包
tlmgr list --only-installed
# 安装 GPG 公钥 (只限Win/Mac)
tlmgr --repository http://www.preining.info/tlgpg/ install tlgpg
```

G.15.3 查询和搜索

```
tlmgr search *what*
```

参数 `\*what\*` 是正则表达式

```
tlmgr search --file tikz.sty
```

```
## pgf:
##  texmf-dist/tex/latex/pgf/frontendlayer/tikz.sty
```

等价于

```
tinytex::tlmgr_search('tikz.sty')
```

这样，我们就可以知道要使用 `\usepackage{tikz}` 就得先安装 **pgf** 包，此外，管道命令也是支持的

```
tlmgr search --file font | grep math
```

查询 CTAN 仓库列表

```
tlmgr repository list
```

一般地，只显示已安装的 LaTeX 宏包的名字及大小

```
tlmgr info --list --only-installed --data name,size
```

更多命令详见tlmgr 管理器手册

G.15.4 TikZ 绘图工具

TikZ 绘制书籍封面 https://latexdraw.com/how-to-create-a-beautiful-cover-page-in-latex-using-tikz/

TikZ 绘制知识清单，书籍章节结构等 https://www.latexstudio.net/index/lists/barsearch/author/1680.html

更多例子参考 https://github.com/FriendlyUser/LatexDiagrams

G.16 Octave 科学计算

```
%% fig1
tx = ty = linspace (-8, 8, 41)';
[xx, yy] = meshgrid (tx, ty);
r = sqrt (xx .^ 2 + yy .^ 2) + eps;
tz = sin (r) ./ r;
mesh (tx, ty, tz);
xlabel ("tx");
ylabel ("ty");
zlabel ("tz");
title ("3-D Sombrero plot");


% fig2
x = 0:0.01:3;
hf = figure ();
plot (x, erf (x));
hold on;
plot (x, x, "r");
axis ([0, 3, 0, 1]);
text (0.65, 0.6175, ['$\displaystyle\leftarrow x = {2\over\sqrt{\pi}}'...
                     '\int_{0}^{x}e^{-t^2} dt = 0.6175$']);
xlabel ("x");
ylabel ("erf (x)");
title ("erf (x) with text annotation");
set (hf, "visible", "off");
print (hf, "plot15_7.pdf", "-dpdflatexstandalone");
set (hf, "visible", "on");
system ("pdflatex plot15_7");
open ("plot15_7.pdf");

%% fig3

clf ();
surf (peaks);
```

© 草稿


```
peaks(50)
print -dpswrite -PPS_printer

%% images/peaks-inc
hf = figure (1);
surf (peaks);
print (hf, "peaks.pdf", "-dpdflatexstandalone");

%% windows
hf = figure (1);
peaks(10);
print (hf, "peaks.pdf", "-dpdf");
print (hf, "peaks.eps", "-color"," -deps");

print (hf, "peaks.svg", "-color"," -dsvg");


%% windows
hf = figure (1);
peaks(50);
print (hf, "peaks-more.eps", "-color"," -deps");

print (hf, "peaks-more.svg", "-color"," -dsvg");
```

G.17 Python 环境配置

首先创建一个 Python 虚拟环境，环境隔离可以减少对系统的侵入，方便迭代更新和项目管理。创建一个虚拟环境，步骤非常简单，下面以 CentOS 8 为例：

1. 安装虚拟模块 virtualenv

```
sudo dnf install -y virtualenv
```

2. 准备 Python 虚拟环境存放位置

```
sudo mkdir -p /opt/.virtualenvs/r-tensorflow
```

3. 给虚拟环境必要的访问权限

```
sudo chown -R $(whoami):$(whoami) /opt/.virtualenvs/r-tensorflow
```

4. 初始化虚拟环境

```
virtualenv -p /usr/bin/python3 /opt/.virtualenvs/r-tensorflow
```

5. 激活虚拟环境，安装必要的模块

```
source /opt/.virtualenvs/r-tensorflow/bin/activate
pip install numpy
```

© 基础出版

一般来讲，系统自带的 pip 版本较低，可以考虑升级 pip 版本。

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple pip -U
```

根据项目配置文件 requirements.txt 安装多个 Python 模块，每个 Python 项目都应该有这么个文件来描述项目需要的依赖环境，包含 Python 模块及其版本号。

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple -r requirements.txt
```

指定 Python 模块的镜像地址，加快下载速度，特别是对于国内的环境，加速镜像站点非常有意义，特别是遇到大型的 Python 模块，比如 tensorflow 框架

```
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple tensorflow
```

conda 创建 Python 3.8 虚拟环境，并命名为 tensorflow

```
conda create -n tensorflow python=3.8
```

激活 tensorflow 环境

```
conda activate tensorflow
```

G.18 Python 基础绘图

Python 的 matplotlib 模块支持保存的图片格式有 eps, pdf, pgf, png, ps, raw, rgba, svg, svgz，不支持 cairo_pdf 绘图设备，所以这里使用 pdf 设备，但是这样会导致图形没有字体嵌入，从而不符合出版要求。一个解决办法是在后期嵌入字体，图形默认使用数学字体 STIX 和英文字体 DejaVu Sans，所以需要预先安装这些字体。

```
# CentOS 8
sudo dnf install -y dejavu-fonts-common dejavu-sans-fonts \
  dejavu-serif-fonts dejavu-sans-mono-fonts
```

借助 **grDevices** 包提供的 `embedFonts()` 函数，它支持 postscript 和 pdf 图形设备，嵌入字体借助了 Ghostscript 以及 PDF 阅读器 MuPDF

注意

Windows 系统下需要手动指定 Ghostscript 安装路径，特别地，如果你想增加可选字体范围，需要指定相应字体搜索路径，而 Linux/MacOS 平台下不需要关心 Ghostscript 的安装路径问题，

```
Sys.setenv(R_GSCMD = "C:/Program Files/gs/gs9.26/bin/gswin64c.exe")
embedFonts(
  file = "cm.pdf", outfile = "cm-embed.pdf",
  fontpaths = system.file("fonts", package = "fontcm")
)
embedFonts(file = "cm.pdf", outfile = "cm-embed.pdf")
```

另一个解决办法是使用 LaTeX 渲染图片中的文字，这就需要额外安装一些 LaTeX 宏包，此时默认执行渲染的 LaTeX 引擎是 PDFLaTeX。

```
tlmgr install type1cm cm-super dvipng psnfss ucs ncntrsbk helvetic
```

© 基础

每年 4 月是 TeX Live 的升级月，升级指导见 https://www.tug.org/texlive/upgrade.html，升级之后，需要更新所有 LaTeX 宏包。

```
tlmgr update --self --all
```

如下图所示，我们采用第二个方法，它可以支持更好的数学公式显示，更多详情见 https://matplotlib.org/tutorials/text/mathtext.html。

提示

如果你的系统是 Windows/MacOS 可以添加 GPG 验证以增加安全性，最简单的方式就是：

```
tlmgr --repository http://www.preining.info/tlgpg/ install tlgpg
```

二维函数 $f(x,y) = 20 + x^2 + y^2 - 10 * \cos(2 * \pi * x) - 10 * \cos(2 * \pi * y)$ 最小值 0，最大值 80

```
from math import cos, pi
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from matplotlib import cm

from matplotlib import rcParams
rcParams.update({'font.size': 18, 'text.usetex': True})
# 其它可配置选项见 rcParams.keys()
plt.switch_backend('agg')

xDomain = np.arange(-5.12, 5.12, .08)
yDomain = np.arange(-5.12, 5.12, .08)

X, Y = np.meshgrid(xDomain, yDomain)
z = [20 + x**2 + y**2 - (10*(cos(2*pi*x) + cos(2*pi*y))) for x in xDomain for y in yDomain]
Z = np.array(z).reshape(128,128)

fig = plt.figure(figsize = (12,10))
ax = fig.gca(projection='3d')
surf = ax.plot_surface(X, Y, Z, cmap=cm.RdYlGn, linewidth=1, antialiased=False)

ax.set_xlim(-5.12, 5.12)
ax.set_ylim(-5.12, 5.12)
ax.set_zlim(0, 80)
# fig.colorbar(surf, aspect=30)
# plt.title(r'Rastrigin Function in Two Dimensions')
plt.tight_layout()
plt.show()
```

© 基础统计

G.19 Python 基础操作

- 张量操作 numpy https://numpy.org/ 向量、矩阵操作
- 科学计算 scipy https://scipy.org/ 统计、优化和方程
- 数据操作 pandas https://pandas.pydata.org/ 面向数据分析
- 数据可视化 matplotlib https://matplotlib.org/ 静态图形
- 交互可视化 bokeh https://bokeh.org/
- 机器学习 scikit-learn https://scikit-learn.org/ 面向机器学习
- 深度学习 tensorflow https://tensorflow.org/ 面向深度学习

A Python implementation of global optimization with gaussian processes. Bayesian Optimization

用 numpy 实现一个统计类的算法，比如线性回归、稳健的线性回归、广义线性回归，数据集用 Python 内置的

```
import numpy as np
np.zeros(3) # vector
```

```
## array([0., 0., 0.])
```

```
np.ones(3) # vector
```

```
## array([1., 1., 1.])
```

```
np.diag([1,1,1]) # identy matrix
# np.multiply()
```

```
## array([[1, 0, 0],
##        [0, 1, 0],
##        [0, 0, 1]])
```

```
np.cumsum([1,1,1])
```

```
## array([1, 2, 3])
```

Python 模块 scikit-learn [Pedregosa et al., 2011] 内置的数据集 iris 为例 https://scikit-learn.org/stable/datasets/index.html

导入正则表达式库，

```
import re
m = re.search('(?<=abc)def', 'abcdef')
m.group(0) # 必须调用 print 函数打印结果
```

```
## 'def'
```

```
print(m.group(0))
```

```
## def
```

```
import sys
print(sys.path)
```

```
## ['', '/usr/bin', '/usr/lib/python38.zip', '/usr/lib/python3.8', '/usr/lib/python3.8/lib-dynload', '/opt/.
```

字符串基本操作，如拆分

© 黄湘


```
dir(str)
```

```
## ['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__'
```

```
print(dir(str.split))
```

```
## ['__call__', '__class__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__g
```

```
import re
print(dir(re.split))
```

```
## ['__annotations__', '__call__', '__class__', '__closure__', '__code__', '__defaults__', '__delattr__
```

```
import sys
# 模块存放路径
print(sys.path)
# 已安装的模块
sys.modules.keys()
```

```
dict_keys(['sys', 'builtins', '_frozen_importlib', '_imp', '_warnings', '_frozen_importlib_external',
'_io', 'marshal', 'posix', '_thread', '_weakref', 'time', 'zipimport', '_codecs', 'codecs',
'encodings.aliases', 'encodings.cp437', 'encodings', 'encodings.utf_8', '_signal', '__main__',
'encodings.latin_1', '_abc', 'abc', 'io', '_stat', 'stat', '_collections_abc', 'genericpath',
'posixpath', 'os.path', 'os', '_sitebuiltins', 'site', 'readline', 'atexit', 'rlcompleter'])
```

```
pip3 install virtualenv
virtualenv -p python3 <desired-path>
source <desired-path>/bin/activate
source /opt/virtualenv/tensorflow/bin/activate
```

- LaTeX 专家黄晨成写的译文 Matplotlib 教程
- 周沫凡 制作的莫烦 Python 系列视频教程之 Matplotlib 数据可视化神器
- 陈治兵维护的在线 Matplotlib 中文文档
- Sebastian Raschka 和 Vahid Mirjalili 合著的 Python Machine Learning (3rd Edition) [Raschka and Mirjalili, 2017]

编译书籍使用的 Python 3 模块有

```
pip3 list --format=columns
```

Package	Version
absl-py	1.0.0
astunparse	1.6.3
cachetools	5.0.0
certifi	2021.10.8
charset-normalizer	2.0.12
cycler	0.11.0
flatbuffers	2.0
fonttools	4.32.0
gast	0.5.3

© 基础工程

Package	Version
google-auth	2.6.4
google-auth-oauthlib	0.4.6
google-pasta	0.2.0
graphviz	0.8.4
grpcio	1.44.0
h5py	3.6.0
idna	3.3
importlib-metadata	4.11.3
joblib	1.1.0
kaleido	0.2.1
keras	2.8.0
Keras-Preprocessing	1.1.2
kiwisolver	1.4.2
libclang	13.0.0
Markdown	3.3.6
matplotlib	3.5.1
mpmath	1.2.1
mxnet	1.9.0
numpy	1.22.3
oauthlib	3.2.0
opt-einsum	3.3.0
packaging	21.3
pandas	1.4.2
patsy	0.5.2
Pillow	9.1.0
pip	20.0.2
pkg-resources	0.0.0
plotly	5.7.0
protobuf	3.20.0
pyasn1	0.4.8
pyasn1-modules	0.2.8
pyparsing	3.0.8
python-dateutil	2.8.2
pytz	2022.1
requests	2.27.1
requests-oauthlib	1.3.1
rsa	4.8
scikit-learn	1.0.2
scipy	1.8.0
setuptools	44.0.0
six	1.16.0
statsmodels	0.13.2
sympy	1.10.1

© 黄湘云

Package	Version
tenacity	8.0.1
tensorboard	2.8.0
tensorboard-data-server	0.6.1
tensorboard-plugin-wit	1.8.1
tensorflow	2.8.0
tensorflow-io-gcs-filesystem	0.24.0
termcolor	1.1.0
tf-estimator-nightly	2.8.0.dev2021122109
threadpoolctl	3.1.0
typing-extensions	4.1.1
urllib3	1.26.9
Werkzeug	2.1.1
wheel	0.34.2
wrapt	1.14.0
zipp	3.8.0

```
# 安装 Python 虚拟环境管理器 virtualenv
sudo dnf install -y python3-pip python3-virtualenv
# 创建虚拟环境
virtualenv -p /usr/bin/python3 $RETICULATE_PYTHON_ENV
# 激活虚拟环境
source $RETICULATE_PYTHON_ENV/bin/activate
# 将虚拟环境位置写入配置文件
echo "export RETICULATE_PYTHON_ENV=$HOME/.virtualenvs/r-tensorflow" >> ~/.bashrc
source ~/.bashrc
# 安装 numpy matplotlib 等模块
pip install -r requirements.txt
# 导出模块版本信息
pip freeze >> requirements.txt
```

```
import os
os.listdir('.git')
```

```
## ['info', 'description', 'hooks', 'config', 'objects', 'shallow', 'logs', 'HEAD', 'refs', 'branches',
```

多个代码块共享同一个 Python 进程

```
os.path
```

```
## <module 'posixpath' from '/usr/lib/python3.8/posixpath.py'>
```

matplotlib 绘图，支持交叉引用[2]，如下图所示

[2]早些时候，在 R Markdown 中设置 `python.reticulate = TRUE` 调用 reticulate 包，带来的副作用是不支持交叉引用的 https://d.cosx.org/d/420680-python-reticulate-true。RStudio 1.2 已经很好地集成了 reticulate，对 Python 的支持更加到位了 https://blog.rstudio.com/2018/10/09/rstudio-1-2-preview-reticulated-python/。截至本文写作时间 2022 年 04 月 13 日使用 reticulate 版本 1.24，本文没有对之前的版本进行测试。

© 基础设施

```
import matplotlib.pyplot as plt
from matplotlib import rcParams
# 其它可配置选项见 rcParams.keys()
rcParams.update({'font.size': 10, 'text.usetex': True})
# rcParams.update({'font.family':     ['sans-serif'],
#                  'font.monospace':  ['DejaVu Sans Mono'],
#                  'font.sans-serif': ['DejaVu Sans'],
#                  'font.serif':      ['DejaVu Serif']})
plt.switch_backend('agg')
plt.plot([0, 2, 1, 4])
plt.xlabel(r'Coord $x$')
plt.ylabel(r'Coord $y$')
plt.tight_layout()
plt.show()
```

有了 reticulate 包，我们可以把任意想要导入到 R 环境中的 Python 模块导进来，实现 R 与 Python 的数据交换和函数调用 [3]

```
os <- reticulate::import("os") # 导入 Python 模块
x <- os$listdir(".git") # 调用 os.listdir() 函数
x # 得到 python 中的向量 vector 或数组 array
```

```
##  [1] "info"        "description" "hooks"       "config"      "objects"
##  [6] "shallow"     "logs"        "HEAD"        "refs"        "branches"
## [11] "index"       "FETCH_HEAD"
```

```
# https://docs.bokeh.org/en/latest/docs/user_guide/quickstart.html#userguide-quickstart
from bokeh.plotting import figure, output_file, show
# 准备一些数据
x = [1, 2, 3, 4, 5]
y = [6, 7, 2, 4, 5]
# 将动态图形以静态 HTML 文件的方式保存
output_file("lines.html")
# 创建一个简单的图形，设置标题、x,y 轴标签
p = figure(title="simple line example", x_axis_label='x', y_axis_label='y')
# 添加一条折线，设置图例，线宽
p.line(x, y, legend_label="Temp.", line_width=2)
# 显示结果
show(p)
```

将静态图形嵌入到 R Markdown 中

```
htmltools::includeHTML("lines.html")
```

R 和 Python 之间的交互，Python 负责数据处理和建模，R 负责绘图，有些复杂的机器学习模型及其相关数据操作需要在 Python 中完成，数据集清理至数据框的形式后导入到 R 中，画各种静态或者动态图，这时候需要加载 reticulate 包，只是设置 `python.reticulate = TRUE` 还不够

[3] 朱俊辉的帖子 — 在 R 中使用 gluon https://d.cosx.org/d/419785-r-gluon

© 基础操作

提示

R Markdown 文档 [Xie et al., 2018] 中的 Python 代码块是由 knitr 包 [Xie, 2015] 负责调度处理的，展示 Matplotlib 绘图的结果使用了 reticulate 包 [Ushey et al., 2021] 提供的 Python 引擎而不是 knitr 自带的。

在 `knitr::opts_chunk` 中设置 `python.reticulate = TRUE` 意味着所有的 Python 代码块共享一个 Python Session，而 `python.reticulate = FALSE` 意味着使用 knitr 提供的 Python 引擎，所有的 Python 代码块独立运行。

pandas 读取数据，整理后由 reticulate 包传递给 R 环境中的 data.frame 对象，加载 ggplot2 绘图

```
library(ggplot2)
theme_set(theme_minimal())
library(patchwork)
p1 <- ggplot(py$iris2, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point(aes(color = Species)) +
  labs(title = "Call iris from Python")
p2 <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point(aes(color = Species)) +
  labs(title = "Call iris from R")
p1 + p2
```

图 G.12: 从 R 调用 Python 数据对象

以 NumPy 为例

```
import numpy as np
a = np.arange(15).reshape(3, 5)
a
```

```
## array([[ 0,  1,  2,  3,  4],
##        [ 5,  6,  7,  8,  9],
##        [10, 11, 12, 13, 14]])
```

```
a.shape
```

```
## (3, 5)
```

统计学习方法 ©

```
a.ndim
```

```
## 2
```

```
a.dtype.name
```

```
## 'int64'
```

```
a.itemsize
```

```
## 8
```

```
a.size
```

```
## 15
```

```
type(a)
```

```
## <class 'numpy.ndarray'>
```

```
b = np.array([6, 7, 8])
b
```

```
## array([6, 7, 8])
```

```
type(b)
```

```
## <class 'numpy.ndarray'>
```

```
a.transpose() @ b
```

```
## array([115, 136, 157, 178, 199])
```

Python 里面的点号 · 对应于 R 里面的 $

```
np <- import("numpy", convert=FALSE) # 导入 Python 模块
a <- np$arange(0, 15)$reshape(3L, 5L)
a
```

```
## [[ 0.  1.  2.  3.  4.]
##  [ 5.  6.  7.  8.  9.]
##  [10. 11. 12. 13. 14.]]
```

```
a$shape
```

```
## (3, 5)
```

```
a$ndim
```

```
## 2
```

```
a$dtype$name
```

```
## float64
```

```
a$itemsize
```

```
## 8
```

© 基础操作


```
a$size
```

```
## 15
```

```
a$ctypes
```

```
## <numpy.core._internal._ctypes>
```

```
a$dtype # data type 数据类型
```

```
## float64
```

```
a$astype
```

```
## <built-in method astype of numpy.ndarray>
```

```
builtins <- import_builtins() # Python 内建的函数，不需要导入第三方模块
builtins$type(a)
```

```
## <class 'numpy.ndarray'>
```

基本线性代数运算

```
a$transpose() # 转置
```

```
## [[ 0.  5. 10.]
##  [ 1.  6. 11.]
##  [ 2.  7. 12.]
##  [ 3.  8. 13.]
##  [ 4.  9. 14.]]
```

```
a$trace() # 迹
```

```
## 18.0
```

```
np$eye(2L) # 单位矩阵
```

```
## [[1. 0.]
##  [0. 1.]]
```

```
a$diagonal() # 对角
```

```
## [ 0.  6. 12.]
```

```
# 两个矩阵的乘法
b <- np$array(c(6, 7, 8, 9, 10))$reshape(5L, 1L)
b
```

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

© 王道漫画

```
b$shape
```

```
## (5, 1)
```

```
np$multiply(b$transpose(), a) # b 乘以 a
```

```
## [[  0.   7.  16.  27.  40.]
##  [ 30.  42.  56.  72.  90.]
##  [ 60.  77.  96. 117. 140.]]
```

Python 对象转化为 R 对象

```
py_to_r(b)
```

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

G.20 VBox 虚拟机

G.20.1 从命令行启动虚拟机

当前我的虚拟机里安装了两个系统 Fedora 29 和 CentOS 8.2

```
VBoxManage list vms
```

```
"Fedora 29" {d316fe8d-c053-4941-8a45-a59fd476898d}
"CentOS 8.2" {f1613f26-ea65-4f02-9cb6-6a79a758a60e}
```

以无图形化界面的方式启动虚拟机 CentOS 8.2

```
VBoxManage startvm "CentOS 8.2" --type headless
# 或者
VBoxHeadless --startvm "CentOS 8.2"
```

其它常用的命令还有

```
VBoxManage list runningvms # 列出运行中的虚拟机
VBoxManage controlvm "CentOS 8.2" acpipowerbutton # 关闭虚拟机，等价于点击系统关闭按钮，正常关机
VBoxManage controlvm "CentOS 8.2" poweroff # 关闭虚拟机，等价于直接关闭电源，非正常关机
VBoxManage controlvm "CentOS 8.2" pause # 暂停虚拟机的运行
VBoxManage controlvm "CentOS 8.2" resume # 恢复暂停的虚拟机
VBoxManage controlvm "CentOS 8.2" savestate # 保存当前虚拟机的运行状态
```

更多细节解释见 VBox 官方文档

© 黄湘云

G.21 Docker 虚拟环境

docker 创建云实例 rstudio DigitalOcean，docker 支持的驱动类型 https://docs.docker.com/machine/drivers/。Rocker 项目组提供的 shiny 容器 https://github.com/rocker-org/shiny 和构建过程 https://hub.docker.com/r/rocker/shiny/dockerfile

主机 80 端口映射给 shiny 容器 3838 端口

```
docker run --user shiny -d -p 80:3838 \
    -v /srv/shinyapps/:/srv/shiny-server/ \
    -v /srv/shinylog/:/var/log/shiny-server/ \
    rocker/shiny
```

shiny 服务器默认支持从 80 端口访问 http://localhost:80，shiny 应用放在目录 `/srv/shinyapps/appdir`，访问 Shiny 应用的位置 http://localhost/appdir/，使用 boot2docker 则访问 http://192.168.59.103:80/appdir/

Docker 相比虚拟机占用资源少，拉起来就可以用，虚拟机还需要各种环境配置，很多与 R 有关的项目现在都提供 Docker 镜像，大大方便了开发人员和数据分析师。当然 docker 的环境隔离性，对主机系统侵入小，即使挂了，再拉起来也就是了，安全性和可靠性高。

基于 The Rocker Project 快速构建数据分析环境，Rocker 项目 站在 Debian 和 R 的肩膀上，在 Docker 内配置众多数据分析和开发的工具，免去用户手动配置的复杂性。此事非有心者不能为之，因为需费时费力找寻依赖库，编译 R 包，还要尽可能地给 Docker 镜像减负，以便部署。如果想抢先试水的赶快去 Rocker 项目主页。

- 由 Dirk Eddelbuettel 等人担纲的 Rocker 项目，项目主页 和 Docker 镜像
- Wei-Chen Chen 等人的大数据编程项目 Programming with Big Data in R，项目主页 和 Docker 镜像
- 非常详细的 docker 笔记
- Dockerfile 最佳实践 https://docs.docker.com/develop/develop-images/dockerfile_best-practices/
- build 构建 https://docs.docker.com/engine/reference/builder/#usage

其它容器相关项目有 Singularity 和 Kubernetes 容器集群管理，更多参见高策的博客 https://gaocegege.com

本节介绍与本书配套的 VBox 镜像和 Docker 容器镜像，方便读者直接运行书籍原稿中的例子，尽量不限于软件环境配置的苦海中，因为对于大多数初学者来说，软件配置是一件不小的麻烦事。

本书依赖的 R 包和配置环境比较复杂，所以将整个运行环境打包成 Docker 镜像，方便读者重现，构建镜像的 Dockerfile 文件随同书籍源文件一起托管在 Github 上，方便读者研究。本地编译书籍只需三步走，先将存放在 Github 上的书籍项目克隆到本地，如果本地环境中没有 Git，你需要从它的官网 https://git-scm.com/ 下载安装适配本地系统的 Git 软件。

```
git clone https://github.com/XiangyunHuang/masr.git
```

然后在 Git Bash 的模拟终端器中，启动虚拟机，拉取准备好的镜像文件。为了方便读者重现本书的内容，特将书籍的编译环境打包成 Docker 镜像。在启动镜像前需要确保本地已经安装 Docker 软件 https://www.docker.com/products/docker-desktop，安装过程请看官网教程。

```
docker-machine.exe start default
docker pull xiangyunhuang/masr
```

©黄湘云

最后 cd 进入书籍项目所在目录，运行如下命令编译书籍

```
docker run --rm -u docker -v "/${PWD}://home/docker/workspace" \
  xiangyunhuang/masr make gitbook
```

编译成功后，可以在目录 _book/ 下看到生成的文件，点击文件 index.html 选择谷歌浏览器打开，不要使用 IE 浏览器，推荐使用谷歌浏览器获取最佳阅读体验，尽情地阅读吧！

如果你想了解编译书籍的环境和过程，我推荐你阅读随书籍源文件一起的 Dockerfile 文件，Docker Hub 是根据此文件构建的镜像，打包成功后，大约占用空间 2 Gb，本书在 RStudio IDE 下用 R Markdown [Xie et al., 2018] 编辑的，编译本书获得电子版还需要一些 R 包和软件。Pandoc https://pandoc.org/ 软件是系统 Fedora 30 仓库自带的，版本是 2.2.1，较新的 RStudio IDE 捆绑的 Pandoc 软件一般会高于此版本。如果你打算在本地系统上编译书籍，RStudio IDE 捆绑的 Pandoc 软件版本已经足够，当然你也可以在 https://github.com/jgm/pandoc/releases/latest 下载安装最新版本，此外，你还需参考书籍随附的 Dockerfile 文件配置 C++ 代码编译环境，安装所需的 R 包，并确保本地安装的版本不低于镜像内的版本。

镜像中已安装的 R 包列表可运行如下命令查看。

```
docker run --rm xiangyunhuang/masr \
  Rscript -e 'xfun::session_info(.packages(TRUE))'
```

Docker & Docker Machine & Docker Swarm

1. 容器与镜像的操作

   ```
   docker --version
   # Docker version 18.03.0-ce, build 0520e24302
   ```

   查看容器

   ```
   docker ps -a
   ```

   删除容器 docker rm 容器 ID，删除前要确认已经停止该容器的运行

   ```
   docker rm 6f932357e6ce
   ```

   查看镜像

   ```
   docker images
   ```

   删除镜像

   ```
   docker rmi 镜像 ID
   ```

   ```
   docker rmi 811281c54b23
   ```

2. 拉取镜像

   ```
   docker pull rocker/verse:latest
   ```

3. 运行容器

   ```
   docker run --name verse -d -p 8282:8080 -e ROOT=TRUE \
      -e USER=rstudio -e PASSWORD=cloud rocker/verse
   ```

   将主机端口 8282 映射给虚拟机/容器的 8080 端口，RStudio Server 默认使用的端口是 8787，因此改为 8080 需要修改 /etc/rstudio/rserver.conf 文件，添加

©黄海广

```
www-port=8080
```

然后重启 RStudio Server，之后可以在浏览器中登陆，登陆网址为 http://ip-addr:8080，其中 `ip-addr` 可在容器中运行如下一行命令获得

```
ip addr
```

更多关于服务器版本的 RStudio 介绍，请参考 https://docs.rstudio.com/ide/server-pro/access-and-security.html

Docker Machine

基本命令

- 查看 docker machine 版本信息

```
docker-machine --version
# docker-machine.exe version 0.14.0, build 89b8332
```

- 列出创建的虚拟机

```
# 启动前
docker-machine ls
# NAME      ACTIVE   DRIVER       STATE     URL   SWARM   DOCKER    ERRORS
# default   -        virtualbox   Stopped                 Unknown
# 启动后
docker-machine ls
# NAME      ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER        ER
# default   *        virtualbox   Running   tcp://192.168.99.100:2376           v18.03.0-ce
```

- 查看虚拟机 default 的 ip

```
docker-machine ip default
# 192.168.99.100
```

- 启动虚拟机

```
docker-machine start default
# Starting "default"...
# (default) Check network to re-create if needed...
# (default) Windows might ask for the permission to configure a dhcp server. Sometimes, such con
# (default) Waiting for an IP...
# Machine "default" was started.
# Waiting for SSH to be available...
# Detecting the provisioner...
# Started machines may have new IP addresses. You may need to re-run the `docker-machine env` co
```

- 进入 Docker 环境

```
docker-machine ssh default
```

```
                        ##         .
                  ## ## ##        ==
               ## ## ## ## ##    ===
```

© 基础设施

```
             /"""""""""""""""""\___/ ===
        ~~~ {~~ ~~~~ ~~~ ~~~~ ~~~ ~ /  ===- ~~~
             \_____ o           __/
               \    \         __/
                \____\_______/

 _                 _   ____     _            _
| |__   ___   ___ | |_|___ \ __| | ___   ___| | _____ _ __
| '_ \ / _ \ / _ \| __| __) / _` |/ _ \ / __| |/ / _ \ '__|
| |_) | (_) | (_) | |_ / __/ (_| | (_) | (__|   <  __/ |
|_.__/ \___/ \___/ \__|_____\__,_|\___/ \___|_|\_\___|_|
Boot2Docker version 18.03.0-ce, build HEAD : 404ee40 - Thu Mar 22 17:12:23 UTC 2018
Docker version 18.03.0-ce, build 0520e24
```

- 查看容器

```
docker ps -a
# CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
# 69e6929d269e        rocker/verse        "/init"             3 weeks ago         Exited (0) 10 days a
```

- 启动/停止容器

```
docker start verse
# verse
docker stop verse
# verse
```

- 查看虚拟机 default 的环境

```
docker-machine env default
# export DOCKER_TLS_VERIFY="1"
# export DOCKER_HOST="tcp://192.168.99.100:2376"
# export DOCKER_CERT_PATH="D:\Docker\machines\default"
# export DOCKER_MACHINE_NAME="default"
# export COMPOSE_CONVERT_WINDOWS_PATHS="true"
# # Run this command to configure your shell:
# # eval $("C:\Program Files\Docker Toolbox\docker-machine.exe" env default)
```

- 关闭虚拟机 default

```
docker-machine stop default
# Stopping "default"...
# Machine "default" was stopped.
```

更多详情见帮助文档 https://docs.docker.com/machine/get-started

©黄湘云

G.22 安装的 R 包

警告

本小节仅用于展示目前书籍写作过程中安装的 R 包依赖，不会出现在最终的书稿中

```
sessionInfo(sort(.packages(T)))
```

```
## R version 4.1.3 (2022-03-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.9.0
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.9.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
##  [1] base      compiler  datasets  graphics  grDevices grid      methods
##  [8] parallel  splines   stats     stats4    tcltk     tools     utils
##
## other attached packages:
##   [1] abind_1.4-5                agridat_1.20
##   [3] alabama_2015.3-1           arrow_7.0.0
##   [5] arules_1.7-3               askpass_1.1
##   [7] assertive.base_0.0-9       assertive.properties_0.0-4
##   [9] assertive.types_0.0-3      assertthat_0.2.1
##  [11] autoplotly_0.1.4           backports_1.4.1
##  [13] base64enc_0.1-3            bayesplot_1.9.0
##  [15] BB_2019.10-1               bbmle_1.0.24
##  [17] bdsmatrix_1.3-4            beanplot_1.2
##  [19] beeswarm_0.4.0             BH_1.78.0-0
##  [21] BiocGenerics_0.40.0        BiocManager_1.30.16
##  [23] BiocVersion_3.14.0         bit_4.0.4
##  [25] bit64_4.0.5                bitops_1.0-7
##  [27] blob_1.2.2                 bookdown_0.25
##  [29] boot_1.3-28                brew_1.0-7
##  [31] bridgesampling_1.1-2       brio_1.1.3
##  [33] brms_2.16.3                Brobdingnag_1.2-7
```

©黄海广

```
##  [35] broom_0.7.12              broom.mixed_0.2.9.3
##  [37] bslib_0.3.1               cachem_1.0.6
##  [39] callr_3.7.0               car_3.0-12
##  [41] carData_3.0-5             cellranger_1.1.0
##  [43] checkmate_2.0.0           circlize_0.4.14
##  [45] class_7.3-20              classInt_0.4-3
##  [47] cli_3.2.0                 clipr_0.8.0
##  [49] clue_0.3-60               cluster_2.1.3
##  [51] cmdstanr_0.5.0            coda_0.19-4
##  [53] codetools_0.2-18          colorspace_2.0-3
##  [55] colourpicker_1.1.1        commonmark_1.8.0
##  [57] ComplexHeatmap_2.10.0     config_0.3.1
##  [59] corrplot_0.92             countrycode_1.3.1
##  [61] cowplot_1.1.1             cpp11_0.4.2
##  [63] cranlogs_2.1.1            crayon_1.5.1
##  [65] credentials_1.3.2         crosstalk_1.2.0
##  [67] cubelyr_1.0.1             curl_4.3.2
##  [69] data.table_1.14.2         DBI_1.1.2
##  [71] dbplyr_2.1.1              dendextend_1.15.2
##  [73] Deriv_4.1.3               desc_1.4.1
##  [75] deSolve_1.31              devtools_2.4.3
##  [77] DiagrammeR_1.0.9          diffobj_0.3.5
##  [79] digest_0.6.29             distributional_0.3.0
##  [81] doParallel_1.0.17         downlit_0.4.0
##  [83] downloader_0.4            dplyr_1.0.8
##  [85] DT_0.22                   dtplyr_1.2.1
##  [87] dygraphs_1.1.1.6          e1071_1.7-9
##  [89] echarts4r_0.4.3           egg_0.4.5
##  [91] ellipsis_0.3.2            emo_0.0.0.9000
##  [93] equatiomatic_0.3.1        evaluate_0.15
##  [95] extrafont_0.17            extrafontdb_1.0
##  [97] fansi_1.0.3               farver_2.1.0
##  [99] fastmap_1.1.0             filehash_2.4-3
## [101] flexdashboard_0.5.2       fontawesome_0.2.2
## [103] fontcm_1.1                forcats_0.5.1
## [105] foreach_1.5.2             foreign_0.8-82
## [107] forge_0.2.0               formatR_1.11
## [109] formattable_0.2.1         Formula_1.2-4
## [111] fs_1.5.2                  furrr_0.2.3
## [113] future_1.24.0             gapminder_0.3.0
## [115] gargle_1.2.0              gclus_1.3.2
## [117] gdtools_0.2.4             generics_0.1.2
## [119] geoR_1.8-1                gert_1.6.0
## [121] GetoptLong_1.0.5          ggalluvial_0.12.3
## [123] gganimate_1.0.7           ggbeeswarm_0.6.0
```

©黄湘云

```
## [125] ggbump_0.1.99999          ggdendro_0.1.23
## [127] ggfittext_0.9.1           ggfortify_0.4.14
## [129] ggiraph_0.8.2             ggmosaic_0.3.3
## [131] ggnormalviolin_0.1.2      ggplot2_3.3.5
## [133] ggpubr_0.4.0              ggquiver_0.3.2
## [135] ggrepel_0.9.1             ggridges_0.5.3
## [137] ggsci_2.9                 ggsignif_0.6.3
## [139] ggstream_0.1.0            gh_1.3.0
## [141] gifski_1.4.3-1            git2r_0.30.1
## [143] gitcreds_0.1.1            glmmTMB_1.1.3
## [145] glmnet_4.1-3              GlobalOptions_0.1.2
## [147] globals_0.14.0            glue_1.6.2
## [149] googledrive_2.0.0         googlesheets4_1.0.0
## [151] graph_1.72.0              gridBase_0.4-7
## [153] gridExtra_2.3             gt_0.4.0
## [155] gtable_0.3.0              gtools_3.9.2
## [157] haven_2.4.3               heatmaply_1.3.0
## [159] here_1.0.1                hexbin_1.28.2
## [161] highr_0.9                 Hmisc_4.6-0
## [163] hms_1.1.1                 hrbrthemes_0.8.0
## [165] htmlTable_2.4.0           htmltools_0.5.2
## [167] htmlwidgets_1.5.4         httpuv_1.6.5
## [169] httr_1.4.2                hunspell_3.0.1
## [171] ids_1.0.1                 igraph_1.2.11
## [173] influenceR_0.1.0.1        ini_0.3.1
## [175] inline_0.3.19             IRanges_2.28.0
## [177] isoband_0.2.5             iterators_1.0.14
## [179] janeaustenr_0.1.5         janitor_2.1.0
## [181] jpeg_0.1-9                jquerylib_0.1.4
## [183] jsonlite_1.8.0            kableExtra_1.3.4
## [185] Kendall_2.2.1             kernlab_0.9-29
## [187] KernSmooth_2.23-20        knitr_1.38
## [189] labeling_0.4.2            later_1.3.0
## [191] lattice_0.20-45           latticeExtra_0.6-29
## [193] lazyeval_0.2.2            leaflet_2.1.1
## [195] leaflet.providers_1.9.0   leafletCN_0.2.1
## [197] lifecycle_1.0.1           lightgbm_3.3.2
## [199] listenv_0.8.0             lme4_1.1-28
## [201] loo_2.5.1                 lpSolve_5.6.15
## [203] lpSolveAPI_5.5.2.0-17.7   lubridate_1.8.0
## [205] lwgeom_0.2-8              magick_2.7.3
## [207] magrittr_2.0.3            mapdata_2.3.0
## [209] mapproj_1.2.8             maps_3.4.0
## [211] maptools_1.1-3            markdown_1.1
## [213] MASS_7.3-56               Matrix_1.4-1
```

©黄文海

```
## [215] MatrixModels_0.5-0         matrixStats_0.61.0
## [217] maxLik_1.5-2               mcmc_0.9-7
## [219] memoise_2.0.1              mgcv_1.8-40
## [221] microbenchmark_1.4.9       mime_0.12
## [223] miniUI_0.1.1.1             minqa_1.2.4
## [225] miscTools_0.6-26           modelr_0.1.8
## [227] munsell_0.5.0              mvtnorm_1.1-3
## [229] networkD3_0.4              nleqslv_3.3.2
## [231] nlme_3.1-157               nlmeODE_1.1
## [233] nloptr_2.0.0               nnet_7.3-17
## [235] nomnoml_0.2.5              numDeriv_2016.8-1.1
## [237] odbc_1.3.3                 openssl_2.0.0
## [239] packagemetrics_0.0.1.9001  packrat_0.7.0
## [241] palmerpenguins_0.1.0       parallelly_1.30.0
## [243] patchwork_1.1.1            pbkrtest_0.5.1
## [245] PBSddesolve_1.12.6         pdftools_3.1.1
## [247] pdist_1.2                  pillar_1.7.0
## [249] pkgbuild_1.3.1             pkgconfig_2.0.3
## [251] pkgload_1.2.4              plogr_0.2.0
## [253] plotly_4.10.0              plyr_1.8.7
## [255] png_0.1-7                  polynom_1.4-0
## [257] posterior_1.2.1            praise_1.0.0
## [259] prettydoc_0.4.1            prettyunits_1.1.1
## [261] PrevMap_1.5.4              processx_3.5.3
## [263] productplots_0.1.1         progress_1.2.2
## [265] promises_1.2.0.1           proxy_0.4-26
## [267] ps_1.6.0                   pspearman_0.3-0
## [269] purrr_0.3.4                pwr_1.3-0
## [271] qap_0.1-1                  qpdf_1.1
## [273] quadprog_1.5-8             quantmod_0.4.18
## [275] quantreg_5.88              r2d3_0.2.6
## [277] R6_2.5.1                   RandomFields_3.3.14
## [279] RandomFieldsUtils_1.2.3    randomForest_4.7-1
## [281] rappdirs_0.3.3             raster_3.5-15
## [283] rasterly_0.2.0             rasterVis_0.51.2
## [285] rcmdcheck_1.4.0            RColorBrewer_1.1-2
## [287] Rcpp_1.0.8.3               RcppArmadillo_0.10.8.1.0
## [289] RcppEigen_0.3.3.9.1        RcppParallel_5.1.5
## [291] RcppTOML_0.1.7             reactable_0.2.3
## [293] reactR_0.4.4               ReacTran_1.4.3.1
## [295] readr_2.1.2                readxl_1.4.0
## [297] registry_0.5-1             rematch_1.0.1
## [299] rematch2_2.1.2             remotes_2.4.2
## [301] renv_0.15.4                reprex_2.0.1
## [303] reshape2_1.4.4             reticulate_1.24
```

©黄湘云

```
## [305] rgdal_1.5-29                rgeos_0.5-9
## [307] RgoogleMaps_1.4.5.3         Rgraphviz_2.38.0
## [309] rJava_1.0-6                 rjson_0.2.21
## [311] rlang_1.0.2                 rmarkdown_2.13
## [313] ROI_1.0-0                   ROI.plugin.alabama_1.0-0
## [315] ROI.plugin.lpsolve_1.0-1    ROI.plugin.nloptr_1.0-0
## [317] ROI.plugin.quadprog_1.0-0   ROI.plugin.scs_1.1-1
## [319] rootSolve_1.8.2.3           roxygen2_7.1.2
## [321] rpart_4.1.16                rprojroot_2.0.2
## [323] rsconnect_0.8.25            RSQLite_2.2.11
## [325] rstan_2.26.9                rstantools_2.1.1
## [327] rstatix_0.7.0               rstudioapi_0.13
## [329] Rttf2pt1_1.3.10             rversions_2.1.1
## [331] rvest_1.0.2                 s2_1.0.7
## [333] S4Vectors_0.32.4            sandwich_3.0-1
## [335] sass_0.4.1                  scales_1.1.1
## [337] scatterplot3d_0.3-41        scs_3.0-0
## [339] selectr_0.4-2               seriation_1.3.5
## [341] sessioninfo_1.2.2           sf_1.0-7
## [343] sfarrow_0.4.1               shades_1.4.0
## [345] shape_1.4.6                 shiny_1.7.1
## [347] shinyjs_2.1.0               shinystan_2.6.0
## [349] shinythemes_1.2.0           showtext_0.9-5
## [351] showtextdb_3.0              Sim.DiffProc_4.8
## [353] slam_0.1-50                 sm_2.2-5.7
## [355] snakecase_0.11.0            SnowballC_0.7.0
## [357] sourcetools_0.1.7           sp_1.4-6
## [359] sparkline_2.0               sparklyr_1.7.5
## [361] SparseM_1.81                spatial_7.3-15
## [363] spDataLarge_2.0.5           splancs_2.01-42
## [365] splines2_0.4.5              stackr_0.0.0.9000
## [367] StanHeaders_2.26.9          stars_0.5-5
## [369] stringi_1.7.6               stringr_1.4.0
## [371] SuppDists_1.1-9.7           survival_3.3-1
## [373] svglite_2.1.0               symengine_0.1.6
## [375] sys_3.4                     sysfonts_0.8.8
## [377] systemfonts_1.0.4           tensorA_0.36.2
## [379] tensorflow_2.8.0            terra_1.5-21
## [381] testthat_3.1.3              tfautograph_0.3.2
## [383] tfruns_1.5.0                threejs_0.3.3
## [385] tibble_3.1.6                tidyr_1.2.0
## [387] tidyselect_1.1.2            tidytext_0.3.2
## [389] tidyverse_1.3.1             tikzDevice_0.12.3.1
## [391] timeline_0.9                timelineS_0.1.1
## [393] tint_0.1.3                  tinytex_0.38
```

© 黄湘云

```
## [395] TMB_1.8.1                 tokenizers_0.2.1
## [397] transformr_0.1.3          treemap_2.4-3
## [399] treemapify_2.5.5          truncnorm_1.0-8
## [401] TSP_1.2-0                 TTR_0.24.3
## [403] tweenr_1.0.2              tzdb_0.3.0
## [405] units_0.8-0               usethis_2.1.5
## [407] usmapdata_0.1.0           utf8_1.2.2
## [409] uuid_1.0-4                V8_4.1.0
## [411] vctrs_0.4.0               vioplot_0.3.7
## [413] vipor_0.4.5               viridis_0.6.2
## [415] viridisLite_0.4.0         visNetwork_2.1.0
## [417] vistime_1.2.1             vroom_1.5.7
## [419] waldo_0.4.0               webshot_0.5.2
## [421] whisker_0.4               withr_2.5.0
## [423] wk_0.6.0                  xfun_0.30
## [425] xgboost_1.5.2.1           xkcd_0.0.6
## [427] xml2_1.3.3                xopen_1.0.0
## [429] xtable_1.8-4              xts_0.12.1
## [431] yaml_2.3.5                zip_2.2.0
## [433] zoo_1.8-9
```

```
library(magrittr)
pdb <- tools::CRAN_package_db()
pkg <- subset(desc::desc_get_deps(), subset = type == "Imports", select = "package", drop = TRUE)
pkg <- tools::package_dependencies(packages = pkg, db = pdb, recursive = FALSE) %>% # 是否包含递归依赖
  unlist() %>%
  as.vector() %>%
  c(., pkg) %>%
  unique() %>%
  sort()

pkg_quote <- c(
  "Armadillo", "Rcpp", "R", "Stan", "DataTables", "Dygraphs", "ggplot2",
  "Grobs", "Geospatial", "Eigen", "Sundown", "plog", "TeX Live", "Tidyverse",
  "LaTeX", "ADMB", "matplotlib", "Yihui Xie", "With", "Highcharts",
  "kable", "plotly.js", "Python", "Formattable"
)
# 单引号
pkg_regexp <- paste("'(", paste(pkg_quote, collapse = "|"), ")'", sep = "")
# R 包列表
subset(pdb,
  subset = !duplicated(pdb$Package) & Package %in% pkg,
  select = c("Package", "Version", "Title")
) %>%
  transform(.,
    Title = gsub("(\\\n)", " ", Title),
```

© 黄湘云


```
    Package = paste("**", Package, "**", sep = "")
  ) %>%
  transform(., Title = gsub(pkg_regexp, "\\1", Title)) %>%
  transform(., Title = gsub('"(Grid)"', "\\1", Title)) %>%
  knitr::kable(.,
    caption = "依赖的 R 包", format = "pandoc",
    booktabs = TRUE, row.names = FALSE
  )
```

表 G.4: 依赖的 R 包

Package	Version	Title
abind	1.4-5	Combine Multidimensional Arrays
agridat	1.20	Agricultural Datasets
alabama	2015.3-1	Constrained Nonlinear Optimization
arrow	7.0.0	Integration to 'Apache' 'Arrow'
arules	1.7-3	Mining Association Rules and Frequent Itemsets
assertive.types	0.0-3	Assertions to Check Types of Variables
assertthat	0.2.1	Easy Pre and Post Assertions
autoplotly	0.1.4	Automatic Generation of Interactive Visualizations for Statistical Results
backports	1.4.1	Reimplementations of Functions Introduced Since R-3.0.0
base64enc	0.1-3	Tools for base64 encoding
bayesplot	1.9.0	Plotting for Bayesian Models
bbmle	1.0.24	Tools for General Maximum Likelihood Estimation
bdsmatrix	1.3-4	Routines for Block Diagonal Symmetric Matrices
beanplot	1.3.1	Visualization via Beanplots (Like Boxplot/Stripchart/Violin Plot)
beeswarm	0.4.0	The Bee Swarm Plot, an Alternative to Stripchart
BH	1.78.0-0	Boost C++ Header Files
BiocManager	1.30.16	Access the Bioconductor Project Package Repository
bit64	4.0.5	A S3 Class for Vectors of 64bit Integers
bitops	1.0-7	Bitwise Operations
blob	1.2.3	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')
bookdown	0.25	Authoring Books and Technical Documents with R Markdown
boot	1.3-28	Bootstrap Functions (Originally by Angelo Canty for S)
bridgesampling	1.1-2	Bridge Sampling for Marginal Likelihoods and Bayes Factors
brio	1.1.3	Basic R Input Output
brms	2.16.3	Bayesian Regression Models using Stan
broom	0.7.12	Convert Statistical Objects into Tidy Tibbles
broom.mixed	0.2.9.3	Tidying Methods for Mixed Models
bslib	0.3.1	Custom 'Bootstrap' 'Sass' Themes for 'shiny' and 'rmarkdown'
cachem	1.0.6	Cache R Objects with Automatic Pruning
callr	3.7.0	Call R from R
checkmate	2.0.0	Fast and Versatile Argument Checks
classInt	0.4-3	Choose Univariate Class Intervals
cli	3.2.0	Helpers for Developing Command Line Interfaces

Package	Version	Title
coda	0.19-4	Output Analysis and Diagnostics for MCMC
colorspace	2.0-3	A Toolbox for Manipulating and Assessing Colors and Palettes
commonmark	1.8.0	High Performance CommonMark and Github Markdown Rendering in R
config	0.3.1	Manage Environment Specific Configuration Values
corrplot	0.92	Visualization of a Correlation Matrix
countrycode	1.3.1	Convert Country Names and Country Codes
cowplot	1.1.1	Streamlined Plot Theme and Plot Annotations for ggplot2
cpp11	0.4.2	A C++11 Interface for R's C Interface
crayon	1.5.1	Colored Terminal Output
crosstalk	1.2.0	Inter-Widget Interactivity for HTML Widgets
cubelyr	1.0.1	A Data Cube 'dplyr' Backend
curl	4.3.2	A Modern and Flexible Web Client for R
data.table	1.14.2	Extension of `data.frame`
DBI	1.1.2	R Database Interface
dbplyr	2.1.1	A 'dplyr' Back End for Databases
dendextend	1.15.2	Extending 'dendrogram' Functionality in R
Deriv	4.1.3	Symbolic Differentiation
desc	1.4.1	Manipulate DESCRIPTION Files
deSolve	1.31	Solvers for Initial Value Problems of Differential Equations ('ODE', 'DAE', 'DDE')
devtools	2.4.3	Tools to Make Developing R Packages Easier
DiagrammeR	1.0.9	Graph/Network Visualization
digest	0.6.29	Create Compact Hash Digests of R Objects
downlit	0.4.0	Syntax Highlighting and Automatic Linking
downloader	0.4	Download Files over HTTP and HTTPS
dplyr	1.0.8	A Grammar of Data Manipulation
DT	0.22	A Wrapper of the JavaScript Library DataTables
dtplyr	1.2.1	Data Table Back-End for 'dplyr'
echarts4r	0.4.3	Create Interactive Graphs with 'Echarts JavaScript' Version 5
egg	0.4.5	Extensions for ggplot2: Custom Geom, Custom Themes, Plot Alignment, Labelled Pa
ellipsis	0.3.2	Tools for Working with ...
equatiomatic	0.3.1	Transform Models into LaTeX Equations
evaluate	0.15	Parsing and Evaluation Tools that Provide More Details than the Default
extrafont	0.18	Tools for Using Fonts
extrafontdb	1.0	Package for holding the database for the extrafont package
fansi	1.0.3	ANSI Control Sequence Aware String Functions
fastmap	1.1.0	Fast Data Structures
filehash	2.4-3	Simple Key-Value Database
fontawesome	0.2.2	Easily Work with 'Font Awesome' Icons
fontcm	1.1	Computer Modern font for use with extrafont package
forcats	0.5.1	Tools for Working with Categorical Variables (Factors)
foreach	1.5.2	Provides Foreach Looping Construct
forge	0.2.0	Casting Values into Shape
formatR	1.12	Format R Code Automatically

© 黄海广

Package	Version	Title
fs	1.5.2	Cross-Platform File System Operations Based on 'libuv'
future	1.24.0	Unified Parallel and Distributed Processing in R for Everyone
gapminder	0.3.0	Data from Gapminder
gdtools	0.2.4	Utilities for Graphical Rendering
generics	0.1.2	Common S3 Generics not Provided by Base R Methods Related to Model Fittin
geoR	1.8-1	Analysis of Geostatistical Data
ggalluvial	0.12.3	Alluvial Plots in ggplot2
gganimate	1.0.7	A Grammar of Animated Graphics
ggbeeswarm	0.6.0	Categorical Scatter (Violin Point) Plots
ggbump	0.1.0	Bump Chart and Sigmoid Curves
ggfittext	0.9.1	Fit Text Inside a Box in ggplot2
ggfortify	0.4.14	Data Visualization Tools for Statistical Analysis Results
ggmosaic	0.3.3	Mosaic Plots in the ggplot2 Framework
ggnormalviolin	0.1.2	A ggplot2 Extension to Make Normal Violin Plots
ggplot2	3.3.5	Create Elegant Data Visualisations Using the Grammar of Graphics
ggpubr	0.4.0	ggplot2 Based Publication Ready Plots
ggquiver	0.3.2	Quiver Plots for ggplot2
ggrepel	0.9.1	Automatically Position Non-Overlapping Text Labels with ggplot2
ggridges	0.5.3	Ridgeline Plots in ggplot2
ggsci	2.9	Scientific Journal and Sci-Fi Themed Color Palettes for ggplot2
ggsignif	0.6.3	Significance Brackets for ggplot2
ggstream	0.1.0	Create Streamplots in ggplot2
gifski	1.6.6-1	Highest Quality GIF Encoder
git2r	0.30.1	Provides Access to Git Repositories
glmmTMB	1.1.3	Generalized Linear Mixed Models using Template Model Builder
glmnet	4.1-3	Lasso and Elastic-Net Regularized Generalized Linear Models
globals	0.14.0	Identify Global Objects in R Expressions
glue	1.6.2	Interpreted String Literals
googledrive	2.0.0	An Interface to Google Drive
googlesheets4	1.0.0	Access Google Sheets using the Sheets API V4
gridBase	0.4-7	Integration of base and grid graphics
gridExtra	2.3	Miscellaneous Functions for Grid Graphics
gt	0.4.0	Easily Create Presentation-Ready Display Tables
gtable	0.3.0	Arrange Grobs in Tables
haven	2.4.3	Import and Export 'SPSS', 'Stata' and 'SAS' Files
heatmaply	1.3.0	Interactive Cluster Heat Maps Using 'plotly' and ggplot2
here	1.0.1	A Simpler Way to Find Your Files
hexbin	1.28.2	Hexagonal Binning Routines
highr	0.9	Syntax Highlighting for R Source Code
Hmisc	4.6-0	Harrell Miscellaneous
hms	1.1.1	Pretty Time of Day
hrbrthemes	0.8.0	Additional Themes, Theme Components and Utilities for ggplot2
htmltools	0.5.2	Tools for HTML

统计出版社 ©

Package	Version	Title
htmlwidgets	1.5.4	HTML Widgets for R
httpuv	1.6.5	HTTP and WebSocket Server Library
httr	1.4.2	Tools for Working with URLs and HTTP
igraph	1.3.0	Network Analysis and Visualization
influenceR	0.1.0.1	Software Tools to Quantify Structural Importance of Nodes in a Network
inline	0.3.19	Functions to Inline C, C++, Fortran Function Calls from R
isoband	0.2.5	Generate Isolines and Isobands from Regularly Spaced Elevation Grids
jquerylib	0.1.4	Obtain 'jQuery' as an HTML Dependency Object
jsonlite	1.8.0	A Simple and Robust JSON Parser and Generator for R
kableExtra	1.3.4	Construct Complex Table with kable and Pipe Syntax
Kendall	2.2.1	Kendall Rank Correlation and Mann-Kendall Trend Test
knitr	1.38	A General-Purpose Package for Dynamic Report Generation in R
later	1.3.0	Utilities for Scheduling Functions to Execute Later with Event Loops
lattice	0.20-45	Trellis Graphics for R
latticeExtra	0.6-29	Extra Graphical Utilities Based on Lattice
lazyeval	0.2.2	Lazy (Non-Standard) Evaluation
leaflet	2.1.1	Create Interactive Web Maps with the JavaScript 'Leaflet' Library
leaflet.providers	1.9.0	Leaflet Providers
leafletCN	0.2.1	An R Gallery for China and Other Geojson Choropleth Map in Leaflet
lifecycle	1.0.1	Manage the Life Cycle of your Package Functions
lightgbm	3.3.2	Light Gradient Boosting Machine
lme4	1.1-29	Linear Mixed-Effects Models using Eigen and S4
loo	2.5.1	Efficient Leave-One-Out Cross-Validation and WAIC for Bayesian Models
lpSolve	5.6.15	Interface to 'Lp_solve' v. 5.5 to Solve Linear/Integer Programs
lpSolveAPI	5.5.2.0-17.7	R Interface to 'lp_solve' Version 5.5.2.0
lubridate	1.8.0	Make Dealing with Dates a Little Easier
lwgeom	0.2-8	Bindings to Selected 'liblwgeom' Functions for Simple Features
magick	2.7.3	Advanced Graphics and Image-Processing in R
magrittr	2.0.3	A Forward-Pipe Operator for R
mapdata	2.3.0	Extra Map Databases
mapproj	1.2.8	Map Projections
maps	3.4.0	Draw Geographical Maps
markdown	1.1	Render Markdown with the C Library Sundown
MASS	7.3-56	Support Functions and Datasets for Venables and Ripley's MASS
Matrix	1.4-1	Sparse and Dense Matrix Classes and Methods
MatrixModels	0.5-0	Modelling with Sparse and Dense Matrices
matrixStats	0.61.0	Functions that Apply to Rows and Columns of Matrices (and to Vectors)
maxLik	1.5-2	Maximum Likelihood Estimation and Related Tools
mcmc	0.9-7	Markov Chain Monte Carlo
memoise	2.0.1	'Memoisation' of Functions
mgcv	1.8-40	Mixed GAM Computation Vehicle with Automatic Smoothness Estimation
mime	0.12	Map Filenames to MIME Types
minqa	1.2.4	Derivative-free optimization algorithms by quadratic approximation

© 中国铁道出版社

Package	Version	Title
modelr	0.1.8	Modelling Functions that Work with the Pipe
mvtnorm	1.1-3	Multivariate Normal and t Distributions
networkD3	0.4	D3 JavaScript Network Graphs from R
nleqslv	3.3.2	Solve Systems of Nonlinear Equations
nlme	3.1-157	Linear and Nonlinear Mixed Effects Models
nloptr	2.0.0	R Interface to NLopt
nomnoml	0.2.5	Sassy 'UML' Diagrams
numDeriv	2016.8-1.1	Accurate Numerical Derivatives
odbc	1.3.3	Connect to ODBC Compatible Databases (using the DBI Interface)
openssl	2.0.0	Toolkit for Encryption, Signatures and Certificates Based on OpenSSL
palmerpenguins	0.1.0	Palmer Archipelago (Antarctica) Penguin Data
patchwork	1.1.1	The Composer of Plots
pdftools	3.1.1	Text Extraction, Rendering and Converting of PDF Documents
pdist	1.2	Partitioned Distance Function
pillar	1.7.0	Coloured Formatting for Columns
pkgbuild	1.3.1	Find Tools Needed to Build R Packages
pkgconfig	2.0.3	Private Configuration for R Packages
pkgload	1.2.4	Simulate Package Installation and Attach
plogr	0.2.0	The plog C++ Logging Library
plotly	4.10.0	Create Interactive Web Graphics via plotly.js
plyr	1.8.7	Tools for Splitting, Applying and Combining Data
png	0.1-7	Read and write PNG images
polynom	1.4-1	A Collection of Functions to Implement a Class for Univariate Polynomial Man
posterior	1.2.1	Tools for Working with Posterior Distributions
prettydoc	0.4.1	Creating Pretty Documents from R Markdown
PrevMap	1.5.4	Geostatistical Modelling of Spatially Referenced Prevalence Data
processx	3.5.3	Execute and Control System Processes
productplots	0.1.1	Product Plots for R
progress	1.2.2	Terminal Progress Bars
promises	1.2.0.1	Abstractions for Promise-Based Asynchronous Programming
pspearman	0.3-0	Spearman's rank correlation test
purrr	0.3.4	Functional Programming Tools
pwr	1.3-0	Basic Functions for Power Analysis
qpdf	1.1	Split, Combine and Compress PDF Files
quadprog	1.5-8	Functions to Solve Quadratic Programming Problems
quantreg	5.88	Quantile Regression
r2d3	0.2.6	Interface to 'D3' Visualizations
R6	2.5.1	Encapsulated Classes with Reference Semantics
RandomFields	3.3.14	Simulation and Analysis of Random Fields
rappdirs	0.3.3	Application Directories: Determine Where to Save Data, Caches, and Logs
raster	3.5-15	Geographic Data Analysis and Modeling
rasterly	0.2.0	Easily and Rapidly Generate Raster Image Data with Support for 'Plotly.js'
rasterVis	0.51.2	Visualization Methods for Raster Data

Package	Version	Title
rcmdcheck	1.4.0	Run 'R CMD check' from R and Capture Results
RColorBrewer	1.1-3	ColorBrewer Palettes
Rcpp	1.0.8.3	Seamless R and C++ Integration
RcppArmadillo	0.11.0.0.0	Rcpp Integration for the Armadillo Templated Linear Algebra Library
RcppEigen	0.3.3.9.2	Rcpp Integration for the Eigen Templated Linear Algebra Library
RcppParallel	5.1.5	Parallel Programming Tools for Rcpp
RcppTOML	0.1.7	Rcpp Bindings to Parser for Tom's Obvious Markup Language
reactable	0.2.3	Interactive Data Tables Based on 'React Table'
reactR	0.4.4	React Helpers
readr	2.1.2	Read Rectangular Text Data
readxl	1.4.0	Read Excel Files
registry	0.5-1	Infrastructure for R Package Registries
remotes	2.4.2	R Package Installation from Remote Repositories, Including 'GitHub'
reprex	2.0.1	Prepare Reproducible Example Code via the Clipboard
reshape2	1.4.4	Flexibly Reshape Data: A Reboot of the Reshape Package
reticulate	1.24	Interface to Python
rgdal	1.5-30	Bindings for the Geospatial Data Abstraction Library
rgeos	0.5-9	Interface to Geometry Engine - Open Source ('GEOS')
rlang	1.0.2	Functions for Base Types and Core R and Tidyverse Features
rmarkdown	2.13	Dynamic Documents for R
ROI	1.0-0	R Optimization Infrastructure
ROI.plugin.alabama	1.0-0	'alabama' Plug-in for the R Optimization Infrastructure
ROI.plugin.lpsolve	1.0-1	'lp_solve' Plugin for the R Optimization Infrastructure
ROI.plugin.nloptr	1.0-0	'nloptr' Plug-in for the R Optimization Infrastructure
ROI.plugin.quadprog	1.0-0	'quadprog' Plug-in for the R Optimization Infrastructure
ROI.plugin.scs	1.1-1	'SCS' Plug-in for the R Optimization Infrastructure
rootSolve	1.8.2.3	Nonlinear Root Finding, Equilibrium and Steady-State Analysis of Ordinary Different
roxygen2	7.1.2	In-Line Documentation for R
rprojroot	2.0.3	Finding Files in Project Subdirectories
RSQLite	2.2.12	SQLite Interface for R
rstan	2.21.5	R Interface to Stan
rstantools	2.2.0	Tools for Developing R Packages Interfacing with Stan
rstatix	0.7.0	Pipe-Friendly Framework for Basic Statistical Tests
rstudioapi	0.13	Safely Access the RStudio API
Rttf2pt1	1.3.10	'ttf2pt1' Program
rversions	2.1.1	Query R Versions, Including 'r-release' and 'r-oldrel'
rvest	1.0.2	Easily Harvest (Scrape) Web Pages
s2	1.0.7	Spherical Geometry Operators Using the S2 Geometry Library
sass	0.4.1	Syntactically Awesome Style Sheets ('Sass')
scales	1.1.1	Scale Functions for Visualization
scatterplot3d	0.3-41	3D Scatter Plot
scs	3.0-0	Splitting Conic Solver
seriation	1.3.5	Infrastructure for Ordering Objects Using Seriation

© 谢益辉

Package	Version	Title
sessioninfo	1.2.2	R Session Information
sf	1.0-7	Simple Features for R
sfarrow	0.4.1	Read/Write Simple Feature Objects ('sf') with 'Apache' 'Arrow'
shape	1.4.6	Functions for Plotting Graphical Shapes, Colors
shiny	1.7.1	Web Application Framework for R
shinystan	2.6.0	Interactive Visual and Numerical Diagnostics and Posterior Analysis for Bayesi
showtext	0.9-5	Using Fonts More Easily in R Graphs
showtextdb	3.0	Font Files for the 'showtext' Package
slam	0.1-50	Sparse Lightweight Arrays and Matrices
sm	2.2-5.7	Smoothing Methods for Nonparametric Regression and Density Estimation
sourcetools	0.1.7	Tools for Reading, Tokenizing and Parsing R Code
sp	1.4-6	Classes and Methods for Spatial Data
sparkline	2.0	'jQuery' Sparkline 'htmlwidget'
sparklyr	1.7.5	R Interface to Apache Spark
SparseM	1.81	Sparse Linear Algebra
splancs	2.01-42	Spatial and Space-Time Point Pattern Analysis
splines2	0.4.5	Regression Spline Functions and Classes
StanHeaders	2.21.0-7	C++ Header Files for Stan
stars	0.5-5	Spatiotemporal Arrays, Raster and Vector Data Cubes
stringi	1.7.6	Character String Processing Facilities
stringr	1.4.0	Simple, Consistent Wrappers for Common String Operations
SuppDists	1.1-9.7	Supplementary Distributions
survival	3.3-1	Survival Analysis
svglite	2.1.0	An 'SVG' Graphics Device
symengine	0.1.6	Interface to the 'SymEngine' Library
sysfonts	0.8.8	Loading Fonts into R
tensorflow	2.8.0	R Interface to 'TensorFlow'
terra	1.5-21	Spatial Data Analysis
testthat	3.1.3	Unit Testing for R
tfautograph	0.3.2	Autograph R for 'Tensorflow'
tfruns	1.5.0	Training Run Tools for 'TensorFlow'
tibble	3.1.6	Simple Data Frames
tidyr	1.2.0	Tidy Messy Data
tidyselect	1.1.2	Select from a Set of Strings
tidyverse	1.3.1	Easily Install and Load the Tidyverse
tikzDevice	0.12.3.1	R Graphics Output in LaTeX Format
timeline	0.9	Timelines for a Grammar of Graphics
timelineS	0.1.1	Timeline and Time Duration-Related Tools
tint	0.1.3	'tint' is not 'Tufte'
tinytex	0.38	Helper Functions to Install and Maintain TeX Live, and Compile LaTeX Docum
TMB	1.8.1	Template Model Builder: A General Random Effect Tool Inspired by ADMB
transformr	0.1.3	Polygon and Path Transformations
treemap	2.4-3	Treemap Visualization

Package	Version	Title
treemapify	2.5.5	Draw Treemaps in ggplot2
truncnorm	1.0-8	Truncated Normal Distribution
TSP	1.2-0	Traveling Salesperson Problem (TSP)
tweenr	1.0.2	Interpolate Data for Smooth Animations
units	0.8-0	Measurement Units for R Vectors
usethis	2.1.5	Automate Package and Project Setup
uuid	1.0-4	Tools for Generating and Handling of UUIDs
V8	4.1.0	Embedded JavaScript and WebAssembly Engine for R
vctrs	0.4.0	Vector Helpers
vioplot	0.3.7	Violin Plot
vipor	0.4.5	Plot Categorical Data Using Quasirandom Noise and Density Estimates
viridis	0.6.2	Colorblind-Friendly Color Maps for R
viridisLite	0.4.0	Colorblind-Friendly Color Maps (Lite Version)
visNetwork	2.1.0	Network Visualization using 'vis.js' Library
vistime	1.2.1	Pretty Timelines in R
webshot	0.5.2	Take Screenshots of Web Pages
withr	2.5.0	Run Code With Temporarily Modified Global State
xfun	0.30	Supporting Functions for Packages Maintained by Yihui Xie
xgboost	1.5.2.1	Extreme Gradient Boosting
xkcd	0.0.6	Plotting ggplot2 Graphics in an XKCD Style
xml2	1.3.3	Parse XML
xtable	1.8-4	Export Tables to LaTeX or HTML
yaml	2.3.5	Methods to Convert R Data to YAML and Back
zoo	1.8-9	S3 Infrastructure for Regular and Irregular Time Series (Z's Ordered Observations)

提示

本书意欲覆盖的内容

```
inla_pdb <- data.frame(
  Package = "INLA",
  Title = paste(
    "Full Bayesian Analysis of Latent Gaussian Models",
    "using Integrated Nested Laplace Approximations"
  )
)
pkgs <- c(
  "ggplot2", "cowplot", "patchwork", "rgl", "MASS", "nlme", "mgcv",
  "lme4", "gee", "gam", "gamm4", "cgam", "cglm", "pscl",
  "GLMMadaptive", "gee4", "geoR", "LaplacesDemon", "glmnet",
  "betareg", "quantreg", "agridat", "moments", "R2BayesX",
  "geoRglm", "spaMM", "spBayes", "CARBayes", "PrevMap",
  "FRK", "lgcp", "HSAR", "spNNGP", "MuMIn", "BANOVA",
  "rpql", "QGglmm", "glmmsr", "glmmboot", "glmm",
```

© 统计出版社


```
  "glmmML", "glmmEP", "r2glmm", "hglm", "glmmLasso",
  "blme", "MCMCglmm", "MCMCpack", "glmmTMB", "geepack",
  "glmmfields", "rstan", "rstanarm", "brms", "greta",
  "BayesX", "Boom", "nimble", "rjags", "R2OpenBUGS",
  "R2BayesX", "BoomSpikeSlab", "inlabru", "INLABMA",
  "lmtest", "VGAM", "plotly", "leaflet", "LatticeKrig"
)
pdb <- tools::CRAN_package_db()
book_pdb <- subset(pdb,
  subset = !duplicated(pdb$Package) & Package %in% pkgs,
  select = c("Package", "Title")
)
book_pdb <- rbind.data.frame(book_pdb, inla_pdb)
book_pdb$Title <- gsub("(\\\n)", " ", book_pdb$Title)
book_pdb$Title <- gsub("'(Armadillo|BayesX|Eigen|ggplot2|lme4|mgcv|Stan|Leaflet|plotly.js)'", "\\1",
book_pdb$Package <- paste("**", book_pdb$Package, "**", sep = "")
knitr::kable(book_pdb,
  caption = "本书使用的 R 包", format = "pandoc",
  booktabs = TRUE, row.names = FALSE
)
```

表 G.5: 本书使用的 R 包

Package	Title
agridat	Agricultural Datasets
BANOVA	Hierarchical Bayesian ANOVA Models
BayesX	R Utilities Accompanying the Software Package BayesX
betareg	Beta Regression
blme	Bayesian Linear Mixed-Effects Models
Boom	Bayesian Object Oriented Modeling
BoomSpikeSlab	MCMC for Spike and Slab Regression
brms	Bayesian Regression Models using Stan
CARBayes	Spatial Generalised Linear Mixed Models for Areal Unit Data
cgam	Constrained Generalized Additive Model
cglm	Fits Conditional Generalized Linear Models
cowplot	Streamlined Plot Theme and Plot Annotations for ggplot2
FRK	Fixed Rank Kriging
gam	Generalized Additive Models
gamm4	Generalized Additive Mixed Models using mgcv and lme4
gee	Generalized Estimation Equation Solver
geepack	Generalized Estimating Equation Package
geoR	Analysis of Geostatistical Data
ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics
glmm	Generalized Linear Mixed Models via Monte Carlo Likelihood Approximation
GLMMadaptive	Generalized Linear Mixed Models using Adaptive Gaussian Quadrature

Package	Title
glmmEP	Generalized Linear Mixed Model Analysis via Expectation Propagation
glmmfields	Generalized Linear Mixed Models with Robust Random Fields for Spatiotemporal Modeling
glmmLasso	Variable Selection for Generalized Linear Mixed Models by L1-Penalized Estimation
glmmML	Generalized Linear Models with Clustering
glmmsr	Fit a Generalized Linear Mixed Model
glmmTMB	Generalized Linear Mixed Models using Template Model Builder
glmnet	Lasso and Elastic-Net Regularized Generalized Linear Models
greta	Simple and Scalable Statistical Modelling in R
hglm	Hierarchical Generalized Linear Models
INLABMA	Bayesian Model Averaging with INLA
inlabru	Bayesian Latent Gaussian Modelling using INLA and Extensions
LaplacesDemon	Complete Environment for Bayesian Inference
LatticeKrig	Multi-Resolution Kriging Based on Markov Random Fields
leaflet	Create Interactive Web Maps with the JavaScript Leaflet Library
lgcp	Log-Gaussian Cox Process
lme4	Linear Mixed-Effects Models using Eigen and S4
lmtest	Testing Linear Regression Models
MASS	Support Functions and Datasets for Venables and Ripley's MASS
MCMCglmm	MCMC Generalised Linear Mixed Models
MCMCpack	Markov Chain Monte Carlo (MCMC) Package
mgcv	Mixed GAM Computation Vehicle with Automatic Smoothness Estimation
moments	Moments, cumulants, skewness, kurtosis and related tests
MuMIn	Multi-Model Inference
nimble	MCMC, Particle Filtering, and Programmable Hierarchical Modeling
nlme	Linear and Nonlinear Mixed Effects Models
patchwork	The Composer of Plots
plotly	Create Interactive Web Graphics via plotly.js
PrevMap	Geostatistical Modelling of Spatially Referenced Prevalence Data
pscl	Political Science Computational Laboratory
QGglmm	Estimate Quantitative Genetics Parameters from Generalised Linear Mixed Models
quantreg	Quantile Regression
R2BayesX	Estimate Structured Additive Regression Models with BayesX
r2glmm	Computes R Squared for Mixed (Multilevel) Models
R2OpenBUGS	Running OpenBUGS from R
rgl	3D Visualization Using OpenGL
rjags	Bayesian Graphical Models using MCMC
rpql	Regularized PQL for Joint Selection in GLMMs
rstan	R Interface to Stan
rstanarm	Bayesian Applied Regression Modeling via Stan
spaMM	Mixed-Effect Models, with or without Spatial Random Effects
spBayes	Univariate and Multivariate Spatial-Temporal Modeling
spNNGP	Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes
VGAM	Vector Generalized Linear and Additive Models

© 基础统计

Package	Title
INLA	Full Bayesian Analysis of Latent Gaussian Models using Integrated Nested Laplace Approximatio

© 王建国

附录 H　符号说明

> Fabio Mulazzani: I need to obtain all the 9.somethingExp157 permutations that can be given from the numbers from 1 to 100.
>
> Ted Harding: To an adequate approximation there are 10^{158} of them. Simply to obtain them all (at a rate of 10^{10} per second, which is faster than the CPU frequency of most desktop computers) would take 10^{148} seconds, or slightly longer than 3×10^{140} years. Current estimates of the age of the Universe are of the order of 1.5×10^{10} years, so the Universe will have to last about 2×10^{130} times as long as it has already existed, before the task could be finished.
>
> So: why do you want to do this?
>
> — Fabio Mulazzani and Ted Harding [1]

数学符号约定参考花书 https://github.com/goodfeli/dlbook_notation

Flexible Imputation of Missing Data 的 符号约定章节

> 矩阵、向量用粗体大写表示，特殊情况下，Y 只有一列

$$\mathsf{Y} = \mathsf{X}\beta + \epsilon$$

Y 叫做因变量或者响应变量 response variables，X 叫做自变量、协变量 covariate 或者预报变量 predictor variables

线性回归模型

$$y = X\beta + \epsilon$$

其中 y 是 $n \times 1$ 的观测向量，X 为 $n \times p$ 的设计矩阵，β 为未知参数向量，β_0 为常数项，$\beta_1, \ldots, \beta_{p-1}$ 为回归系数，ϵ 为 $n \times 1$ 随机误差向量，其均值为 0，即 $E(\epsilon_i) = 0$

模型假设

1. 误差项方差齐性，即

$$Var(\epsilon_i) = \sigma^2, \quad i = 1, 2, \ldots, n$$

2. 误差项彼此不相关，即

$$Cov(\epsilon_i, \epsilon_j) = 0, \quad i \neq j, \quad i, j = 1, \ldots, n$$

[1] https://stat.ethz.ch/pipermail/r-help/2008-November/180820.html

© 黄湘云

线性模型中**线性**二字实质上是指 y 关于未知参数 β_i 的关系是线性的。

A, B, C, D 斜体表示普通的集合，X, Y, Z 表示矩阵，a, b, c, d 表示常数，$\alpha, \beta, \theta, \phi, \kappa$ 表示模型或者分布函数的参数，Θ 表示参数空间，$\mathbb{R}^n, \mathbb{C}^n$ 表示特殊的 n 维实（复）数域，$\mathscr{A}, \mathscr{B}, \mathscr{C}, \mathscr{D}$ 表示一般的数域，$\mathcal{S}, \mathcal{P}, \mathcal{G}$ 分别表示随机过程、概率空间和图

表 H.1: 数学符号表

符号	含义	符号	含义
$\mathbf{A}$	粗体	Ω	全集
A	集合	$\mathbb{R}, \mathbb{C}$	实（复）数集
$\mathcal{A}$	集族	$\varnothing$	空集
A	矩阵	A^-	矩阵的广义逆
$\mathsf{A}^\top$	矩阵转置	$\bar{x}$	平均值
A^{-1}	矩阵的逆	$\lvert a\rvert$	标量绝对值
$\mathsf{A}^\star$	伴随矩阵	$\mathrm{diag}(\mathsf{A})$	矩阵的对角
$\lVert\mathsf{A}\rVert_1$	矩阵的 1 范数	I	单位矩阵
$\lVert\mathsf{A}\rVert_2$	矩阵的 2 范数	I_n	n 阶单位矩阵
$\lVert\mathsf{A}\rVert_\infty$	矩阵的无穷范数	$\mathsf{1}$	全 1 矩阵
$\lVert\mathsf{X}\rVert_1$	向量的 1 范数	$\mathsf{1}_n$	n 阶全 1 矩阵
$\lVert\mathsf{X}\rVert_2$	向量的 2 范数	$\lVert\mathsf{X}\rVert_\infty$	向量的无穷范数
$\langle\mathsf{X}, \mathsf{Y}\rangle$	向量的内积	$f(\mathsf{X})$	随机变量的函数
$\mathsf{X} \wedge \mathsf{Y}$	向量的外积	$\nabla\mathsf{X}$	向量微分或梯度
β	模型系数	θ	模型或分布参数
α	模型截距	Θ	参数空间
$\hat{\beta}_{ls}$	模型系数的 LS 估计	$f(x)$	标量值函数
$\hat{\beta}_{mle}$	模型系数的 MLE 估计	$f(\mathsf{X})$	向量的函数
$\hat{\beta}_{bayes}$	模型系数的 Bayes 估计	$\mathcal{X}$	概率空间
ρ	相关系数	κ	贝塞尔函数的阶
ϕ	尺度参数	u	距离 $\lVert\mathsf{x}_1 - \mathsf{x}_2\rVert$
$\mathbb{R}^2$	二维实数域	$S(x)$	空间过程
$\mathcal{S}$	$\mathcal{S} = \{S(x) : x \in \mathbb{R}^2\}$	$\mathcal{S}^\star$	随机过程 $\mathcal{S}$ 的近似
$\triangleq$	定义为或记为	$\hat{\beta}_{ridge}$	β 的岭回归估计
$A \geq 0$	矩阵 A 半正定	$\hat{\beta}_{lar}$	β 的最小角回归估计
$A > 0$	矩阵 A 正定	$\hat{\beta}_{subset}$	β 的最优子集回归估计
$A \otimes B$	矩阵 A 与 B 的 Kronecher 积	MSE 均方误差	$\frac{1}{n}\sum_{i=1}^n (y_i - \hat{y}_i)^2$
$\mathcal{M}(A)$	矩阵 A 的列向量张成的子空间	RMSE 均方根误差	$\sqrt{\frac{1}{n}\sum_{i=1}^n (y_i - \hat{y}_i)^2}$
$\lVert A\rVert$	矩阵 A 的范数	MAE 平均绝对误差	$\frac{1}{n}\sum_{i=1}^n \lvert y_i - \hat{y}_i\rvert$
$\lvert A\rvert$	矩阵 A 的行列式	LSE	最小二乘估计
$rk(A)$	矩阵 A 的秩	BLUE	最佳线性无偏估计
$tr(A)$	方阵 A 的迹	MVUE	最小方差无偏估计
A^{-1}	矩阵 A 的逆	UMVUE	一致最小方差无偏估计
A^-	矩阵 A 的广义逆	MINQUE	最小范数二次无偏估计
$\hat{\beta}_{ols}$	β 的普通最小二乘估计	OLS	普通最小二乘估计

© 黄湘云

符号	含义	符号	含义
$\hat{\beta}_{pca}$	β 的主成分分析估计	PLS	偏最小二乘估计
$\hat{\beta}_{pls}$	β 的偏最小二乘估计	GLS	广义最小二乘估计
$\hat{\beta}_{svm}$	β 的支持向量机估计	WLS	带权最小二乘估计
$\hat{\beta}_{lasso}$	β 的 Lasso 估计	-	-

多元统计分析高惠璇矩阵符号表示，深度学习符号表示 https://github.com/XiangyunHuang/dlbook

举例，线性模型的表示，此处 Y 为 $n \times 1$ 列向量，X 为 $p \times n$ 的矩阵，β 为 $p \times 1$ 的列向量，ϵ 为 $n \times 1$ 列向量

$$Y = X'\beta + \epsilon$$

$$\mathsf{A} = \Gamma^\top \Lambda \Gamma$$

$$\mathsf{u} = (u_1, u_2, \cdots, u_n)$$

$$\mathsf{x} = (x_1, x_2, \cdots, x_n)$$

期望 E 正态分布 $\mathcal{N}(\mathsf{x}; \mu, \Sigma)$ 对数 log 协方差 $\mathsf{Cov}, \mathsf{Var}$

矩阵

$$\mathsf{Y} = (\mathsf{y}^{(1)}, \mathsf{y}^{(n)}, \cdots, \mathsf{y}^{(n)})$$

其中 $\mathsf{y}^{(i)} = (y_{1i}, y_{2i}, \cdots, y_{ni})$ 表示第 i 列

梅隆函数 (Matern function) 是描述空间相关性的常用函数，它带有两参数 κ 和 ϕ，具体形式如下：

$$\rho(u) = \left\{2^{\kappa-1}\Gamma(\kappa)\right\}^{-1}(u/\phi)^{\kappa}K_{\kappa}(u/\phi)$$

其中，$K_\kappa(\cdot)$ 表示 κ 阶修正的贝塞尔函数

© 苏德矿

索引

bookdown, 7

Octave, 4

Pandoc, 7
Python, 4

信仰区间, 1

统计分布, 1
统计功效, 1
置信区间, 1

© 草稿

参考文献

JJ Allaire, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, Winston Chang, and Richard Iannone. *rmarkdown: Dynamic Documents for R*, 2021. URL https://CRAN.R-project.org/package=rmarkdown. R package version 2.9.

Mikkel Meyer Andersen and Søren Højsgaard. Ryacas: A computer algebra system in R. *Journal of Open Source Software*, 4(42), 2019. URL https://doi.org/10.21105/joss.01763.

Douglas M. Bates and Donald G. Watts. *Nonlinear Regression Analysis and Its Applications*. John Wiley & Sons, New York, NY, 1988. URL https://doi.org/10.1002/9780470316757.app2.

Geoffrey Beall. The transformation of data from entomological field experiments so that the analysis of variance becomes applicable. *Biometrika*, 32(3/4):243–262, 1942. doi: 10.2307/2332128. URL https://www.jstor.org/stable/2332128.

Paul Berger and Robert Maurer. *Experimental Design*. Duxbury, 1st edition, 2002. ISBN 0-534-35822-5.

Paul Berger, Robert Maurer, and Giovana B. Celli. *Experimental Design*. Springer International Publishing, New York, NY, 2nd edition, 2018. doi: 10.1007/978-3-319-64583-4. ISBN 978-3-319-64582-7.

Mickaël Binois and Victor Picheny. GPareto: An R package for gaussian-process-based multi-objective optimization and analysis. *Journal of Statistical Software*, 89(8):1–30, 2019. doi: 10.18637/jss.v089.i08.

Hans W. Borchers. *pracma: Practical Numerical Math Functions*, 2021. URL https://CRAN.R-project.org/package=pracma. R package version 2.3.3.

George E. P. Box, J. Stuart Hunter, and William G. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. John Wiley & Sons, Inc, Hoboken, New Jersey, 2nd edition, 2005. ISBN 978-0471-71813-0.

Stephen Boyd and Lieven Vandenberghe. *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. Cambridge University Press, New York, NY, 2018. URL https://web.stanford.edu/~boyd/vmls/vmls.pdf.

Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Journal of the American Statistical Association*, 16(3):199–231, 12 2001. doi: 10.1214/ss/1009213726.

David R. Brillinger. *Time Series: Data Analysis and Theory*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001. ISBN 0-89871-501-6.

John M. Chambers. S, R, and Data Science. *The R Journal*, 12(1):462–476, 2020. doi: 10.32614/RJ-2020-028. URL https://doi.org/10.32614/RJ-2020-028.

Winston Chang, Alexej Kryukov, and Paul Murrell. *fontcm: Computer Modern font for use with extrafont package*, 2014. URL https://github.com/wch/fontcm. R package version 1.1.

© 草稿

C. J. Clopper and E. S. Pearson. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 12 1934. doi: 10.1093/biomet/26.4.404.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd edition, 1988. URL https://www.utstat.toronto.edu/~brunner/oldclass/378f16/readings/CohenPower.pdf. ISBN 0-8058-0283-5.

Jacob Cohen. The earth is round ($p < .05$). *American Psychologist*, 49(12):997–1003, 1994. doi: 10.1037/0003-066x.49.12.997.

Alex Couture-Beil, Jon T. Schnute, Rowan Haigh, Simon N. Wood, and Benjamin J. Cairns. *PBSddesolve: Solver for Delay Differential Equations*, 2019. URL https://CRAN.R-project.org/package=PBSddesolve. R package version 1.12.6.

Kalyanmoy Deb. *Multi-Objective Optimization*, pages 273–316. Springer US, Boston, MA, 2005. ISBN 978-0-387-28356-2. doi: 10.1007/0-387-28356-0_10. URL https://doi.org/10.1007/0-387-28356-0_10.

Peter B Denton, Stephen J Parke, Terence Tao, and Xining Zhang. Eigenvectors from eigenvalues. 2019. URL https://arxiv.org/pdf/1908.03795.pdf.

Annette J. Dobson. *An Introduction to Statistical Modelling*. Chapman and Hall/CRC, London, 1st edition, 1983. doi: 10.1007/978-1-4899-3174-0. ISBN 978-0412248603.

Annette J. Dobson and Adrian G. Barnett. *An Introduction to Generalized Linear Models*. Chapman and Hall/CRC, Boca Raton, Florida, fourth edition, 2018. URL https://www.crcpress.com/p/book/9781138741515. ISBN 978-1138741515.

Morris L. Eaton. *Chapter 8: The Wishart Distribution*, volume 53 of *Lecture Notes – Monograph Series*, pages 302–333. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2007. doi: 10.1214/lnms/1196285114. URL https://doi.org/10.1214/lnms/1196285114.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004. doi: 10.1214/009053604000000067.

T. W. Epps and Lawrence B. Pulley. A test for normality based on the empirical characteristic function. *Biometrika*, 70(3):723–726, 1983. doi: 10.2307/2336512.

Mark A. Espeland and Siu L. Hui. A general approach to analyzing epidemiologic data that contain misclassification errors. *Biometrics*, 43(4):1001–1012, 1987. URL https://www.jstor.org/stable/2531553.

John Fox and Sanford Weisberg. *An R Companion to Applied Regression*. Sage, Thousand Oaks CA, third edition, 2019. URL https://socialsciences.mcmaster.ca/jfox/Books/Companion/.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001. URL https://projecteuclid.org/euclid.aos/1013203451.

Mike Fritz and Paul D. Berger. *Improving the User Experience through Practical Data Analytics: Gain Meaningful Insight and Increase Your Bottom Line*. Morgan Kaufmann, 1st edition, 2015. ISBN 978-0128006351.

Anqi Fu, Balasubramanian Narasimhan, and Stephen Boyd. CVXR: An R package for disciplined convex optimization. *Journal of Statistical Software*, 94(14):1–34, 2020. doi: 10.18637/jss.v094.i14.

Manfred Gilli, Dietmar Maringer, and Enrico Schumann. *Numerical Methods and Optimization in Finance*. Elsevier/Academic Press, Waltham, MA, USA, second edition, 2019. URL http://www.enricoschumann.net/NMOF/. ISBN 978-0128150658.

© 博硕士

Gabor Grothendieck and Thomas Petzoldt. R Help Desk: Date and time classes in R. *R News*, 4(1):29–32, June 2004. URL https://www.r-project.org/doc/Rnews/Rnews_2004-1.pdf.

Zuguang Gu, Roland Eils, and Matthias Schlesner. Complex heatmaps reveal patterns and correlations in multidimensional genomic data. *Bioinformatics*, 2016.

Asad Hasan, Zhiyu Wang, and Alireza S. Mahani. Fast estimation of multinomial logit models: R package mnlogit. *Journal of Statistical Software*, 75(3):1–24, 2016. doi: 10.18637/jss.v075.i03.

C. C. Heyde, E. Seneta, P. Crépel, S. E. Fienberg, and J. Gani. *Statisticians of the Centuries*. Springer-Verlag, New York, NY, 2001. doi: 10.1007/978-1-4613-0179-0. ISBN 978-1-4613-0179-0.

David C. Hoaglin and Roy E. Welsch. The hat matrix in regression and anova. *The American Statistician*, 32(1):17–22, 1978. URL https://www.jstor.org/stable/2683469.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10, 2004. doi: 10.1016/j.ijforecast.2003.09.015.

Kurt Hornik. R FAQ: Frequently asked questions on R, 2020. URL https://CRAN.R-project.org/doc/FAQ/R-FAQ.html.

David W. Hosmer and Stanley Lemeshow. *Applied Logistic Regression*. John Wiley & Sons, New York, NY, second edition, 2000. ISBN 0-471-72214-6.

Torsten Hothorn, Kurt Hornik, Mark A. van de Wiel, and Achim Zeileis. Implementing a class of permutation tests: The coin package. *Journal of Statistical Software*, 28(8):1–23, 2008. doi: 10.18637/jss.v028.i08.

P. L. HSU. Contribution to the theory of "student's" t-test as applied to the problem of two samples. *Statistical Research Memoirs*, 2:1–24, 1938.

P. L. HSU. *Collected Papers*. Springer-Verlag, New York, NY, 1983. ISBN 978-1-49-392241-3.

Ross Ihaka and Robert Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

H. Ishwaran and J.S. Rao. Spike and slab variable selection: frequentist and bayesian strategies. *Ann. Statist.*, 33(2):730–773, 2005. URL http://arXiv.org/abs/math/0505633v1.

Norman L. Johnson and Samuel Kotz. *Leading Personalities in Statistical Sciences: From the Seventeenth Century to the Present*. John Wiley & Sons, New York, NY, 1997. ISBN 0-471-16381-3.

Robert I. Kabacoff. *R in Action: Data Analysis and Graphics with R*. Manning PUblications Co., Shelter Island, NY, 2nd edition, 2015. URL https://github.com/kabacoff/RiA2. ISBN 978-1617291388.

Peter Kampstra. beanplot: A boxplot alternative for visual comparison of distributions. *Journal of Statistical Software*, 28(1):1–9, 2008. URL http://www.jstatsoft.org/v28/c01/.

Peter Kasprzak, Lachlan Mitchell, Olena Kravchuk, and Andy Timmins. Six Years of Shiny in Research - Collaborative Development of Web Tools in R. *The R Journal*, 12(2):155–162, 2021. doi: 10.32614/RJ-2021-004. URL https://doi.org/10.32614/RJ-2021-004.

Seock-Ho Kim and Allan S. Cohen. On the behrens-fisher problem: A review. *Journal of Educational and Behavioral Statistics*, 23(4):356–377, 1998. doi: 10.2307/1165281. URL https://www.jstor.org/stable/1165281.

© 清华大学出版社

Yongdai Kim, Hosik Choi, and Hee-Seok Oh. Smoothly clipped absolute deviation on high dimensions. *Journal of the American Statistical Association*, 103(484):1665–1673, 2008. doi: 10.1198/016214508000001066.

Christian Kleiber and Achim Zeileis. *Applied Econometrics with R*. Springer-Verlag, New York, 2008. URL https://CRAN.R-project.org/package=AER. ISBN 978-0-387-77316-2.

Ron Kohavi, Diane Tang, and Ya Xu. *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing*. Cambridge University Press, Cambridge, United Kingdom, 2020. URL https://experimentguide.com/. ISBN 9781108724265.

Alexandra Kuznetsova, Per B. Brockhoff, and Rune H. B. Christensen. lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, 82(13):1–26, 2017. doi: 10.18637/jss.v082.i13.

Pierre Lafaye de Micheaux and Viet Anh Tran. PoweR: A reproducible research tool to ease monte carlo power simulation studies for goodness-of-fit tests in R. *Journal of Statistical Software*, 69(3):1–42, 2016. doi: 10.18637/jss.v069.i03.

John Lawson. *Design and Analysis of Experiments with R*. Chapman and Hall/CRC, Boca Raton, Florida, 1st edition, 2014. URL http://www.mvstat.net/mvksa/mvksa.pdf. ISBN 978-1498728485.

Lawrence M. Leemis. Relationships among common univariate distributions. *The American Statistician*, 40(2):143–146, 1986. URL https://www.jstor.org/stable/2684876.

Daniel Lüdecke, Philip Waggoner, and Dominique Makowski. insight: A unified interface to access information from model objects in r. *Journal of Open Source Software*, 4(38):1412, 2019. doi: 10.21105/joss.01412.

Dominique Makowski, Mattan Ben-Shachar, and Daniel Lüdecke. bayestestR: Describing effects and their uncertainty, existence and significance within the bayesian framework. *Journal of Open Source Software*, 4(40):1541, 2019. doi: 10.21105/joss.01541.

Ricardo A. Maronna, R. Douglas Martin, and Victor J. Yohai. *Robust Statistics, Theory and Methods*. Wiley Series in Probility and Statistics. John Wiley & Sons, Ltd, 2006. ISBN 0-470-01092-4.

Peter McCullagh and John Nelder. *Generalized Linear Models*. Chapman and Hall/CRC, London, second edition, 1989. URL https://www.crcpress.com/p/book/9780412317606.

A.I. McLeod. *Kendall: Kendall rank correlation and Mann-Kendall trend test*, 2011. URL http://www.stats.uwo.ca/faculty/aim. R package version 2.2.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, Amit Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. SymPy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL https://doi.org/10.7717/peerj-cs.103.

Björn-Helge Mevik and Ron Wehrens. The pls package: Principal component and partial least squares regression in r. *Journal of Statistical Software*, 18(2):1–23, 2007. doi: 10.18637/jss.v018.i02. URL https://www.jstatsoft.org/v018/i02.

Paul Murrell. Integrating grid graphics output with base graphics output. *R News*, 3(2):7–12, 2003.

Paul Murrell and Ross Ihaka. An approach to providing mathematical annotation in plots. *Journal of Computational and Graphical Statistics*, 9(3):582–599, 2000.

John C. Nash. On best practice optimization methods in r. *Journal of Statistical Software*, 60(2):1–14, 2014. doi: 10.18637/jss.v060.i02. URL https://www.jstatsoft.org/v060/i02.

Erich Neuwirth. *RColorBrewer: ColorBrewer Palettes*, 2014. URL https://CRAN.R-project.org/package=RColorBrewer. R package version 1.1-2.

Robert G. Newcombe. Interval estimation for the difference between independent proportions: comparison of eleven methods. *Statistics in Medicine*, 17(8):873–890, 1998. doi: 10.1002/(SICI)1097-0258(19980430)17:8<873::AID-SIM779>3.0.CO;2-I.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Xiaoying Pu and Matthew Kay. A probabilistic grammar of graphics. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376466. URL https://doi.org/10.1145/3313831.3376466.

Yixuan Qiu. showtext: Using system fonts in R graphics. *The R Journal*, 7(1):99–108, jun 2015. doi: 10.32614/RJ-2015-008.

Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning*. Packt Publishing, Birmingham, UK, 2nd edition, 2017. ISBN 978-1787125933.

Brian D. Ripley. Statistical methods need software: A view of statistical computing, 9 2002. URL https://www.stats.ox.ac.uk/~ripley/RSS2002.pdf.

Brian D. Ripley and Kurt Hornik. Date-time classes. *R News*, 1(2):8–11, June 2001. URL https://cran.r-project.org/doc/Rnews/Rnews_2001-2.pdf.

Petr Savicky. *pspearman: Spearman's rank correlation test*, 2014. URL https://CRAN.R-project.org/package=pspearman. R package version 0.3-0.

Luca Scrucca. GA: A package for genetic algorithms in R. *Journal of Statistical Software*, 53(4):1–37, 2013. URL https://www.jstatsoft.org/v53/i04/.

Luca Scrucca. On some extensions to GA package: hybrid optimisation, parallelisation and islands evolution. *The R Journal*, 9(1):187–206, 2017. URL https://journal.r-project.org/archive/2017/RJ-2017-008/.

Karline Soetaert and Filip Meysman. Reactive transport in aquatic ecosystems: Rapid model prototyping in the open source software R. *Environmental Modelling & Software*, 32:49–60, 2012.

Reto Stauffer, Georg J. Mayr, Markus Dabernig, and Achim Zeileis. Somewhere over the rainbow: How to make effective use of colors in meteorological visualizations. *Bulletin of the American Meteorological Society*, 96(2):203–216, 2009. doi: 10.1175/BAMS-D-13-00155.1.

"Student". The probable error of a mean. *Biometrika*, 6:1–25, 1908.

© 清华大学出版社

Yuan Tang. autoplotly: An r package for automatic generation of interactive visualizations for statistical results. *Journal of Open Source Software*, 3, 2018. URL https://doi.org/10.21105/joss.00657.

Yuan Tang, Masaaki Horikoshi, and Wenxuan Li. ggfortify: Unified interface to visualize statistical results of popular r packages. *The R Journal*, 8(2):474–485, 2016. doi: 10.32614/RJ-2016-060. URL https://journal.r-project.org/archive/2016/RJ-2016-060/RJ-2016-060.pdf.

Terry M. Therneau and Patricia M. Grambsch. *Modeling Survival Data: Extending the Cox Model*. Springer, New York, 2000. ISBN 0-387-98784-3.

Peter F. Thall and Stephen C. Vail. Some covariance models for longitudinal count data with overdispersion. *Biometrics*, 46(3):657–671, 1990. URL https://www.jstor.org/stable/2532086.

Stefan Theußl, Florian Schwendinger, and Kurt Hornik. ROI: An extensible R optimization infrastructure. *Journal of Statistical Software*, 94(15):1–64, 2020. doi: 10.18637/jss.v094.i15.

Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. URL http://www.jstor.org/stable/2346178.

Emilio Torres-Manzanera. *xkcd: Plotting ggplot2 Graphics in an XKCD Style*, 2018. R package version 0.0.6.

Kenneth E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, New York, NY, second edition, 2009. ISBN 9780511805271.

Michail Tsagris and Manos Papadakis. Taking r to its limits: 70+ tips. *PeerJ Preprints*, 6:e26605v1, 2018. ISSN 2167–9843. doi: 10.7287/peerj.preprints.26605v1. URL https://doi.org/10.7287/peerj.preprints.26605v1.

Berwin A. Turlach. *quadprog: Functions to Solve Quadratic Programming Problems.*, 2019. URL https://CRAN.R-project.org/package=quadprog. R package version 1.5-8.

Kevin Ushey, JJ Allaire, and Yuan Tang. *reticulate: Interface to Python*, 2021. URL https://github.com/rstudio/reticulate. R package version 1.20.

M.P.J. van der Loo. The stringdist package for approximate string matching. *The R Journal*, 6:111–122, 2014. URL https://CRAN.R-project.org/package=stringdist.

Ravi Varadhan and Paul Gilbert. BB: An R package for solving a large system of nonlinear equations and for optimizing a high-dimensional nonlinear objective function. *Journal of Statistical Software*, 32(4): 1–26, 2009. URL https://www.jstatsoft.org/v32/i04/.

W. N. Venables and Brian D. Ripley. *Modern Applied Statistics with S*. Springer-Verlag, New York, NY, fourth edition, 2002. URL http://www.stats.ox.ac.uk/pub/MASS4. ISBN 0-387-95457-0.

Bob Wheeler. *SuppDists: Supplementary Distributions*, 2020. URL https://CRAN.R-project.org/package=SuppDists. R package version 1.1-9.5.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York, 2nd edition, 2016. URL https://ggplot2-book.org/. ISBN 978-3319242774.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 6 1927. doi: 10.1080/01621459.1927.10502953.

Peter R. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6 (3):324–342, 1960. doi: 10.1287/mnsc.6.3.324.

© 清华大学出版社

Yihui Xie. animation: An R package for creating animations and demonstrating statistical methods. *Journal of Statistical Software*, 53(1):1–27, 2013. URL http://www.jstatsoft.org/v53/i01/.

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL https://yihui.org/knitr/. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC, Boca Raton, Florida, 2016. URL https://github.com/rstudio/bookdown. ISBN 978-1138700109.

Yihui Xie. TinyTeX: A lightweight, cross-platform, and easy-to-maintain latex distribution based on TeX Live. *TUGboat*, (1):30–32, 2019. URL https://tug.org/TUGboat/Contents/contents40-1.html.

Yihui Xie, J.J. Allaire, and Garrett Grolemund. *R Markdown: The Definitive Guide*. Chapman and Hall/CRC, Boca Raton, Florida, 2018. URL https://bookdown.org/yihui/rmarkdown. ISBN 9781138359338.

Jelmer Ypma. *R Interface to NLopt*, 2020. URL https://github.com/jyypma/nloptr. R package version 1.2.2.2.

Achim Zeileis and Torsten Hothorn. Diagnostic checking in regression relationships. *R News*, 2(3):7–10, 2002. URL https://CRAN.R-project.org/doc/Rnews/.

Achim Zeileis, Kurt Hornik, and Paul Murrell. Escaping RGBland: Selecting colors for statistical graphics. *Computational Statistics & Data Analysis*, 53(9):3259–3270, 2009. doi: 10.1016/j.csda.2008.11.033.

Achim Zeileis, Jason C. Fisher, Kurt Hornik, Ross Ihaka, Claire D. McWhite, Paul Murrell, Reto Stauffer, and Claus O. Wilke. colorspace: A toolbox for manipulating and assessing colors and palettes. arXiv 1903.06490, arXiv.org E-Print Archive, March 2019. URL http://arxiv.org/abs/1903.06490.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894 – 942, 2010. doi: 10.1214/09-AOS729.

宋泽熙. 两个二项总体成功概率的比较. 中国校外教育（理论）, z1:81, 2011. doi: 10.3969/j.issn.1004-8502-B.2011.z1.0919.

茆诗松, 周纪芗, and 陈颖. 试验设计. 中国统计出版社, 北京, 1st edition, 2004. ISBN 7-5037-4316-6.

茆诗松, 程依明, and 濮晓龙. 高等数理统计. 高等教育出版社, 北京, 2nd edition, 2006. ISBN 978-7-04-019321-3.

陈希孺. 数理统计引论. 科学出版社, 北京, 1981.

韦博成. 《红楼梦》前 80 回与后 40 回某些文风差异的统计分析（两个独立二项总体等价性检验的一个应用）. 应用概率统计, 25(4):441–448, 2009. doi: 10.3969/j.issn.1001-4268.2009.04.012.