

# 《R 语言实战》识记、练习、注释手册

（2018 年 2 月 19 日版）

by 李浩宇

## 前言

本手册的内容：

本手册来源于《R 语言实战》和我本人的 R 学习笔记，是对 R 语言经典教材《R 语言实战》的重新编排和梳理，而且就一些需要进一步解释或者值得补充的地方添加了脚注（本手册对原书添加了一百余处注释，全部放在了手册的脚注中；如在第 8 章的注释 100 中补充了如何将回归结果工整地导入 Excel）。在一些地方，本手册采用更便于理解和更通俗的语句阐释了命令的含义和功能（如 7.2.1 节对三维列联表边际频数的阐述）。

本手册参照《R 语言实战》第一版进行编排（包含了第一版的第 1 章至第 15 章），还纳入了《R 语言实战》第二版新添加的“第 15 章——时间序列分析”。另外，有一些案例代码使用了第二版的内容（如 4.3.1 节中选入观测的代码以及 7.1.2 节中有关 `by()` 分组计算描述性统计量的代码）。凡是没有特别说明的，都是参照第一版内容所写，凡是在第 1 章至第 15 章使用了第二版内容的地方都会加以注明。

本手册指出了若干第一版存在翻译错误且第二版仍未更正的地方（如 9.3.1 节的倒数第二段），若干原文作者的代码错误或遗漏（如 9.3.2 节绘制图 9-4 的代码），若干原文作者理解错误的地方（如对代码清单 9-3 中 `table(dose)` 的解释），以及若干作者未加以阐述而又难以理解的地方（如对 5.2.3 节中 `yaxs = "i"` 参数的解释以及为什么在设定了 `option(digits=3)` 参数的情况下数字仍呈现了不同的有效位数）。

本手册力争做到对代码的严谨解读。对疑似有问题的地方，会同时参考中文第一、二版，英文原版第一、二版，再结合其他参考资料进行阐述。

本手册的目的：

《R 语言实战》是很多同学学习 R 语言的首选材料。按该书中文第一版封底的一句话来说：“通读本书，你将全面掌握使用 R 语言进行数据分析，数据挖掘的技巧，并领略大量探索和展示数据的图形功能。”然而，仅“通读”显然是不够的。

其一，相信大家在初学 R 的时候遇到的一大困难就是难以在短时间内理解、识记庞杂的命令代码。经常是看到后一页时已经忘了前一页的代码，看到第二章的时候已经忘了第一章的内容，一段时间不复习连 R 是怎么工作的都忘了。实际上如果学习和使用过程是同步的，以上问题能够很好的解决。但受到很多现实原因的限制，往往学习过程和使用过程是分开的。很多同学在使用 R 之前会抽出一整段时间学习《R 语言实战》，以应对未来的使用，但往往等到使用的时候，已经遗忘了学习到的命令。于是又只能重新学习，陷入不断重新学习的怪圈。本识记，练习手册的目的在于让 R 学习者能不用上机实操就够方便快捷地识记、练习 R 命令（当然实操练习是无法被取代的）。

其二，对于新手而言，《R 语言实战》对有些命令的解释不够充分，也不便于理解。有一种情况是作者在后文中采用了前面讲过的命令，进而不再赘述。但是对于新手来说看到后面的时候早就忘了前面的内容。本手册使用了大量更加具体，更加生动的语句对命令进行了重新描述，基本做到了对原文的所有命令进行逐行逐个描述。如果通过

阅读原文无法理解某个命令的含义，读者可以在本手册对应处中找到更“傻瓜”的描述。我想在这一方面，这本手册应该不会让大家失望，因为我在学习 R 时就是一个“傻瓜”。

本手册的特点：

1. 在排版上采用了右描述，左命令的方式。对于有“识”要求的读者，可以遮住右侧（解释部分），思考左侧命令的含义。对于有“记”要求的读者，可以遮住左侧（命令部分），通过右侧的释义写出相应的命令。
2. 右侧的描述以祈使句、疑问句，或需要填空的句子撰写，相当于每一个描述都是一个练习题，由此达到练习的目的。
3. 本手册本着零基础用户也能看懂的原则，对一些值得补充或者需要进一步明确的内容进行了注释，也在原书的基础上增加了一些批注和辅助理解、记忆的内容（这使得本手册的一些注释显得比较啰嗦）。
4. 为了达到练习的目的，或是为原书中仅以文字描述的内容增添举例，本手册改写了一些书中的命令，并丰富了一些命令的实例。

本手册的使用方法：

1. 复习时遮住右侧和左侧，分别练习“识”和“记”。
2. 个别内容只在右侧出现了描述，而左边代码部分为空白，此时只需要阅读右侧内容即可。
3. 对于初学者来说越是基础的章节（第 1 章至第 7 章）越需要熟练掌握其代码，而具体到某种统计或计量方法的章节反而没有那么重要。

其他说明：

“\_\_\_\_\_”表示此处需要填空。

“【补】”表示此处是此手册补充的内容。

本手册于 2018 年 1 月 18 日首发于人大经济论坛

<http://bbs.pinggu.org/thread-6187301-1-1.html>

李浩宇

2018/01/18

于 世界尽头与冷酷仙境

## 目录

第 1 章 R 语言介绍 .....	1
1.1 为何要使用 R? .....	1
1.2 R 的获取和安装 .....	1
1.3 R 的使用 .....	1
1.4 包 .....	2
1.5 批处理 .....	3
1.6 将输出用为输入——结果的重用 .....	3
1.7 处理大数据 .....	3
1.8 示例实践 .....	3
第 2 章 创建数据集 .....	4
2.1 数据集的概念 .....	4
2.2 数据结构 .....	4
2.3 数据的输入 .....	7
2.4 数据集的标注 .....	7
2.5 处理数据对象的实用函数 .....	8
第 3 章 图形初阶 .....	9
3.1 使用图形 .....	9
3.2 一个简单的例子 .....	9
3.3 图形参数 .....	9
3.4 添加文本、自定义坐标轴和图例 .....	12
3.5 图形的组合 .....	16
第 4 章 基本数据管理 .....	20
4.1 一个示例 .....	20
4.2 创建新变量 .....	20
4.3 变量的重编码 .....	21
4.4 变量的重命名 .....	22
4.5 缺失值 .....	22
4.6 日期值 .....	23
4.7 类型转换 .....	24
4.8 数据排序 .....	25
4.9 数据集的合并 .....	25
4.10 数据集取子集 .....	26
4.11 使用 SQL 语句操作数据框 .....	28
第 5 章 高级数据管理 .....	29
5.1 一个数据处理难题 .....	29
5.2 数值和字符处理函数 .....	29
5.3 数据处理难题的一套解决方案 .....	38
5.4 控制流 .....	40

5.5 自编函数 .....	41
5.6 整合与重构 .....	43
第 6 章 基本图形 .....	47
6.1 条形图 .....	47
6.2 饼图 .....	48
6.3 直方图 .....	49
6.4 核密度图 .....	51
6.5 箱线图 .....	52
6.6 点图 .....	53
第 7 章 基本统计分析 .....	54
7.1 描述性统计分析 .....	54
7.2 频数表和列联表 .....	58
7.3 相关 .....	62
7.4 t 检验 .....	64
7.5 组间差异的非参数检验 .....	66
7.6 组间差异的可视化 .....	67
第 8 章 回归 .....	69
8.1 回归的多面性 .....	69
8.2 OLS 回归 .....	70
8.3 回归诊断 .....	73
8.4 异常观测值 .....	77
8.5 改进措施 .....	78
8.6 选择“最佳”的回归模型 .....	80
8.7 深层次分析 .....	82
第 9 章 方差分析 .....	84
9.1 术语速成 .....	84
9.2 ANOVA 模型拟合 .....	85
9.3 单因素方差分析 .....	87
9.4 单因素协方差分析 .....	90
9.5 双因素方差分析 .....	92
9.6 重复测量方差分析 .....	93
9.7 多元方差分析 .....	93
9.8 用回归来做 ANOVA .....	95
第 10 章 功效分析 .....	96
10.1 假设检验速览 .....	96
10.2 用 pwr 包做功效分析 .....	96
10.3 绘制功效分析图形 .....	100
10.4 其他软件包 .....	101
第 11 章 中级绘图 .....	102

11.1 散点图 .....	102
11.2 折线图 .....	105
11.3 相关图 .....	107
11.4 马赛克图 .....	108
第 12 章 重抽样与自助法 .....	109
12.1 置换检验 .....	109
12.2 用 coin 包做置换检验 .....	109
12.3 lmpPerm 包的置换检验 .....	112
12.4 置换检验点评 .....	114
12.5 自助法 .....	114
12.6 boot 包中的自助法 .....	114
第 13 章 广义线性模型 .....	117
13.1 广义线性模型和 glm() 函数 .....	117
13.2 Logistic 回归 .....	119
13.3 泊松回归 .....	122
第 14 章 主成分分析和因子分析 .....	125
14.1 R 中的主成分和因子分析 .....	125
14.2 主成分分析 .....	125
14.3 探索性因子分析 .....	128
14.4 其他潜变量模型 .....	130
第 15 章 处理缺失数据的高级方法 .....	131
15.1 处理缺失值的步骤 .....	131
15.2 识别缺失值 .....	131
15.3 探索缺失值模式 .....	132
15.4 理解缺失数据的来由和影响 .....	133
15.5 理性处理不完整数据 .....	133
15.6 完整实例分析（行删除） .....	133
15.7 多重插补 .....	133
15.8 处理缺失值的其他方法 .....	134
第 16 章 时间序列 .....	135
16.1 在 R 中生成时序对象 .....	136
16.2 时序的平滑化和季节性分解 .....	137
16.3 指数预测模型 .....	139
16.4 ARIMA 预测模型 .....	141
16.5 延伸阅读 .....	145



# 第 1 章 R 语言介绍

## 1.1 为何要使用 R?

## 1.2 R 的获取和安装

## 1.3 R 的使用

<code>&lt;-</code>	R 使用_____而不是传统的=作为赋值符号
<code>#</code>	注释由符号_____开头

### 1.3.1 新手上路

<code>q ( )</code>	函数_____将结束会话并退出 R
<code>age &lt;- c(1, 2, 3)</code>	创建一个由 <code>age</code> 表示的向量，元素为 1, 2, 3
<code>weight &lt;- c(4, 5, 6)</code>	创建一个由 <code>weight</code> 表示的向量，元素为 4, 5, 6
<code>mean (weight)</code>	求 <code>weight</code> 的平均值
<code>sd (weight)</code>	求 <code>weight</code> 的标准差
<code>cor (age, weight)</code>	求 <code>age</code> 和 <code>weight</code> 间的相关系数
<code>plot (age, weight)</code>	绘制横轴变量为 <code>age</code> ，纵轴变量为 <code>weight</code> 的散点图

### 1.3.2 获取帮助

<code>help.start ( )</code>	打开帮助文档首页
<code>help (foo)</code> 或 <code>help ("foo")</code> 或 <code>?foo</code>	查看函数 <code>foo</code> 的帮助
<code>help.search ("foo")</code> 或 <code>??foo</code>	以 <code>foo</code> 为关键词搜索本地帮助文档
<code>example (foo)</code> 或 <code>example ("foo")</code>	函数 <code>foo</code> 的使用示例
<code>RSiteSearch ("foo")</code>	以 <code>foo</code> 为关键词搜索在线文档和邮件列表存档
<code>apropos ("foo", mode="function")</code>	列出名称中含有 <code>foo</code> 的所有可用函数
<code>data ( )</code>	列出当前已加载包中所含的所有可用示例数据集
<code>vignette ( )</code>	列出当前已安装包中所有可用的 <code>vignette</code> 文档
<code>vignette ("foo")</code>	为主题 <code>foo</code> 显示指定的 <code>vignette</code> 文档

### 1.3.3 工作空间

<code>getwd ( )</code>	显示当前的工作目录
<code>setwd ("C:/myprojects/project1")</code>	修改当前的工作目录为 C:/myprojects/project1 <sup>1</sup>
<code>ls ( )</code>	列出当前工作空间中的对象

<sup>1</sup> 可以在 RStudio 中通过 Session – Set Working Directory – Choose Working Directory 实现。



<code>rm (objectlist)</code>	移除（删除）一个或多个对象
<code>help (options)</code>	显示可用选项的说明
<code>options ( )</code>	显示或设置当前选项
<code>history (#)</code>	显示最近使用过的#个命令（默认值为 25） <sup>2</sup>
<code>savehistory ("myfile")</code>	保存命令历史到文件 <code>myfile</code> 中（默认值为 <code>.Rhistory</code> ）
<code>loadhistory ("myfile")</code>	载入一个命令历史文件（默认值为 <code>.Rhistory</code> ）
<code>save.image ("myfile")</code>	保存工作空间到文件 <code>myfile</code> 中（默认值为 <code>.RData</code> ） <sup>3</sup>
<code>save (objectlist, file="myfile")</code>	保存指定对象到一个文件中
<code>load ("myfile")</code>	读取一个工作空间到当前会话中（默认值为 <code>.RData</code> ） <sup>4</sup>

<code>options (digits=3)</code>	设定默认小数点后三位有效数字
<code>x &lt;- runif (20)</code>	创建包含 20 个均匀分布随机变量的向量，命名为 <code>x</code>
<code>summary (x)</code>	生成 <code>x</code> 的统计摘要
<code>hist (x)</code>	生成 <code>x</code> 的直方图

### 1.3.4 输入和输出<sup>5</sup>

## 1.4 包

### 1.4.1 什么是包

<code>library ( )</code>	函数_____可以显示库中有那些包
<code>search ( )</code>	命令_____ 可以显示哪些包已加载并可以使用 <sup>6</sup>

### 1.4.2 包的安装

<code>install.packages ("gclus")</code>	安装 <code>gclus</code> 包
<code>update.packages ("gclus")</code>	更新已安装的 <code>gclus</code> 包 <sup>7</sup>
<code>installed.packages ( )</code>	查看已安装包的描述

### 1.4.2 包的载入

<code>library (gclus)</code>	载入 <code>gclus</code> 包
------------------------------	-------------------------

### 1.4.3 包的使用方法

<sup>2</sup> 在 RStudio 的 History 面板中能查看过往命令。

<sup>3</sup> 可以在 RStudio 中通过 Session—Save Workspace AS 实现。

<sup>4</sup> 可以在 RStudio 中通过 Session – Load Workspace 实现。

<sup>5</sup> 在 RStudio 中生成图像后，通过 Plots 面板的 Export 选项可以将图像保存为包括 PDF、PNG、JPEG、BMP 在内的多种格式的文件。

<sup>6</sup> 在 RStudio 的 Packages 面板中能够查看、查询、更新已经安装并且可使用的包。

<sup>7</sup> 在 RStudio 的 Packages 面板中能够查看、查询、更新已经安装并且可使用的包。

<code>help (package="gclus")</code>	输出 <code>gclus</code> 包简短描述以及包中的函数名称和数据集名称的列表
-------------------------------------	---

## 1.5 批处理

## 1.6 将输出用为输入——结果的重用

<code>lmfit &lt;- lm (mpg~wt, data=mtcars)</code>	针对 <code>mtcars</code> 数据集，做 <code>mpg</code> 对 <code>wt</code> 的简单线性回归 <sup>8</sup> ，结果保存在 <code>lmfit</code> 中
<code>summary (lmfit)</code>	显示分析上述回归结果的统计概要
<code>plot (lmfit)</code>	生成上述回归的诊断图形

## 1.7 处理大数据

## 1.8 示例实践

---

<sup>8</sup> 本手册采用伍德里奇在《计量经济学导论》中的方式描述回归，“做 `mpg` 对 `wt` 的简单线性回归”意味 `mpg` 为因变量，`wt` 为自变量。下文（由其在第 8 章）同。

## 第 2 章 创建数据集

### 2.1 数据集的概念

### 2.2 数据结构

#### 2.2.1 向量

<code>a &lt;- c(1, 2, 5, 6)</code>	创建数值型向量 <b>a</b> ，元素为：1，2，5，6
<code>b &lt;- c("one", "two", "three")</code>	创建字符型向量 <b>b</b> ，元素为：one，two，three
<code>c &lt;- c(TRUE, TRUE, FALSE)</code>	创建逻辑型向量 <b>c</b> ，元素为：真，真，假
“:”，如 <code>2 * 1 : 5</code> 返回 2 4 6 8 10	【补】在 R 中 _____ 的运算级别最高
<code>a &lt;- seq(1, 10, 2)</code>	【补】生成从 1 开始，步长为 2，到 10 为止的向量 <b>a</b>
<code>b &lt;- seq(10, 1, -1)</code>	【补】生成从 10 开始，步长为-1，到 1 为止的向量 <b>b</b>
<code>c &lt;- seq(1, by=2, length=10)</code>	【补】生成从 1 开始，步长为 2，包含 10 个元素的向量 <b>c</b>
<code>d &lt;- rep(c(1, 3), 3)</code>	【补】用生成重复元素的函数生成向量 <b>d</b> : 1, 3, 1, 3, 1, 3
<code>e &lt;- rep(c(1, 3), each=3)</code>	【补】用生成重复元素的函数生成向量 <b>e</b> : 1, 1, 1, 3, 3, 3
<code>a[3]</code>	提取向量 <b>a</b> 的第三个元素
<code>a[c(2, 4)]</code>	提取向量 <b>a</b> 的第二个和第四个元素
<code>a[2:6]</code>	提取向量 <b>a</b> 的第二个至第六个元素
<code>a[a&gt;2]</code>	【补】提取向量 <b>a</b> 中大于 2 的元素

#### 2.2.2 矩阵

<code>mymatrix &lt;- matrix(1:20, nrow=5, ncol=4, byrow=TRUE, dimnames=list(c("R1", "R2", "R3", "R4", "R5"), c("C1", "C2", "C3", "C4")))</code>	以 1 至 20 的整数为元素，创建 5 行 4 列的矩阵 <b>mymatrix</b> ，按行填充，行名称依次为 R1, R2, R3, R4, R5，列名称依次为 C1, C2, C3, C4。
<code>mymatrix[2, ]</code> <code>mymatrix[, 2]</code> <code>mymatrix[1, 4]</code> <code>mymatrix[1, c(3, 4)]</code>	分别提取 <b>mymatrix</b> 第二行的元素；第二列的元素；第一行第四列的元素；第一行第三、四列的元素

#### 2.2.3 数组

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

	由 dim1, dim2, dim3 生成三维数组 z, 各维度标签名称为向量自身名称
--	---

## 2.2.4 数据框

<pre>patientdata &lt;- data.frame(patientID, age, diabetes, status)</pre>	<p>已知:</p> <pre>patientID &lt;- c(1, 2, 3, 4) age &lt;- c(25, 34, 28, 52) diabetes &lt;- c("Type1", "Type2", "Type1", "Type1") status &lt;- c("Poor", "Improved", "Excellent", "Poor")</pre> <p>由 patientID, age, diabetes, status 生成数据框 patientdata</p>
<code>patientdata[1:2]</code>	提取 patientdata 的第一列和第二列
<code>patientdata[c("diabetes", "status")]</code>	提取 patientdata 的 diabetes 和 status 变量
<code>table(patientdata\$diabetes, patientdata\$status)</code>	生成 patientdata 中 diabetes 和 status 变量的列联表
<code>new &lt;- patientdata[which(patientdata\$age&gt;30), ]</code>	【补】提取 patientdata 中 age 大于 30 的观测，保存在 new 中 <sup>9</sup>

### 1. attach(), detach() 和 with()

<code>attach(), detach(), with()</code>	为了避免每次都用“数据集名\$”来绑定数据集，可以使用 _____, _____ 和 _____
<pre>attach(mtcars) summary(mpg) plot(mpg, disp) detach(mtcars)</pre>	<p>绑定数据集 mtcars;</p> <p>对变量 mpg 做摘要总结;</p> <p>绘制 mpg 和 disp 的散点图;</p> <p>解除绑定</p>
<pre>with(mtcars, { nokeep &lt;- summary(mpg) keep &lt;-&lt; summary(mpg) })</pre>	<p>绑定数据集 mtcars;</p> <p>对变量 mpg 做摘要总结，保存在 nokeep 中;</p> <p>对变量 mpg 做摘要总结，保存在 keep 中，保证 keep 在 with() 结构之外的全局环境中也能使用;</p> <p>解除绑定</p>

### 2. 实例标识符

<sup>9</sup> 4.10.3 节提供了更多关于选入观测的信息

<code>patientdata &lt;- data.frame(patientID, age, diabetes, status, row.names = patientID)</code>	由 <code>patientID</code> , <code>age</code> , <code>diabetes</code> , <code>status</code> 生成数据框 <code>patientdata</code> , 并用病人编号 ( <code>patientID</code> ) 区分 (命名) 数据集中不同的个体
--	--

## 2.2.5 因子

<code>diabetes &lt;- factor(diabetes)</code>	<p>已知向量:</p> <pre>diabetes &lt;- c("Type1", "Type2", "Type1", "Type1")</pre> <p>将 <code>diabetes</code> 转变成因子 (名义型) (即数字为定类尺度, 相同数字代表一个类别)</p>
<code>status &lt;- factor(status, order=TRUE)</code>	<p>已知向量:</p> <pre>status &lt;- c("Poor", "Improved", "Excellent", "Poor")</pre> <p>将 <code>status</code> 转变成因子 (有序型) (次序默认按字母顺序依次创建)</p>
<code>status &lt;- factor(status, order=TRUE, levels=c("Poor", "Improved", "Excellent"))</code>	<p>已知向量:</p> <pre>status &lt;- c("Poor", "Improved", "Excellent", "Poor")</pre> <p>将 <code>status</code> 转变成因子 (有序型) (自定义 1=Poor, 2=Improved, 3=Excellent)</p>

<code>str(diabetes)</code>	显示变量 <code>diabetes</code> 的结构 (变量类型)
<code>str(patientdata)</code>	显示数据框 <code>patientdata</code> 的结构 (其行列结构、变量组成、各变量类型) <sup>10</sup>

## 2.2.6 列表

<code>mylist &lt;- list(title = g, ages = h, j, k)</code>	<p>已知:</p> <pre>g &lt;- "My First List" h &lt;- c(25, 26, 18, 39) j &lt;- matrix(1:10, nrow = 5) k &lt;- c("one", "two", "three")</pre> <p>生成由以上四个分量组成的列表 <code>mylist</code>, 将 <code>g</code> 命名为 <code>title</code>, 将 <code>h</code> 命</p>
---	--

<sup>10</sup> `str()` 函数是一个实用性很强的命令, 建议牢记

	名为 <code>ages</code>
<code>mylist[[2]]</code>	提取列表 <code>mylist</code> 中的第二个分量
<code>mylist[["ages"]]</code> 【补】或 <code>mylist\$ages</code>	提取列表 <code>mylist</code> 中的 <code>ages</code> 分量
<code>length(mylist)</code>	【补】展现列表 <code>mylist</code> 分量的数目
<code>names(mylist)</code>	【补】展现列表 <code>mylist</code> 各分量名字
<code>mylist[[2]][1:3]</code>	【补】提取列表 <code>mylist</code> 第二个分量的第 1 至第 3 个值

## 2.3 数据的输入

### 2.3.1 使用键盘输入数据

<code>mydata&lt;- data.frame(age=numeric(0), gender=character(0), weight=numeric(0))</code>	创建一个指定模式但不含实际数据的空数据框 <code>mydata</code> ，其含有三个变量 <code>age</code> （数值型）， <code>gender</code> （字符型）， <code>weight</code> （数值型）。
<code>newdata&lt;- edit(mydata)</code>	调用文本编辑器，键入、编辑数据，并保存为新数据框 <code>newdata</code>
<code>fix(mydata)</code>	调用文本编辑器，直接修改 <code>mydata</code>

### 2.3.2 从带分隔符的文本文件中导入数据 <sup>11</sup>

### 2.3.3 导入 Excel 数据 <sup>12</sup>

### 2.3.4 导入 XML 数据

### 2.3.5 从网页抓取数据 <sup>13</sup>

### 2.3.6 导入 SPSS 数据 <sup>14</sup>

### 2.3.7 导入 SAS 数据 <sup>15</sup>

### 2.3.8 导入 Stata 数据 <sup>16</sup>

### 2.3.9 导入 netCDF 数据

### 2.3.10 导入 HDF5 数据

### 2.3.11 访问数据库管理系统

### 2.3.12 通过 Stat/Transfer 导入数据

## 2.4 数据集的标注

<sup>11</sup> 可以在 RStudio 中通过 Environment — Import Dataset — From CSV 实现。在点击 Import Dataset 后出现的面板中，建议为数据集取一个简洁的由英文字母组成的名称，方便导入数据后，直接引用数据集名称调用该数据集。此外，还可以在面板中定义各变量的性质（字符型，数值型，时间等）以及第一行是否作为变量名称。从其他来源导入数据也可参考此操作。

<sup>12</sup> 可以在 RStudio 中通过 Environment — Import Dataset — From Excel 实现。

<sup>13</sup> 《R 语言网络爬虫初学者指南（使用 rvest 包）》（<http://www.ituring.com.cn/article/465317>）细致全面地介绍了用 rvest 包做爬虫。此文中的方法十分容易上手。

<sup>14</sup> 可以在 RStudio 中通过 Environment — Import Dataset — From SPSS 实现。

<sup>15</sup> 可以在 RStudio 中通过 Environment — Import Dataset — From SAS 实现。

<sup>16</sup> 可以在 RStudio 中通过 Environment — Import Dataset — From Stata 实现。

## 2.5 处理数据对象的实用函数

<code>length(object)</code>	显示对象中元素/成分的数量 <sup>17</sup>
<code>dim(object)</code>	显示某个对象的维度 <sup>18</sup>
<code>str(object)</code>	显示某个对象的结构 <sup>19</sup>
<code>class(object)</code>	显示某个对象的类或类型 <sup>20</sup>
<code>mode(object)</code>	显示某个对象的模式
<code>names(object)</code>	显示某对象中各成分的名称
<code>c(object, object, ...)</code>	将对象合并入一个向量
<code>cbind(object, object, ...)</code>	按列合并对象
<code>rbind(object, object, ...)</code>	按行合并对象
<code>Object</code>	输出某个对象
<code>head(object)</code>	列出某个对象的开始部分 <sup>21</sup>
<code>tail(object)</code>	列出某个对象的最后部分 <sup>22</sup>
<code>ls()</code>	显示当前的对象列表
<code>rm(object, object, ...)</code>	删除一个或多个对象
<code>rm(list=ls())</code>	删除当前工作环境中的几乎所有对象
<code>newobject &lt;- edit(object)</code>	编辑对象并另存为 newobject
<code>fix(object)</code>	直接编辑对象

<sup>17</sup> 例如显示某数据框有  $n$  个变量。

<sup>18</sup> 例如显示某数据框是  $n$  行  $m$  列。

<sup>19</sup> 例如显示某数据框是  $n$  行  $m$  列，具体有哪些变量，各变量的类型是什么，各变量有哪些值。

<sup>20</sup> 例如，如果 `object` 是数据框会返回"`data.frame`"。

<sup>21</sup> 默认前 6 行。

<sup>22</sup> 默认后 6 行。

## 第 3 章 图形初阶

### 3.1 使用图形<sup>23</sup>

<pre>attach(mtcars) plot(wt, mpg) abline(lm(mpg~wt)) title("Regression") detach(mtcars)</pre>	<p>绑定 <code>mtcars</code> 数据集；</p> <p>以横轴 <code>wt</code>，纵轴 <code>mpg</code> 作散点图；</p> <p>添加最优拟合曲线；</p> <p>添加标题：Regression；</p> <p>解除绑定</p>
---	--

### 3.2 一个简单的例子

<pre>plot(dose, drugA, type = "b")</pre>	<p>已知：</p> <p><code>dose &lt;- c(20, 30, 40, 45, 60)</code> 为五个水平的药物剂量</p> <p><code>drugA &lt;- c(16, 20, 27, 40, 60)</code> 为对药物 A 的响应</p> <p><code>drugB &lt;- c(15, 18, 25, 31, 40)</code> 为对药物 B 的响应</p> <p>绘制药物 A 的剂量（<code>dose</code>）和响应关系（<code>drugA</code>）的折线图，类型为 <code>b</code></p>
--	---

### 3.3 图形参数

<pre>opar &lt;- par(no.readonly = TRUE) par(lty = 2, pch = 17) plot(dose, drugA, type = "b") par(opar)</pre>	<p>复制一份当前图形参数设置的列表；</p> <p>将线条类型修改成虚线（2），将点符号修改成实心三角（17）；</p> <p>绘制药物 A 的剂量（<code>dose</code>）和响应关系（<code>drugA</code>）的折线图，类型为 <code>b</code>；</p> <p>还原图形参数设置</p>
<pre>plot(dose, drugA, type = "b", lty = 2, pch = 17)</pre>	<p>绘制药物 A 的剂量（<code>dose</code>）和响应关系（<code>drugA</code>）的折线图，类型为 <code>b</code>，线条类型为虚线（2），点符号为实心三角（17）</p>

#### 3.3.1 符号和线条

<code>pch</code>	指定绘制点时使用的符号
<code>cex</code>	指定符号的大小。以相对默认值的大小来表示（默认值为 1）。
<code>lty</code>	指定线条类型

<sup>23</sup> 本节中有关图形保存的命令，可以在 RStudio 中通过 Plots — Export — Save as Image 或 Save as PDF 实现。本节中关于查看多个图形的方法，在 RStudio 中可以通过 Plots 面板中的左、右箭头实现。



<b>lwd</b>	指定线条宽度。以相对默认值的大小来表示（默认值为 1）。
------------	------------------------------

<code>plot(dose, drugA, type = "b", lty = 3, lwd = 3, pch = 15, cex = 2)</code>	绘制药物 A 的剂量（dose）和响应关系（drugA）的折线图，类型为 b，线条类型为点线（3）宽度为默认宽度的 3 倍，点符号为实心正方形（15），点符号大小为默认值的 2 倍
---	--

`plot()`中表示折线图类型的参数 `type` 的可选值<sup>24</sup>：

<b>p</b>	只有空心点
<b>l</b>	只有线
<b>o</b>	实心点和线（即线覆盖在点上）
<b>b</b>	线段和空心点
<b>c</b>	线段（不绘制点）
<b>s</b>	阶梯线（先水平再竖直）
<b>S</b>	阶梯线（先竖直再水平）
<b>h</b>	直方图式的垂直线
<b>n</b>	不生成任何点和线（通常用来为后面的命令创建坐标轴）

### 3.3.2 符号和线型

<b>col</b> <b>lines</b> 和 <b>pie</b> <code>col=c("red", "blue")</code>	默认的绘图颜色； 某些函数（如_____和_____）可以接受一个含有颜色值的向量并自动循环使用； 设置可自动循环使用的含有颜色值的向量，例如在绘制三条线时，则第一条线将为红色，第二条线为蓝色，第三条线又将为红色
<b>col.axis</b>	坐标轴刻度文字的颜色
<b>col.lab</b>	坐标轴标签（名称）的颜色
<b>col.main</b>	标题颜色
<b>col.sub</b>	副标题颜色
<b>fg</b>	图形的前景色
<b>bg</b>	图形的背景色

<b>colors()</b>	返回所有可用颜色的名称
<b>rainbow(10)</b>	生成 10 种连续的“彩虹型”颜色，

<sup>24</sup> 该部分内容为 11.2 节中的内容，在这里提前展示，目的在于更快地上手作图。

<code>gray(0: 10/10)</code>	生成 10 阶灰度色
<code>n &lt;- 10</code> <code>mycolors &lt;- rainbow(n)</code> <code>pie(rep(1, n), labels = mycolors, col = mycolors)</code>	令 <code>n &lt;- 10</code> 生成 <code>n</code> 种连续的“彩虹型”颜色，保存在 <code>mycolors</code> 中； 绘制均分为 <code>n</code> 块的饼图，颜色采用 <code>mycolors</code> ，并在每个色块旁注释颜色的编号
<code>n &lt;- 10</code> <code>mygrays &lt;- gray(0: n/n)</code> <code>pie(rep(1, n), labels = mygrays, col = mygrays)</code>	令 <code>n &lt;- 10</code> 生成 10 阶灰度色，保存在 <code>mygrays</code> 中； 绘制均分为 <code>n</code> 块的饼图，颜色采用 <code>mygrays</code> ，并在每个色块旁注释颜色的编号

### 3.3.3 文本属性

<code>cex</code>	表示相对于默认大小缩放倍数的数值。默认大小为 1，1.5 表示放大为默认值的 1.5 倍，0.5 表示缩小为默认值的 50%，等等
<code>cex.axis</code>	坐标轴刻度文字的缩放倍数。类似于 <code>cex</code>
<code>cex.lab</code>	坐标轴标签（名称）的缩放倍数。类似于 <code>cex</code>
<code>cex.main</code>	标题的缩放倍数。类似于 <code>cex</code>
<code>cex.sub</code>	副标题的缩放倍数。类似于 <code>cex</code>

<code>font</code> 常规=1; 粗体=2; 斜体=3 粗斜体=4; 符号字体=5	整数。用于指定绘图使用的字体样式。 常规=_____; 粗体=_____; 斜体=_____; 粗斜体=_____; 符号字体=_____
<code>font.axis</code>	坐标轴刻度文字的字体样式
<code>font.lab</code>	坐标轴标签（名称）的字体样式
<code>font.main</code>	标题的字体样式
<code>font.sub</code>	副标题的字体样式
<code>ps</code> <code>ps*cex</code>	字体磅值（1 磅约为 1/72 英寸）； 文本的最终大小为_____

family	绘制文本时使用的字体族。标准的取值为 serif（衬线）、sans（无衬线）和 mono（等宽） <sup>25</sup>
--------	--

### 3.3.4 图形尺寸与边界尺寸

pin	以英寸表示的图形尺寸（宽和高）
pin=c(4, 3)	生成一幅 4 英寸宽，3 英寸高的图形。
mai	以数值向量表示的边界大小，顺序为“下、左、上、右”，单位为英寸
main=c(1, 0.5, 1, 0.2)	生成一幅下、左、上、右边界分别为 1, 0.5, 1, 0.2 英寸的图形。
mar c(5, 4, 4, 2) + 0.1	以数值向量表示的边界大小，顺序为“下、左、上、右”，单位为英寸。默认值为_____

## 3.4 添加文本、自定义坐标轴和图例

<pre>plot(dose, drugA, type = "b",       col = "red",       lty = 2, pch = 2, lwd = 2,       main = "Clinical Trials for Drug A",       sub = "This is hypothetical data",       xlab = "Dosage",       ylab = "Drug Response",       xlim = c(0, 60),       ylim = c(0, 70))</pre>	<p>以 dose 为横轴变量，drugA 为纵轴变量作 b 类折线图，颜色为红色，</p> <p>线型为 2，点符号为 2，线宽为 2，</p> <p>主标题为 Clinical Trials for Drug A，</p> <p>副标题为 This is hypothetical data，</p> <p>横轴标签（名称）为 Dosage，</p> <p>纵轴标签（名称）为 Drug Response，</p> <p>横轴范围为 0 至 60，</p> <p>纵轴范围为 0 至 70</p>
ann=FALSE	某些高级绘图函数已经包含了默认的标题和标签。可以通过在 plot() 语句或单独 par() 语句中添加_____来移除它们。

### 3.4.1 标题

<pre>title title(main="main title", sub="subtitle",       xlab="x-axis label", ylab="y-axis label")</pre>	<p>可以使用_____函数为图形添加<sup>26</sup>标题和坐标轴标签。其语法格式为_____</p>
<pre>title(main="My Title", col.main="red",       sub="My Subtitle", col.sub="blue",       xlab="My X label",</pre>	<p>用上述函数生成主标题 My Title，颜色为红色，</p> <p>副标题 My Subtitle，颜色为蓝色，</p>

<sup>25</sup> serif（衬线）、sans（无衬线）和 mono（等宽）可以在 Word 中找到对应的字体：serif（衬线）对应的是 Times New Roman；sans（无衬线）对应的是 Arial；mono（等宽）对应的是 Courier New。

<sup>26</sup> 即在画图时不添加主、副标题和坐标轴标签，而完成作图后用 title() 在作好的图上添加标题。

<code>ylab="My Y label", col.lab="green", cex.lab=0.75)</code>	<p>横轴标签（名称）为 My X label,</p> <p>纵轴标签（名称）为 My Y label,</p> <p>横、纵轴标签的颜色为绿色,</p> <p>横、纵轴标签的字体大小为默认大小的 75%</p>
--	---

### 3.4.2 坐标轴

<b>axis</b>	可以使用_____函数为图形创建自定义的坐标轴
-------------	-------------------------

axis()函数的可选参数包括:

<b>side</b> 下=1, 左=2, 上=3, 右=4	<p>一个整数,表示在图形的哪边绘制坐标轴</p> <p>下=_____</p> <p>左=_____</p> <p>上=_____</p> <p>右=_____</p>
<b>at</b>	一个数值型向量,表示需要绘制刻度线的位置
<b>labels</b> 直接使用 at 中的值	一个字符型向量,表示置于刻度线旁边的文字标签(如果为 NULL,则_____)
<b>pos</b>	坐标轴线绘制位置的坐标(即与另一条坐标轴相交位置的值) <sup>27</sup>
<b>lty</b>	线条类型
<b>col</b>	线条和刻度线颜色
<b>las</b> 平行于(=0)坐标轴 垂直于(=2)坐标轴	标签是否平行于(=_____)或垂直于(=_____)坐标轴 <sup>28</sup>
<b>tck</b> 负值表示在图形外侧 正值表示在图形内侧 0 表示禁用刻度 1 表示绘制网格线 默认值为 -0.01	<p>刻度线的长度,以相对于绘图区域大小的分数表示</p> <p>(负值表示_____,正值表示_____,0 表示_____,1 表示_____);默认值为_____</p>

<b>axes=FALSE</b>	禁用全部坐标轴,包括坐标轴框线
<b>xaxt="n"</b>	去除 x 轴刻度,但留下框线
<b>yaxt="n"</b>	去除 y 轴刻度,但留下框线

<sup>27</sup> 当 pos 为 NULL 时,默认刻度线及其附属的坐标轴线是绘制在对应变量所在的,处于图像四周的边上的。如果定义了 pos=c(n,m),则刻度线及其附属的坐标轴线会挪至 c(n,m)所定义的位置。例如,在代码清单 3-2 中,可以尝试 axis(2, at = x, labels = x, col.axis = "red", las = 2,pos=c(3,0))的效果。

<sup>28</sup> las 可赋予的值并不只有 0 和 2。在一些情况下,若 las=0,横轴和纵轴标签都平行于各自的标签轴;若 las=1,横轴标签平行于 x 轴,而纵轴标签垂直于 y 轴;若 las=2,横轴和纵轴标签都垂直于各自的标签轴;若 las=3,横轴标签垂直于 x 轴,而纵轴标签平行于 y 轴。具体可参考 9.3.2 节 Q-Q 图的注释。

<code>ann=FALSE</code>	去掉横轴和纵轴的默认标签（禁用所有标题和标签）
------------------------	-------------------------

自定义坐标轴的示例（代码清单 3-2）

<code>opar &lt;- par(no.readonly = TRUE)</code>	<p>已知：</p> <pre>x &lt;- c(1:10) y &lt;- x z &lt;- 10/x</pre> <p>复制一份当前图形参数设置的列表</p>
<code>par(mar = c(5, 4, 4, 8) + 0.1)</code>	增加右边界大小至 8，其余边界保持默认大小
<code>plot(x, y, type = "b", pch = 21, col = "red", yaxt = "n", lty = 3, ann = FALSE)</code>	绘制横轴为 $x$ ，纵轴为 $y$ 的折线图，类型为 $b$ ，点符号为 21，颜色为红色，去除 $y$ 轴刻度但留下框线，线条类型为 3，去掉横轴和纵轴的默认标签（禁用所有标题和标签）
<code>lines(x, z, type = "b", pch = 22, col = "blue", lty = 2)</code>	添加横轴为 $x$ ，纵轴为 $z$ 的折线图，类型为 $b$ ，点符号为 22，颜色为蓝色，线条类型为 2
<code>axis(2, at = x, labels = x, col.axis = "red", las = 2)</code>	在图形左侧边绘制坐标轴，刻度线刻度标定 $x$ ，刻度线文字标签为 $x$ ，坐标轴颜色为红色，刻度线标签垂直于坐标轴
<code>axis(4, at = z, labels = round(z, digits = 2), col.axis = "blue", las = 2, cex.axis = 0.7, tck = -0.01)</code>	在图形右侧边绘制坐标轴，刻度线刻度标定 $z$ ，刻度线文字标签为 $z$ （四舍五入，小数点后保留 2 位有效数字），坐标轴颜色为蓝色，刻度线标签垂直于坐标轴，刻度线文字标签字体大小为默认大小的 0.7 倍，刻度线长度为-0.01
<code>mtext("y=1/x", side = 4, line = 3, cex.lab = 1, las = 2, col = "blue")</code>	在图形左侧添加文本“ $y=1/x$ ”，文本外移 3，坐标轴标签（名称）缩放倍数为 1，本文与坐标轴垂直，颜色为蓝色
<code>title("An Example of Creative Axes", xlab = "X values", ylab = "Y=X")</code>	为图形添加标题“An Example of Creative Axes”，横轴标题为“X value”，纵轴标题为“Y=X
<code>par(opar)</code>	恢复图形参数设置

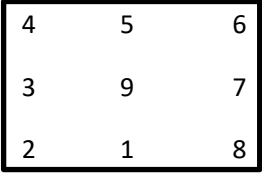
次要刻度线

Hmisc 包的 <code>minor.tick()</code>	_____ 包的 _____ 函数可以添加次要刻度线
<code>minor.tick(nx=2, ny=3, tick.ratio=0.5)</code>	添加次要刻度线，在 $x$ 轴每两条主刻度线之间添加 1 条次要刻度线，在 $y$ 轴每两条主刻度线之间添加 2 条次要刻度线，次要刻度线的长度是主刻度线长度的一半

### 3.4.3 参考线

<code>abline(h=yvalues, v=xvalues)</code>	_____函数可以添加参考线
<code>abline(h=c(1, 5, 7))</code>	在 y 为 1、5、7 的位置添加水平实线
<code>abline(v=seq(1, 10, 2), lty=2, col="blue")</code>	在 x 为 1、3、5、7、9 的位置添加了垂直的蓝色虚线

### 3.4.4 图例

<code>legend(location, title, legend, ...)</code>	_____函数可以添加图例
<code>locator(1)</code>	若想用鼠标单击确定图例的位置，需要在 <code>legend ( )</code> 函数中给出参数_____
1: "bottom" 2: "bottomleft" 3: "left" 4: "topleft" 5: "top" 6: "topright" 7: "right" 8: "bottomright" 9: "center"  <code>inset=</code>	 <p>若在上图中数字 1 至 9 所处的位置放置图例，需要分别在 <code>legend ( )</code> 函数中输入的关键字是_____</p> <p>如果使用了以上某个关键字，那么可以同时使用参数_____指定图例向图形内侧移动的大小（以绘图区域大小的分数表示）</p>
<code>title=" "</code>	为图例添加标题的参数是
<code>legend("topleft", inset=.05, title="Drug Type", c("A", "B"), lty=c(1, 2), pch=c(15, 17), col=c("red", "blue"))</code>	添加图例：图例处于左上方，向图形内侧移动 0.05，图例标题为 Drug Type，图例标签分别为 A，B；线条类型分别为 1，2；点符号分别为 15，17；颜色分别为红色，蓝色

### 3.4.5 文本标注

<code>text(location, "text to place", pos, ...)</code>	_____函数可以在绘图区域内部添加文本
<code>mtext("text to place", side, line=n, ...)</code>	_____函数可以在图形的四个边界之一添加文本

text 的可选参数

<code>location</code> <code>locator(1)</code>	_____是文本的位置参数。可为一对 x、y 坐标，也可通过指定 _____为使用鼠标交互式地确定摆放位置
<code>pos</code> 下=1 左=2 上=3 右=4	_____可控制文本相对于位置参数的方位。  下= 左= 上=

<code>offset=</code>	右=  在指定了文本相对于位置参数的方位后, 可以用_____参数作为偏移量, 以相对于单个字符宽度的比例表示
----------------------	---

#### mtext 的可选参数

<code>side</code> 下=1 左=2 上=3 右=4  <code>line=</code>  <code>adj=0</code> <code>adj=1</code>	_____可指定用来放置文本的边  下= 左= 上= 右=  可以指定参数_____来内移或外移文本, 随着值的增加, 文本将外移。  可使用_____将文本向左下对齐, 或使用_____将文本向右上对齐
---	--

<code>attach(mtcars)</code> <code>plot(wt, mpg, pch = 18, col = "blue")</code>	绑定 <code>mtcars</code> 数据集, 绘制 <code>wt</code> 为横轴, <code>mpg</code> 为纵轴的散点图, 点符号为 18, 颜色为蓝色
<code>text(wt, mpg, row.names(mtcars), cex = 0.6, pos = 4, col = "red")</code>	在上图的每一个点的右侧添加文本, 文本为每个点在 <code>mtcars</code> 中的行名称, 字体大小为默认大小的 60%, 颜色为红色

<code>plot(1:7, 1:7, type = "n")</code>	绘制 7*7 单位的坐标轴, 内容为空
<code>text(4, 4, family = "mono", "Example of mono-spaced text")</code>	在上图的 (4, 4) 坐标处添加文字 <code>Example of mono-spaced text</code> , 字体为 <code>mono</code>

### 3.5 图形的组合

<code>par( )</code> 或 <code>layout( )</code>	使用_____函数或_____函数能够组合多幅图形为一幅总括图形
--	----------------------------------

<code>mfrow=c(nrows, ncols)</code>	在 <code>par( )</code> 函数中使用_____参数来创建按行填充的、行数为 <code>nrows</code> 、列数为 <code>ncols</code> 的图形矩阵
<code>mfcol=c(nrows, ncols)</code>	在 <code>par( )</code> 函数中使用_____参数来创建按列填充的、行数为 <code>nrows</code> 、列数为 <code>ncols</code> 的图形矩阵

使用 `par()` 函数组合图形：

<pre>attach(mtcars) opar &lt;- par(no.readonly = TRUE) par(mfrow = c(2, 2)) plot(wt, mpg) plot(wt, disp) hist(wt) boxplot(wt) par(opar) detach(mtcars)</pre>	<p>创建四幅图形并将其排布在两行两列的图形矩阵中：</p> <p>绑定 <code>mtcars</code> 数据集，</p> <p>复制一份当前图形参数设置的列表，</p> <p>创建 2 行 2 列按行填充的图形矩阵，</p> <p>第一幅图为 <code>wt</code> 为横轴，<code>mpg</code> 为纵轴的散点图，</p> <p>第二幅图为 <code>wt</code> 为横轴，<code>disp</code> 为纵轴的散点图，</p> <p>第三幅图为 <code>wt</code> 的直方图，</p> <p>第四幅图为 <code>wt</code> 的箱线图，</p> <p>还原图形参数</p> <p>解除绑定 <code>mtcars</code> 数据集</p>
<pre>attach(mtcars) opar &lt;- par(no.readonly = TRUE) par(mfrow = c(3, 1)) hist(wt) hist(mpg) hist(disp) par(opar) detach(mtcars)</pre>	<p>创建三幅图形并将其排布在三行一列的图形矩阵中：</p> <p>绑定 <code>mtcars</code> 数据集，</p> <p>复制一份当前图形参数设置的列表，</p> <p>创建 3 行 1 列按行填充的图形矩阵，</p> <p>第一幅图为 <code>wt</code> 的直方图，</p> <p>第二幅图为 <code>mpg</code> 的直方图，</p> <p>第三幅图为 <code>disp</code> 的直方图，</p> <p>还原图形参数</p> <p>解除绑定 <code>mtcars</code> 数据集</p>
<pre>main=" " ann=FALSE</pre>	<p>高级绘图函 <code>hist()</code> 包含了一个默认的标题（使用_____可以禁用它，抑或使用_____来禁用所有标题和标签）。</p>

使用 `layout()` 函数组合图形：

<pre>attach(mtcars) layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))<sup>29</sup> hist(wt) hist(mpg)</pre>	<p>绑定 <code>mtcars</code> 数据集，</p> <p>生成 2 行 2 列的图形矩阵，按行填充，其中第一幅图占据第一行，第二幅图占据矩阵第二行第一列的位置，第三幅图占据矩阵第二</p>
---	---

<sup>29</sup> `layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE))`意味着生成了 2\*2 的矩阵，并且按行填充，而 `matrix(c(1, 1, 2, 3))`参数定义了图 1，图 2，图 3 处于这个 2\*2 的矩阵中的位置（1 代表图 1 的位置，2 代表图 2 的位置，3 代表图 3 的位置）：

1	1
2	3

也就是：

1
2    3

又如 `matrix(c(1, 2, 1, 3))`参数定义图 1，图 2，图 3 的位置是（1 代表图 1 的位置，2 代表图 2 的位置，3 代表图 3 的位置）：

1	2
1	3

也就是：

1	2
	3



<pre>hist(displ) detach(mtcars)</pre>	<p>行第二列的位置，</p> <p>第一幅图为 <b>wt</b> 的直方图，</p> <p>第二幅图为 <b>mpg</b> 的直方图，</p> <p>第三幅图为 <b>displ</b> 的直方图，</p> <p>解除绑定 <b>mtcars</b> 数据集</p>
<pre>attach(mtcars) layout(matrix(c(1, 1, 2, 3), 2, 2, byrow = TRUE), widths = c(3, 1), heights = c(1, 2)) hist(wt) hist(mpg) hist(displ) detach(mtcars)</pre>	<p>绑定 <b>mtcars</b> 数据集，</p> <p>生成 2 行 2 列的图形矩阵，按行填充，其中第一幅图占据第一行，第二幅图占据矩阵第二行第一列的位置，第三幅图占据矩阵第二行第二列的位置，其中，第一列占全图宽度的<math>\frac{3}{4}</math>，第二列占全图宽度的<math>\frac{1}{4}</math>，第一行占全图高度的<math>\frac{1}{3}</math>，第二行占全图高度的<math>\frac{2}{3}</math>。<sup>30</sup></p> <p>第一幅图为 <b>wt</b> 的直方图，</p> <p>第二幅图为 <b>mpg</b> 的直方图，</p> <p>第三幅图为 <b>displ</b> 的直方图，</p> <p>解除绑定 <b>mtcars</b> 数据集</p>

图形布局的精细控制：

<pre>fig=</pre>	<p>可以在 <b>par()</b> 函数中定义_____参数实现图形布局的精细控制</p>
<pre>opar &lt;- par(no.readonly = TRUE) par(fig = c(0, 0.8, 0, 0.8)) plot(mtcars\$wt, mtcars\$mpg) par(fig = c(0, 0.8, 0.55, 1), new = TRUE) boxplot(mtcars\$wt, horizontal = TRUE, axes = FALSE) par(fig = c(0.65, 1, 0, 0.8), new = TRUE) boxplot(mtcars\$mpg, axes = FALSE) mtext("Enhanced Scatterplot", side = 3, outer = TRUE, line = -3) par(opar)</pre>	<p>多幅图形布局的精细控制：</p> <p>复制一份当前图形参数设置的列表：</p> <p>设定第一幅图的位置为处于横轴 0 至 0.8，纵轴 0 至 0.8 的位置；</p> <p>第一幅图：针对 <b>mtcars</b> 数据集，作横轴为 <b>wt</b>，纵轴为 <b>mpg</b> 的散点图；</p> <p>设定第二幅图的位置为处于横轴 0 至 0.8，纵轴 0.55 至 1 的位置，添加至第一幅图；</p> <p>第二幅图：针对 <b>mtcars</b> 数据集，作 <b>wt</b> 的箱线图，水平摆放，去除外侧框线；</p> <p>设定第三幅图的位置为处于横轴 0.65 至 1，纵轴 0 至 0.8 的位置，添加至第一幅图；</p>

<sup>30</sup> 这里采用了一种更直观的表述，原文的描述是：“第 1 行中图形的高度是第 2 行中图形高度的二分之一。除此之外，右下角图形的宽度是左下角图形宽度的三分之一”。

	第三幅图：针对 <code>mtcars</code> 数据集，作 <code>mpg</code> 的箱线图，去除外侧框线； 在图形的上方添加文本 <code>Enhanced Scatterplot</code> ，使用外侧边缘 <sup>31</sup> ， 内移文本 3 个单位； 还原图形参数
在整个图形的范围内修改各个子图占据的区域位置 and 大小	如果遇到了“ <code>Error in plot.new( ) : figure margins too large</code> ”这样的错误，可以尝试_____

---

<sup>31</sup> 使用外侧边缘的参数为 `outer=TRUE`，其作用是把文本放在整体组合图形的外侧，而不是图形组合内部某小图的外侧。例如，这里如过使用命令 `mtext("Enhanced Scatterplot", side = 3)`将会把 `Enhanced Scatterplot` 添加到第三幅图的上方，而不是整个图形组合的上方。

## 第 4 章 基本数据管理

### 4.1 一个示例

<pre>leadership &lt;- data.frame(manager, date, country, gender, age, q1, q2, q3, q4, q5, stringsAsFactors = FALSE)</pre>	<p>已知：</p> <pre>manager &lt;- c(1, 2, 3, 4, 5) date &lt;- c("10/24/08", "10/28/08", "10/1/08", "10/12/08", "5/1/09") country &lt;- c("US", "US", "UK", "UK", "UK") gender &lt;- c("M", "F", "F", "M", "F") age &lt;- c(32, 45, 25, 39, 99) q1 &lt;- c(5, 3, 3, 3, 2) q2 &lt;- c(4, 5, 5, 3, 2) q3 &lt;- c(5, 2, 5, 4, 1) q4 &lt;- c(5, 5, 5, NA, 2) q5 &lt;- c(5, 5, 2, NA, 1)</pre> <p>创建包含以上变量的数据框 leadership, 要求 date, country, gender 三个变量为字符型而不是因子型。</p>
---	---

### 4.2 创建新变量

算术运算符：

+	加
-	减
*	乘
/	除
^或**	求幂
x%%y 1	求余 5%%2 的结果为_____
x%/%y 2	整数除法 5%/%2 的结果为_____

<p>方法一：</p> <pre>mydata\$sumx &lt;- mydata\$x1 + mydata\$x2 mydata\$meanx &lt;- (mydata\$x1 + mydata\$x2) /</pre>	<p>已知：</p> <pre>mydata &lt;- data.frame (x1=c(2,2,6,4), x2=c(3,4,2,8))</pre>
---	--

2 方法二： <code>attach(mydata)</code> <code>mydata\$sumx &lt;- x1 + x2</code> <code>mydata\$meanx &lt;- (x1 + x2)/2</code> <code>detach(mydata)</code> 方法三： <code>mydata &lt;- transform(mydata,</code> <code>sumx = x1 + x2,</code> <code>meanx = (x1 + x2)/2)</code>	使用三种方法在 <code>mydata</code> 数据框中创建新变量 <code>sumx</code> ( <code>x1</code> 和 <code>x2</code> 的和) 和 <code>meanx</code> ( <code>x1</code> 和 <code>x2</code> 的平均数)
--	--

### 4.3 变量的重编码

逻辑运算符：

<code>&lt;</code>	小于
<code>&lt;=</code>	小于或等于
<code>&gt;</code>	大于
<code>&gt;=</code>	大于或等于
<code>==</code>	严格等于
<code>!=</code>	不等于
<code>!x</code>	非 <code>x</code>
<code>x y</code>	<code>x</code> 或 <code>y</code>
<code>x&amp;y</code>	<code>x</code> 和 <code>y</code>
<code>isTRUE(x)</code>	测试 <code>x</code> 是否为 <code>TRUE</code>

<code>leadership\$age[leadership\$age == 99] &lt;- NA</code>	针对 <code>leadership</code> 数据集，将 <code>age</code> 变量中的 99 岁的年龄值重编码为缺失值
格式一： <code>leadership\$agecat[leadership\$age &gt; 75] &lt;- "Elder"</code> <code>leadership\$agecat[leadership\$age &gt;= 55 &amp; leadership\$age &lt;= 75] &lt;- "Middle Aged"</code> <code>leadership\$agecat[leadership\$age &lt; 55] &lt;- "Young"</code> 格式二： <code>leadership &lt;- within(leadership, {</code> <code>  agecat &lt;- NA</code> <code>  agecat[age &gt; 75] &lt;- "Elder"</code> <code>  agecat[age &gt;= 55 &amp; age &lt;= 75] &lt;- "Middle Aged"</code> <code>})</code>	用两种命令格式将 <code>leadership</code> 数据集中 <code>age</code> 变量重编码为类别型变量 <code>agecat</code> ( <code>Young</code> 、 <code>Middle Aged</code> 、 <code>Elder</code> )。其中，年轻人 ( <code>Young</code> ) 定义为小于 55 岁，中年人 ( <code>Middle Aged</code> ) 定义为 55 到 75 岁，老年人 ( <code>Elder</code> ) 定义为大于 75 岁。

<code>agecat[age &lt; 55] &lt;- "Young" })</code>	
---	--

## 4.4 变量的重命名

<code>fix(leadership)</code>	调用交互的编辑器来重命名
<code>rename( )</code>  <code>library(reshape)</code> <code>leadership &lt;- rename(leadershi p, c(manag er="managerID", date="testDate"))</code>	reshape 包的_____函数用于修改变量名 <sup>32</sup>  使用上述函数修改 leadership 数据框的 manager 和 date 变量的名称, 其中 manager 修改为 managerID, date 修改为 testDate。
<code>names(leadershi p)</code> <code>names(leadershi p)[2] &lt;- "testDate"</code> <code>names(leadershi p)[6:10] &lt;- c("item1", "i tem2", "item3", "item4", "item5")</code>	用 name( )函数修改变量名:  获取 leadership 数据集的变量名;  将 leadership 数据集的第二列变量的变量名改为 testDate;  将 leadership 数据集的第六列至第十列变量的变量名改为 item1, item2, item3, item4, item5

## 4.5 缺失值

<code>i s. na(y)</code> 将返回 <code>c(FALSE, FALSE, FALSE, TRUE)</code>	已知  <code>y &lt;- c(1, 2, 3, NA)</code>  检测 y 中的缺失值, 将返回_____
<code>i s. na(leadershi p[, 6:10])</code>	检测数据集 leadership 第 6 列至第 10 列的缺失值

### 4.5.1 重编码某些值为缺失值

<code>leadershi p\$age[leadershi p\$age == 99] &lt;- NA</code>	将 leadership 数据集 age 变量中等于 99 的值替换为 NA
--	--

### 4.5.2 在分析中排除缺失值

NA	已知:  <code>x &lt;- c(1, 2, NA, 3)</code>  <code>y &lt;- x[1] + x[2] + x[3] + x[4]</code>  <code>z &lt;- sum(x)</code>  y 和 z 的返回值是_____
<code>na.rm=TRUE</code>	_____参数可以在计算之前移除缺失值并使用剩余值进行计算
<code>y &lt;- sum(x, na. rm=TRUE)</code>	已知:

<sup>32</sup> 在原书第二版中, 作者介绍的是 plyr 包中的 rename( )函数。实际上, reshape 包中的 rename( )函数和 plyr 包中的 rename( )函数的语法格式、功能、结果是相同的。

	<pre>x &lt;- c(1, 2, NA, 3)</pre> <p>计算移除向量 <b>x</b> 中缺失值后的平均值</p>
<code>na.omit( )</code>	_____函数可以删除所有含有缺失数据的行
<code>newdata &lt;- na.omit(leadership)</code>	删除 <b>leadership</b> 数据集所有含有缺失数据的行, 保存在 <b>newdata</b> 中

## 4.6 日期值

<pre>as.Date()</pre> <pre>as.Date(x, "input_format")</pre> <p>"input_format" 对应 <b>X</b> 的格式</p>	<p>_____函数可以将字符串形式的日期值转换成数值形式的日期变量, 解释其语法格式</p>
--	---

input\_format 的可选日期格式:

%d	数字表示的日期 (0~31)
%a	缩写的星期名 (Mon)
%A	非缩写星期名 (Monday)
%m	月份 (00~12)
%b	缩写的月份 (Jan)
%B	非缩写月份 (January)
%y	两位数的年份 (07)
%Y	四位数的年份 (2007)

yyyy-mm-dd	as.Date( ) 的默认输入格式为_____
<pre>mydates &lt;- as.Date(c("2007-06-22", "2004-02-13"))</pre>	将 c("2007-06-22", "2004-02-13") 转化为日期格式, 保存在 <b>mydates</b> 中
<pre>dates &lt;- as.Date(strDates, "%m/%d/%Y")</pre>	<p>已知:</p> <pre>strDates &lt;- c("01/05/1965", "08/16/1975")</pre> <p>将 <b>strDates</b> 转化为日期格式, 保存在 <b>dates</b> 中</p>

<code>Sys.Date( )</code>	_____函数可以返回当天的日期,
<code>date( )</code>	_____函数可以返回当前的日期和时间

<code>format(x, format="output_format")</code>	_____函数可接受一个参数 (日期) 并按某种格式 (参照
--	--------------------------------

	input_format 的可选日期格式) 输出结果
today <- Sys.Date() format(today, format="%B %d %Y")	获得当日日期, 并输出为“非缩写月份 数字表示的日期 四位数年份”格式的日期值
today <- Sys.Date() format(today, format="%A")	获得当日日期, 并输出为“非缩写星期名”格式的日期值

1970 年 1 月 1 日 天数 负数	R 的内部在存储日期时, 是使用自_____以来的_____表示的, 更早的日期则表示为_____。
days <- enddate - startdate	已知: startdate <- as.Date("2004-02-13") enddate <- as.Date("2011-01-22")  求 startdate 和 enddate 间相差的天数, 保存在 days 中
difftime(today, dob, units="weeks")	已知: today <- Sys.Date( ) dob <- as.Date("1956-10-12")  求 today 和 dob 间相差的周数
format(dob, format="%A")	已知: dob <- as.Date("1956-10-12")  求 dob 是星期几

#### 4.6.1 将日期转换为字符型变量

as.character( )	函数可将日期值转换为字符型
-----------------	---------------

#### 4.6.2 更进一步

### 4.7 类型转换

类型转换函数<sup>33</sup>:

is.numeric( )	as.numeric( )	判断是否是/转换为 数值型
is.character( )	as.character( )	判断是否是/转换为 字符型
is.vector( )	as.vector( )	判断是否是/转换为 向量

<sup>33</sup> 这个表中提供的函数只能一个一个去试某个对象是否属于某类型, 要想直接得知某个对象所属的类型, 可以用 str( )或 mode( )函数。

<code>is.matrix( )</code>	<code>as.matrix( )</code>	判断是否是/转换为 矩阵
<code>is.data.frame( )</code>	<code>as.data.frame( )</code>	判断是否是/转换为 数据框
<code>is.factor( )</code>	<code>as.factor( )</code>	判断是否是/转换为 因子
<code>is.logical( )</code>	<code>as.logical( )</code>	判断是否是/转换为 逻辑型

<pre>is.numeric(a) is.vector(a) a &lt;- as.character(a) is.character(a)</pre>	<p>已知：</p> <pre>a &lt;- c(1,2,3)</pre> <p>判断是 a 否是数值型；</p> <p>判断是 a 否是向量；</p> <p>把 a 转换为字符型；</p> <p>判断是 a 否是字符型</p>
---	---

## 4.8 数据排序

<pre>newdata &lt;- leadership[order(leadership\$age), ]</pre>	将 leadership 数据框按照 age 变量升序排列，生成新数据框 newdata
<pre>attach(leadership) newdata &lt;- leadership[order(gender, -age), ] detach(leadership)</pre>	<p>绑定数据框 leadership；</p> <p>将 leadership 数据框按照女性到男性<sup>34</sup>、同样性别中按年龄降序排序，生成新数据框 newdata；</p> <p>解除绑定</p>

## 4.9 数据集的合并

### 4.9.1 添加列

merge( )	_____函数可以通过一个或多个共有变量的联结，横向合并两个数据框（数据集）												
total <- merge(dataframeA, dataframeB, by="ID")	将 dataframeA 数据框和 dataframeB 数据框按照 ID 进行合并												
total <- merge(dataframeA, dataframeB, by=c("ID", "Country"))	将 dataframeA 数据框和 dataframeB 数据框按照 ID 和 Country 进行合并												
<table><tr><td></td><td>b</td><td>a. x</td><td>a. y</td></tr><tr><td>1</td><td>1</td><td>Zhao</td><td>Wu</td></tr><tr><td>2</td><td>2</td><td>Qian</td><td>Zeng</td></tr></table>		b	a. x	a. y	1	1	Zhao	Wu	2	2	Qian	Zeng	<p>【补】已知：</p> <p>x=data.frame(a=c("Zhao","Qian","Sun","Li","Ma"),b=c(1,2,19,4,5))</p> <p>y=data.frame(a=c("He","Wu","Zeng","Wang","Tang"),b=c(5,1,2,19,4))</p>
	b	a. x	a. y										
1	1	Zhao	Wu										
2	2	Qian	Zeng										

<sup>34</sup> 实则按字母表顺序排序，若第一个字母相同则比较第二个字母的顺序，以此类推。Female（F）较 Male（M）先于字母表排列。



3	4	Sun	Tang	命令 <code>total=merge(x,y,by="b")</code> 的返回结果是_____																				
4	5	Li	He																					
5	19	Ma	Wang																					
<table><tr><td></td><td>b</td><td>a. x</td><td>a. y</td></tr><tr><td>1</td><td>1</td><td>Zhao</td><td>Wu</td></tr><tr><td>2</td><td>2</td><td>Qian</td><td>Zeng</td></tr><tr><td>3</td><td>4</td><td>Li</td><td>Tang</td></tr><tr><td>4</td><td>5</td><td>Ma</td><td>He</td></tr></table> <p>【补】已知： <code>x=data.frame(a=c("Zhao","Qian","Sun","Li","Ma"),b=c(1,2,19,4,5))</code> <code>y=data.frame(a=c("He","Wu","Zeng","Wang","Tang"),b=c(5,1,2,0,4))</code>  命令 <code>total=merge(x,y,by="b")</code> 的返回结果是_____<sup>35</sup></p>						b	a. x	a. y	1	1	Zhao	Wu	2	2	Qian	Zeng	3	4	Li	Tang	4	5	Ma	He
	b	a. x	a. y																					
1	1	Zhao	Wu																					
2	2	Qian	Zeng																					
3	4	Li	Tang																					
4	5	Ma	He																					
<table><tr><td></td><td>b</td><td>c</td><td>a. x</td><td>a. y</td></tr><tr><td>1</td><td>5</td><td>10</td><td>Ma</td><td>He</td></tr></table> <p>【补】已知： <code>x=data.frame(a=c("Zhao","Qian","Sun","Li","Ma"),b=c(1,2,19,4,5),c=6:10)</code> <code>y=data.frame(a=c("He","Wu","Zeng","Wang","Tang"),b=c(5,1,2,19,4),c=10:6)</code>  命令 <code>total=merge(x,y,by=c("b","c"))</code> 的返回结果是_____</p>						b	c	a. x	a. y	1	5	10	Ma	He										
	b	c	a. x	a. y																				
1	5	10	Ma	He																				
<p><code>total &lt;- cbind(A, B)</code></p> <p>当数据框 A 和数据框 B 的行数（观测）相同时，_____函数可以机械地把 A 和 B 横向合并（即不考虑共有变量的联结，单纯地把 A 和 B 拼起来），保存在 total 中</p>																								

#### 4.9.2 添加行

<code>total &lt;- rbind(dataframeA, dataframeB)</code>	_____函数可以纵向合并两个数据框（数据集）dataframeA,和 dataframeB，保存在 total 中
<p>相同的变量 顺序</p> <p>删除 dataframeA 中的多余变量 在 dataframeB 中创建追加的变量并将其值设为 NA（缺失）</p>	<p><code>rbind()</code> 要求两个数据框必须拥有_____，不过它们的_____不必一定相同。</p> <p>如果 dataframeA 中拥有 dataframeB 中没有的变量，可以_____或者_____</p>

### 4.10 数据集取子集

#### 4.10.1 选入（保留）变量（列）

<code>newdata &lt;- leadership[, c(6:10)]</code>	选入 leadership 数据框的第 6 列至第 10 列，保存在 newdata 中
<code>newdata &lt;- leadership[c("q1", "q2", "q3", "q4", "q5")]</code>	选入 leadership 数据框的 q1、q2、q3、q4 和 q5 变量，保存在 newdata 中

<sup>35</sup> 在 x\_b 的 19 和 y\_b 的 0 在对方集合中找不到对应时，merge 后的数据框中不再含有相应的观测。

	中
<code>paste("q", 1:5, sep="")</code>	用命令_____可以简便生成 q1、q2、q3、q4 和 q5

#### 4.10.2 删除（丢弃）变量（列）

<pre>myvars &lt;- names(leadershi p) %i n% c("q3", "q4") newdata &lt;- leadershi p[!myvars]</pre>	<p>删除 leadership 数据框中的 q3 和 q4 变量：</p> <p>生成包含 leadership 列名称的向量，将匹配 q3 和 q4 的列名称定义为 TRUE，不匹配的定义为 FALSE，保存在 myvars 中；</p> <p>删除 q3 和 q4 变量</p>
<pre>newdata &lt;- leadershi p[c(-8, -9)]</pre>	<p>已知 q3 和 q4 是 leadership 数据框中的第 8 个和第 9 个变量</p> <p>删除这两个变量</p>
<pre>leadershi p\$q3 &lt;- leadershi p\$q4 &lt;- NULL</pre>	使用“未定义”删除 leadership 数据框中的 q3 和 q4 变量

#### 4.10.3 选入观测（行）

<pre>newdata &lt;- leadershi p[1:3, ]</pre>	选入 leadership 数据框中的第 1 行至第 3 行观测，保存在 newdata 中
<pre>newdata &lt;- leadershi p[leadershi p\$gender =="M" &amp; leadershi p\$age &gt; 30, ]<sup>36</sup></pre>	选入 leadership 数据框中 gender（性别）为 M，age（年龄）大于 30 的观测，保存在 newdata 中
<pre>attach(leadershi p) newdata &lt;- leadershi p[gender=='M' &amp; age &gt; 30, ] detach(leadershi p)</pre>	<p>绑定 leadership 数据框；</p> <p>选入 gender（性别）为 M，age（年龄）大于 30 的观测，保存在 newdata 中；</p> <p>解除绑定</p>
<pre>leadershi p\$date &lt;- as.Date(leadershi p\$d ate, "%m/%d/%y") startdate &lt;- as.Date("2009-01-01") enddate &lt;- as.Date("2009-10-31") newdata &lt;- leadershi p[leadershi p\$date &gt;= startdate &amp; leadershi p\$date &lt;= enddat e, ]</pre>	<p>已知 leadership 数据框的 date（日期）变量的格式是 mm/dd/yy（诸如 10/28/08）</p> <p>选入在 2009 年 1 月 1 日到 2009 年 12 月 31 日之间的观测，保存在 newdata 中</p> <ol style="list-style-type: none"> <li>1）将 date（日期）变量转换为日期格式</li> <li>2）定义起始日期</li> <li>3）定义结束日期</li> <li>4）获取符合条件的观测</li> </ol>

<sup>36</sup> 该命令来自原书第二版，第一版采用的命令为 `newdata <- leadershi p[whi ch(leadershi p$gender=="M" & leadershi p$age > 30), ]`，这里省去 `whi ch( )` 也能达到相同的结果。

#### 4.10.4 subset()函数

<code>newdata &lt;- subset(leadership, age &gt;= 35   age &lt; 24, select=c(q1, q2, q3, q4))</code>	用 subset() 选取 leadership 数据框中 age 值大于等于 35 或 age 值小于 24 的观测，保留 q1, q2, q3, q4 变量，保存在 newdata 中
<code>newdata &lt;- subset(leadership, gender=="M" &amp; age &gt; 25, select=gender: q4)</code>	用 subset() 选取 leadership 数据框中 age 值大于 25 且 gender 为 M 的观测，保留 gender, q4 及其之间的变量，保存在 newdata 中

#### 4.10.5 随机抽样

<code>mysample &lt;- leadership[sample(1:nrow(leadership), 3, replace=FALSE), ]</code>	从 leadership 数据集中无放回地随机抽取一个大小为 3 的样本，保存在 mysample 中
<code>b=sample(a)</code>	<p>【补】已知：</p> <p><code>a &lt;- c(1,2,3,4,5,6)</code></p> <p>将 a 中的元素随机排序生成向量 b</p>

#### 4.11 使用 SQL 语句操作数据框

## 第 5 章 高级数据管理

### 5.1 一个数据处理难题

### 5.2 数值和字符处理函数

#### 5.2.1 数学函数

<code>abs(x)</code>	绝对值
<code>sqrt(x)</code>	平方根
<code>ceiling(x)</code>	不小于 $x$ 的最小整数
<code>floor(x)</code>	不大于 $x$ 的最大整数
<code>trunc(x)</code>	向 0 的方向截取的 $x$ 中的整数部分
<code>round(x, digits=n)</code>	将 $x$ 舍入为指定位的小数
<code>signif(x, digits=n)</code>	将 $x$ 舍入为指定的有效数字位数
<code>cos(x)</code> 、 <code>sin(x)</code> 、 <code>tan(x)</code>	余弦、正弦和正切
<code>acos(x)</code> 、 <code>asin(x)</code> 、 <code>atan(x)</code>	反余弦、反正弦和反正切
<code>cosh(x)</code> 、 <code>sinh(x)</code> 、 <code>tanh(x)</code>	双曲余弦、双曲正弦和双曲正切
<code>acosh(x)</code> 、 <code>asinh(x)</code> 、 <code>atanh(x)</code>	反双曲余弦、反双曲正弦和反双曲正切
<code>log(x, base=n)</code>	对 $x$ 取以 $n$ 为底的对数
<code>log(x)</code>	自然对数
<code>log10(x)</code>	常用对数
<code>exp(x)</code>	指数函数
每一个独立的值 <code>c(2, 4, 5)</code>	当以上函数被应用于数值向量，矩阵或数据框时，它们会作用于_____。  例如， <code>sqrt(c(4, 16, 25))</code> 的返回值为_____

4	<code>abs(-4)</code> 返回值为_____
5 <code>25^(0.5)</code>	<code>sqrt(25)</code> 返回值为_____, 和_____等价
4	<code>ceiling(3.475)</code> 返回值为_____
3	<code>floor(3.475)</code> 返回值为_____
5	<code>trunc(5.99)</code> 返回值为_____
3.48	<code>round(3.475,digits=2)</code> 返回值为_____

3. 5	signif (3.475,digits=2)返回值为_____
- 0. 416	cos (2)返回值为_____
2	acos (-0.416)返回值为_____
3. 627	sinh (2)返回值为_____
2	asinh (3.627)返回值为_____
2. 3026	log (10)返回值为_____
1	log10 (10)返回值为_____
10	exp (2.3026)返回值为_____

### 5.2.2 统计函数

mean(x)	平均数
mean(x, trim= )	截尾平均数
medi an(x)	中位数
sd(x)	标准差
var(x)	方差
mad(x)	绝对中位差 (median absolute deviation) <sup>37</sup>
quantile(x, probs) 其中 x 为待求分位数的数值型向量, probs 为一个由 [0, 1]之间的概率值组成的数值向量	求分位数
range(x)	求值域
sum(x)	求和
di ff(x, lag=n) lag 用以指定滞后几项。默认的 lag 值为 1	滞后差分
mi n(x)	求最小值
max(x)	求最大值
scale(x, center=TRUE, scale=TRUE)	为数据对象 x 按列进行中心化 <sup>38</sup> (_____)或标准化 (_____) <sup>39</sup>

<sup>37</sup> 绝对中位差:用原始数据减去中位数后得到的新数据的绝对值的中位数。在 R 中, mad() 返回的是标准差的估计=1.4826\*绝对中位差。

<sup>38</sup> 中心化, 例如令 x=c(1,2,3,4,5), 现对 x 进行中心化 scale(x, center=TRUE, scale=FALSE), 即 x 中各值减去均值, 可得到:

```
[1]
[1,] -2
[2,] -1
[3,] 0
[4,] 1
[5,] 2
```

<sup>39</sup> 中心化, 例如令 x=c(1,2,3,4,5), 现对 x 进行标准化 scale(x, center=TRUE, scale=TRUE), 即 x 中各值减去均值再除以标准差, 可得到:

```
[1]
[1,] -1.2649111
[2,] -0.6324555
[3,] 0.0000000
```

中心化(center=TRUE) 标准化(center=TRUE, scale=TRUE)	
--	--

2.5	mean(c(1,2,3,4))返回值为_____
z <- mean(x, trim = 0.05, na.rm=TRUE)	已知: x<- c(1,2,3,4)  求 x 的截尾平均数, 舍弃最大 5%的数和最小 5%的数, 并且不纳入缺失值, 保存到 z
2.5	median(c(1,2,3,4))返回值为_____
1.29	sd(c(1,2,3,4))返回值为_____
1.67	var(c(1,2,3,4))返回值为_____
1.48	mad(c(1,2,3,4))返回值为_____
y <- quantile(x, c(.3, .84))	求 x 的 30%和 84%分位点
c(1, 4) 3	已知: x <- c(1,2,3,4)  range(x)返回值为_____ diff(range(x))返回值为_____
10	sum(c(1,2,3,4))返回值为_____
c(4, 18, 6)	已知: x<- c(1, 5, 23, 29)  diff(x)返回值为_____
1	min(c(1,2,3,4))返回值
4	max(c(1,2,3,4))返回值

数据的标准化:

scale( )	函数_____对矩阵或数据框的所有列 <sup>40</sup> 进行均值为 0、标准差为
----------	---

[4,] 0.6324555

[5,] 1.2649111

<sup>40</sup> 原文这里用的“指定列”不妥, 若 mydata 是一个多变量数据框, scale(mydata)会对 mydata 的所有列(变量)进行均值为 0, 标准差为 1 的变换。

	1 的标准化
<code>newdata &lt;- scale(mydata)*SD + M</code>	若想自定义均值为 M，标准差为 SD 的标准化，则用函数_____
<code>newdata &lt;- transform(mydata, myvar = scale(myvar)*SD + M)</code> <sup>41</sup>	若想对数据框 mydata 中的指定列（列名为 myvar）进行均值为 M，标准差为 SD 的标准化，则用命令_____

### 5.2.3 概率函数

<p>[dpqr] 概率分布函数缩写()</p> <p>d=</p> <p>p=</p> <p>q=</p> <p>r=</p>	<p>在 R 中，概率函数形如_____</p> <p>其中第一个字母表示其所指分布的某一方面：</p> <p>密度函数=</p> <p>分布函数=</p> <p>分位数函数=</p> <p>生成随机数（随机偏差）=</p>
--	--

常用的概率分布函数及其缩写

beta	Beta 分布
binom	二项分布
cauchy	柯西分布
chisq	（非中心）卡方分布
exp	指数分布
f	F 分布
gamma	Gamma 分布
geom	几何分布
hyper	超几何分布
lnorm	对数正态分布
logis	Logistic 分布
multinom	多项分布
nbinom	负二项分布
norm	正态分布
pois	泊松分布
signrank	Wilcoxon 符号秩分布
t	t 分布

<sup>41</sup> 在 4.2 节创建新变量中曾学习过 `transform()` 的示例。

<b>uni f</b>	均匀分布
<b>wei bull</b>	Weibull 分布
<b>wil cox</b>	Wilcoxon 秩和分布

<b>dnorm</b>	指定标准正态分布的密度函数
<b>pnorm</b>	指定标准正态分布的分布函数
<b>qnorm</b>	指定标准正态分布的分位数函数
<b>rnorm</b>	标准正态分布的随机数生成函数
<pre>x &lt;- pretty(c(-3, 3), 30) y &lt;- dnorm(x) plot(x, y, type = "l", yaxs = "i")</pre>	<p>在区间[-3, 3]上绘制标准正态曲线：</p> <p>将区间[-3, 3]等分成 30 个子区间，即得到 31 个在区间内等差的项，保存在 x 中；</p> <p>生成 x 的标准正态分布密度函数值，保存在 y 中；</p> <p>绘制展现 x 与 y 关系的折线图，类型为 l，令纵轴的 0 刻度落在与 x 轴交汇处<sup>42</sup>；</p>
<pre>pnorm(1.96)</pre> <p>等于 0.975</p>	位于 z=1.96 左侧的标准正态曲线下方面积是多少？ <sup>43</sup>
<pre>qnorm(0.9, mean=500, sd=100)</pre> <p>等于 628.16</p>	均值为 500，标准差为 100 的正态分布的 0.9 分位点值为多少？
<pre>rnorm(50, mean=50, sd=10)</pre>	生成 50 个均值为 50，标准差为 10 的正态随机数

设定随机数种子

<b>runi f( )</b>	函数_____可以用来生成 0 到 1 区间上服从均匀分布的伪随机数
<b>set. seed( )</b>	通过函数_____设定随机数种子，让结果可以重现 <sup>44</sup>

生成多元正太数据

<p>MASS 包中的 <b>mvnorm( )</b> 函数</p> <pre>mvnorm(n, mean, sigma)</pre>	<p>_____包中的_____函数可以根据给定均值向量和协方差阵获得多元正态分布的数据。</p> <p>其语法格式为_____</p>
	<p>生成三元正态分布的 500 个观测：</p> <p>已知：</p>

<sup>42</sup> 在 plot( ) 绘制横轴和纵轴的刻度时，为了保证刻度的起点不落在图形的边缘，会默认在刻度起始值与原点（或横轴和纵轴的交汇处）之间留出一小段空白（占一单位刻度的 6%），如果定义了 yaxs = "i" 或 xaxs = "i"，则会取消这一小段空白。

<sup>43</sup> pnorm(x) 求的是位于 x 左侧标准正态分布之下的面积。反之，若已知该面积为 y，用 qnorm(y) 可求得得到该面积的 x 分位数。例如，pnorm(1.96) 得到 0.9750021；qnorm(0.9750021) 得到 1.96。

<sup>44</sup> 要留意设定随机数种子时，数字间不添加逗号。例如，set.seed(1234)，1234 代表一个整数。



<pre>library(MASS) options(digits=3) set.seed(1234) mydata &lt;- mvrnorm(500, mean, sigma) mydata &lt;- as.data.frame(mydata) names(mydata) &lt;- c("y", "x1", "x2") dim(mydata); head(mydata, n=10)</pre>	<p>均值向量为 <code>mean &lt;- c(230.7, 146.7, 3.6)</code></p> <p>协方差阵为 <code>sigma &lt;- matrix(c(15360.8, 6721.2, -47.1, 6721.2, 4700.9, -16.5, -47.1, -16.5, 0.3), nrow=3, ncol=3)</code></p> <p>生成三元正态分布的 500 个观测</p> <p>载入包；</p> <p>设定 3 位有效数字<sup>45</sup>；</p> <p>设定随机数种子 1234；</p> <p>生成三元正态分布的 500 个观测，保存在 <code>mydata</code> 中；</p> <p>将 <code>mydata</code> 转换为数据框；</p> <p>将数据框的三列分别命名为 <code>y</code>, <code>x1</code>, <code>x2</code>；</p> <p>查看 <code>mydata</code> 的维度；</p> <p>查看 <code>mydata</code> 的前 10 行</p>
--	---

## 5.2.4 字符处理函数

<code>nchar(x)</code>	计算 <code>x</code> 中的字符数量
<code>substr(x, start, stop)</code>	提取或替换字符向量 <code>x</code> 中的子串
<code>grep(pattern, x, ignore.case=FALSE, fixed=FALSE)</code>	在 <code>x</code> 中搜索某种模式。若 <code>fixed=FALSE</code> ，则 <code>pattern</code> 为一个正则表达式 <sup>46</sup> 。若 <code>fixed=TRUE</code> ，则 <code>pattern</code> 为一个文本字符串。返回值为匹配的下标。
<code>sub(pattern, replacement, x, ignore.case=FALSE, fixed=FALSE)</code>	在 <code>x</code> 中搜索 <code>pattern</code> ，并以文本 <code>replacement</code> 将其替换。若 <code>fixed=FALSE</code> ，则 <code>pattern</code> 为一个正则表达式。若 <code>fixed=TRUE</code> ，则 <code>pattern</code> 为一个文本字符串。
<code>strsplit(x, split, fixed=FALSE)</code>	在 <code>split</code> 处分割字符向量 <code>x</code> 中的元素。若 <code>fixed=FALSE</code> ，则 <code>split</code> <sup>47</sup> 为一个正则表达式。若 <code>fixed=TRUE</code> ，则 <code>split</code> 为一个文本字符串。
<code>paste(..., sep=" ")</code>	连接字符串，分隔符为 <code>sep</code>
<code>toupper(x)</code>	大写转换

<sup>45</sup> 我们可以发现在 `head(mydata, n=10)` 中有些数字并不是保留了 3 位有效数字，原因是 `options(digits=3)` 设定后，会对 `y`, `x1`, `x2` 每一列都设定 3 位有效数字，优先保证该列整数数位最小的数字保留 3 位有效数字，其他数字则需要和其保留相同的小数位数。例如，`a<-4562.6547954`, `b<-1.2456`, `c<-0.00045789`，设定 `options(digits=4)`，如果我们分开看 `a`, `b`, `c` 的结果会得到 4563, 1.246, 0.0004579，都是 4 个有效数字，但如果我们把它组合在一起呈现，`c(a,b,c)` 会得到 4.563e+03 1.246e+00 4.579e-04。另外，`options(digits=)` 中，`digits` 默认为 7，可选值为 1 至 22。

另外，如果要进行四舍五入到 `n` 位小数的话，使用 `round(x,digit=n)`。可参考第 8 章开头关于回归结果整理的脚注。

<sup>46</sup> 正则表达式涉及的内容很多，可以参考其他资料。

<sup>47</sup> 这里原书第一版和第二版都写的是“`pattern`”，应该更正为“`split`”。

<code>tolower(x)</code>	小写转换
<code>length(x)</code> 返回值为 3 <code>nchar(x[3])</code> 返回值为 5	已知: <code>x &lt;- c("ab", "cde", "fghij")</code>  求 x 中含有几个元素, 其返回值为_____ 求 x 的第三个元素有几个字符, 其返回值为_____
<code>substr(x, 2, 4)</code> 返回值为"bcd" <code>substr(x, 2, 4) &lt;- "222"</code> <sup>48</sup> x 将变成"a222ef"	已知: <code>x &lt;- "abcdef"</code>  提取 x 中第 2 个至第 4 个字母, 其返回值为_____; 将 x 中第 2 个至第 4 个字母替换为 222, 其返回值为_____
<code>grep("A", c("b", "A", "c"), fixed=TRUE)</code> 返回值为 2	已知: <code>c("b", "A", "c")</code>  使用文本字符串搜索"A"以确定其所在的位置, 其返回值为_____
<code>sub("\\s", ". ", "Hello There")</code> 返回值为 Hel l o. There。  使用"\\s"而不用" \s "	已知: <code>"Hello There"</code>  用正则表达式搜索其中的空格, 用"."替换空格, 其返回值为_____ (在 R 中使用正则表达式_____表达空格, 而不是_____)
<code>y &lt;- strsplit("abc", "")</code> 返回一个含有 1 个成分、3 个元素的列表, 包含的内容为"a" "b" "c"  <code>unlist(y)[2]</code> <code>sapply(y, "[", 2)</code>	已知: <code>"abc"</code>  将"abc"拆分成单独的字母, 保存在 y 中, 其返回值为_____ 使用命令_____或_____均可返回"b"
<code>paste("x", 1:3, sep="")</code> <code>paste("x", 1:3, sep="M")</code> <code>paste("Today is", date())</code>	采用拼接的方法将"x"转换为 <code>c("x1", "x2", "x3")</code> 采用拼接的方法将"x"转换为 <code>c("xM1", "xM2", "xM3")</code>

<sup>48</sup> 原文这里是 `substr(x, 2, 4) <- "22222"`, 实际上三个 2 就够了, 原文肯能的意图在于提醒我们如果在替换时赋值符号 (<-) 之后的字符个数超出了需要替换的字符个数, 则只取开始的部分。例如已知 `x <- "abcdef"`, `substr(x, 2, 4) <- "123456789"`, x 变为"a123ef"。而若赋值符号 (<-) 之后的字符个数小于需要替换的字符个数, 则把能替换的部分都替换了。例如例如已知 `x <- "abcdef"`, `substr(x, 2, 4) <- "12"`, x 变为"a12def"。

	采用拼接的方法将"Today is"之后加上当日日期
<code>toupper("abc")</code>	将"abc"转换为"ABC"
<code>tolower("ABC")</code>	将"ABC"转换为"abc"

### 5.2.5 其他实用函数

<code>length(x)</code>	对象 <code>x</code> 的长度
<code>seq(from, to, by)</code>	生成一个序列
<code>rep(x, n)</code>	将 <code>x</code> 重复 <code>n</code> 次
<code>cut(x, n)</code> <code>ordered_result = TRUE</code>	将连续型变量 <code>x</code> 分割为有着 <code>n</code> 个水平的因子。参数_____可以创建一个有序型因子
<code>pretty(x, n)</code>	创建美观的分割点。通过选取 <code>n+1</code> 个等间距的取整值，将一个连续型变量 <code>x</code> 分割为 <code>n</code> 个区间。绘图中常用。 <sup>49</sup>
<code>cat(... , file = "myfile", append = FALSE)</code> 50	连接...中的对象，并将其输出到屏幕上或文件中（如果声明了一个的话）

<code>length(x)</code> 返回值为 4	已知：  <code>x &lt;- c(2, 5, 6, 9)</code>  求 <code>x</code> 中元素的个数，其返回值为_____
<code>c(1, 3, 5, 7, 9)</code>	已知：  <code>indices &lt;- seq(1,10,2)</code>  <code>indices</code> 的值为_____
<code>c(1, 2, 3, 1, 2, 3)</code>	已知：  <code>y &lt;- rep(1:3, 2)</code>  <code>y</code> 的值为_____
<code>[1] (-1, -0.333] (-0.333, 0.333] (0.333, 1]</code> <code>Levels: (-1, -0.333] (-0.333, 0.333] (0.333, 1]</code>	【补】已知：  <code>x &lt;- -1:1</code>

<sup>49</sup> 例如在 5.2.3 节表 5-5 中用 `x <- pretty(c(-3,3), 30)` 将 `(-3, 3)` 的区间等分了 30 个区间。

<sup>50</sup> `append` 是一个逻辑参数，只在 `file` 是一个文件名时使用。如果 `append=TRUE`，连接后的文本将添加在 `file` 中，如果 `append=FALSE`，连接后的文本会覆盖 `file` 中的原文本。

<p>其中, <code>(-1, -0.333]</code> 是 <code>-1</code> 落入的区间, <code>(-0.333, 0.333]</code> 是 <code>0</code> 落入的区间, <code>(0.333, 1]</code> 是 <code>1</code> 落入的区间。</p> <p><code>table(y)</code></p>	<p><code>y &lt;- cut(x,3)</code></p> <p><code>y</code> 返回什么? 如何解读结果?</p> <p>如何查看每个区间落入了几个元素?</p>
<p><code>\n</code> 为换行;</p> <p><code>\t</code> 为制表符;<sup>51</sup></p> <p><code>\'</code> 为单引号;</p> <p><code>\b</code> 为退格<sup>52</sup></p>	<p>用转义字符表示:</p> <p>换行;</p> <p>制表符;</p> <p>单引号;</p> <p>退格</p>
<p><code>cat("Hello", first_name, "\n")</code></p>	<p>已知:</p> <p><code>first_name &lt;- c("Jane")</code></p> <p>将"Jane"添加在"Hello"之后, 添加结束后取新行。</p>
<p><code>cat("Hello", name, "\b.\n", "Isn't R", "\t", "GREAT?\n")</code></p>	<p>已知:</p> <p><code>name &lt;- "Bob"</code></p> <p>生成如下字样:</p> <p>Hello Bob.</p> <p>Isn't R      GREAT?</p> <p>(请注意第二行缩进了一个空格。)</p>
<p>空格</p>	<p>当 <code>cat</code> 输出连接后的对象时, 它会将每一个对象都_____用分开。</p>
<p><code>cat("Hello", name, "\b.\n", "\bIsn't R", "\t", "GREAT?\n")</code></p>	<p>【补】已知:</p> <p><code>name &lt;- "Bob"</code></p> <p>生成如下字样:</p> <p>Hello Bob.</p> <p>Isn't R      GREAT?</p> <p>(请注意第二行不再缩进一个空格。)</p>
<p>Hello Haoyu Mingming</p>	<p>【补】已知:</p>

<sup>51</sup> 相当于键盘上的“Tab”键的效果。

<sup>52</sup> 相当于键盘上的“Backspace”或“←”或“←Backspace”键的效果。

[1] "Hello Haoyu" "Hello Mingming"	<pre>a&lt;- c("Hello") b=c("Haoyu","Mingming")</pre> <p>cat(a,b) 的返回结果是_____</p> <p>paste(a,b)的返回结果是_____</p>
------------------------------------	---

### 5.2.6 将函数应用于矩阵和数据框

sqrt(a)	<p>已知:</p> <pre>a &lt;- 5</pre> <p>求 a 的平方根</p>
round(b)	<p>已知:</p> <pre>b &lt;- c(1.243, 5.654, 2.99)</pre> <p>将 b 中的元素近似至个位</p>
c <- matrix(runif(12), nrow=3)	生成 3 行 4 列的矩阵 c，元素为 12 个均匀分布的数
log(c)	求 c 中的各元素的对数
mean(c)	求矩阵中全部 12 个元素的均值

<pre>apply( ) apply(x, MARGIN, FUN, ...)</pre> <p>MARGIN=1</p> <p>MARGIN=2</p>	<p>_____函数可将一个任意函数“应用”到矩阵、数组、数据框的任何维度上，其语法格式为_____；</p> <p>_____参数表示将函数应用到行；</p> <p>_____参数表示将函数应用到列</p>
<pre>apply(mydata, 1, mean) apply(mydata, 2, mean) apply(mydata, 2, mean, trim=0.2)</pre>	<p>已知:</p> <pre>mydata &lt;- matrix(rnorm(30), nrow=6)</pre> <p>计算 mydata 每行的均值；</p> <p>计算 mydata 每列的均值；</p> <p>计算 mydata 每列的截尾均值（最高和最低 20%的值被忽略）</p>

## 5.3 数据处理难题的一套解决方案

roster <- data.frame(Student, Math, Science, English, stringsAsFactors=FALSE)	已知:
---	-----

	<pre>options(digits=2)  Student &lt;- c("John Davis", "Angela Williams", "Bullwinkle Moose",              "David Jones", "Janice Markhammer", "Cheryl Cushing",              "Reuven Ytzhak", "Greg Knox", "Joel England",              "Mary Rayburn")  Math &lt;- c(502, 600, 412, 358, 495, 512, 410, 625, 573, 522)  Science &lt;- c(95, 99, 80, 82, 75, 85, 80, 95, 89, 86)  English &lt;- c(25, 22, 18, 15, 20, 28, 15, 30, 27, 18)</pre> <p>用以上变量依次组成数据框 <code>roster</code>（不包含因子型数据）</p>
<pre>z &lt;- scale(roster[, 2:4])</pre>	<p>由于 <code>Math</code>, <code>Science</code>, <code>English</code> 的数据不具有可比性，因此将这三个变量（第 2 列至第 4 列）进行均值为 0，标准差为 1 的标准化，将标准化后的数据保存在 <code>z</code> 中</p>
<pre>score &lt;- apply(z, 1, mean)</pre>	<p>计算 <code>z</code> 中每一行的均值，保存在 <code>score</code> 中</p>
<pre>roster &lt;- cbind(roster, score)</pre>	<p>把 <code>score</code> 当做新变量（列）添加在 <code>roster</code> 数据框的最右侧</p>
<pre>y &lt;- quantile(score, c(.8, .6, .4, .2))</pre> <p>y 的返回值为：</p> <pre>80%  60%   40%   20% 0.74 0.44 -0.36 -0.89</pre>	<p>找出 <code>score</code> 中 80%，60%，40%，20% 的分位点，保存在 <code>y</code> 中</p>
<pre>roster\$grade[score &gt;= y[1]] &lt;- "A" roster\$grade[score &lt; y[1] &amp; score &gt;= y[2]] &lt;- "B" roster\$grade[score &lt; y[2] &amp; score &gt;= y[3]] &lt;- "C" roster\$grade[score &lt; y[3] &amp; score &gt;= y[4]] &lt;- "D" roster\$grade[score &lt; y[4]] &lt;- "F"</pre>	<p>生成新变量 <code>grade</code>，如果 <code>score</code> 的值位于 80%-100% 分位则赋值 A，位于 60%-80% 分位则赋值 B，位于 40%-60% 分位则赋值 C，位于 20%-40% 分位则赋值 D，位于 0-20% 分位则赋值 F</p>
<pre>name &lt;- strsplit(roster\$Student, " ")<sup>53</sup></pre>	<p>把变量 <code>student</code> 中学生的名字以空格拆分为姓氏和名字，保存在 <code>name</code> 中</p>
<pre>Firstname &lt;- sapply(name, "[", 1)</pre>	<p>提取列表 <code>name</code> 中每一个成分的第一个元素，保存在 <code>Firstname</code> 中</p>
<pre>Lastname &lt;- sapply(name, "[", 2)</pre>	<p>提取列表 <code>name</code> 中每一个成分的第一个元素，保存在 <code>Lastname</code> 中</p>
<pre>roster &lt;- cbind(Firstname, Lastname, roster[, -1])</pre>	<p>把 <code>Firstname</code> 和 <code>Lastname</code> 作为第一列和第二列放入 <code>roster</code> 中，删除原来的变量 <code>Student</code>（第一列）</p>

<sup>53</sup> 原书第一版和第二版此处的命令为 `name <- strsplit((roster$Student), " ")`，但 `roster$Student` 外侧的括号多余。

<code>roster &lt;- roster[order(Lastname, Firstname), ]</code>	把 roster 按 Lastname 排序, 如果 Lastname 一致的话按 Firstname 排序
--	--

## 5.4 控制流

一条单独的 <b>R</b> 语句或一组复合语句（包含在花括号{ } 中的一组 <b>R</b> 语句, 使用分号分隔）	语句（statement）是_____
一条最终被解析为真（ <b>TRUE</b> ）或假（ <b>FALSE</b> ）的表达式	条件（cond）是_____
一条数值或字符串的求值语句	表达式（expr）是_____
一个数值或字符串序列	序列（seq）是_____

### 5.4.1 重复和循环

for 循环结构

<b>for</b> 循环重复地执行一个语句, 直到某个变量的值不再包含在序列 <b>seq</b> 中为止。	<b>for</b> 循环结构的执行逻辑为_____
<b>for</b> ( <b>var in seq</b> ) <b>statement</b> <b>var</b> 是变量 <b>seq</b> 是变量的取值范围 <b>statement</b> 是执行的语句	<b>for</b> 循环结构的语法结构为_____
<code>for (i in 1:10) print("Hello")</code>	用 <b>for</b> 循环把 Hello 重复输出 10 次
<code>for (i in 1:10) print(i)</code>	【补】用 <b>for</b> 循环输出数字 1 至 10
10 个 <b>i</b>	【补】 <code>for (i in 1:10) print("i")</code> 输出什么?

while 结构

<b>while</b> 循环重复地执行一个语句, 直到条件不为真为止。	<b>while</b> 循环结构的执行逻辑为_____
<b>while</b> ( <b>cond</b> ) <b>statement</b> <b>cond</b> 为条件语句 <b>statement</b> 是执行的语句	<b>while</b> 循环结构的语法结构为_____
<code>i &lt;- 10</code> <code>while (i &gt; 0) {print("Hello"); i &lt;- i - 1}</code>	用 <b>while</b> 循环把 Hello 重复输出 10 次

### 5.4.1 条件执行

if-else 结构

控制结构 <b>if-else</b> 在某个给定条件为真时执行语句。也可以同时在条件为假时执行另外的语句。	<b>if-else</b> 结构的执行逻辑为_____
<b>if</b> ( <b>cond</b> ) <b>statement</b> <b>if</b> ( <b>cond</b> ) <b>statement1</b> <b>else</b> <b>statement2</b>	<b>if-else</b> 结构的语法结构为_____
<code>if (is.character(grade)) grade &lt;- as.factor(grade)</code>	用 <b>if-else</b> 结构实现: 如果 <b>grade</b> 是一个字符向量, 就把它转换为一个因子
<code>if (!is.factor(grade)) grade &lt;- as.factor(grade) else print("Grade already is a factor")</code>	用 <b>if-else</b> 结构实现: 如果 <b>grade</b> 不是一个因子, 就把它转换为一个因子

factor")	因子，否则就输出 Grade already is a factor
<p>if 结构只输入 1 次 Hello</p> <p>while 结构输出 10 次 Hello</p>	<p>【补】</p> <pre>i &lt;- 10 if (i&gt;0) {print("Hello"); i &lt;- i-1}</pre> <p>会输出什么？</p> <p>和 while (i &gt; 0) {print("Hello"); i &lt;- i-1}的结果有什么不同？<sup>54</sup></p>

#### ifelse 结构

<p>ifelse(cond, statement1, statement2)</p> <p>若 cond 为 TRUE，则执行第一个语句；若 cond 为 FALSE，则执行第二个语句</p>	<p>ifelse 结构是 if-else 结构比较紧凑的向量化版本，其语法结构为</p> <p>_____</p>
<pre>outcome &lt;- ifelse (score &gt; 0.5, "Passed", "Failed")</pre>	<p>已知：</p> <pre>score=c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1)</pre> <p>用 ifelse 结构实现：如果 score 中的值大于 0.5 则输出 Passed，否则输出 Failed，将结果保存在向量 outcome 中</p>
ifelse	<p>在程序的行为是二元时，或者希望结构的输入和输出均为向量时，建议使用_____</p>

#### switch 结构

一个表达式的值	switch 根据_____选择语句执行
<pre>for (i in feelings)   print(     switch(i,       happy = "I am glad you are happy",       afraid = "There is nothing to fear",       sad = "Cheer up",       angry = "Calm down now"     )   )</pre>	<p>已知：</p> <pre>feelings &lt;- c("sad", "afraid")</pre> <p>用 for 循环结构和 switch 结构实现：如果 feelings 中的元素 i 是 Happy，就显示出 I am glad you are happy；元素 i 是 afraid，就显示出 There is nothing to fear；如果 i 是 sad，就显示出 Cheer up；如果元素 i 是 angry，就显示出 Calm down now</p>

## 5.5 自编函数

<pre>myfunction &lt;- function(arg1, arg2, ... ) {   statements   return(object) }</pre> <sup>55</sup>	<p>自编函数的语法结构为_____（令函数名为 myfunction）</p> <p>解释 myfunction 内部参数的执行逻辑</p>
--	---

<sup>54</sup> 注意区别 if 和 while 结构的输出结果有什么不同。



<p><b>myfunction</b> 可自定义的函数名  <b>function()</b> 是固定的构建函数的命令  <b>arg1, arg2, ...</b> 是函数中的变量, 诸如 <b>x, y, z</b>  <b>statements</b> 是函数体, 用于构建变量间的关系  <b>return(object)</b> 是指函数返回 (展现) 的结果            至于 <b>arg1, arg2, ...</b> 的具体取值, 需要在函数语法结构之外定义, 再执行 <b>myfunction(x, y, z)</b></p>	
<p>① 函数名为 <b>mystats</b> 意味着在之后需要调用该函数时            需要使用 <b>mystats(变量名)</b> 的形式。函数中的参数 (即            变量) 为 <b>x, parametric, print</b>, 并且赋予后两个            参数默认值为 <b>TRUE</b> 和 <b>FALSE</b>, 意味着在之后调用该函            数时, 如果不给出 <b>parametric, print</b> 参数, 将默            认其为 <b>TRUE</b> 和 <b>FALSE</b>。</p> <p>② <b>if-else</b> 结构, 如果 <b>parametric</b> 为 <b>TRUE</b> (即默            认情况下), 则定义 <b>center</b> 为 <b>x</b> 的均值, <b>spread</b> 为  <b>x</b> 的标准差, 否则 (即 <b>parametric</b> 为 <b>FALSE</b>, 非默            认值) 定义 <b>center</b> 为 <b>x</b> 的中位数, <b>spread</b> 为 <b>x</b> 的中            位差。</p> <p>③ <b>if-else</b> 结构, 如果 <b>print</b> 为 <b>TRUE</b> (非默认) 且  <b>parametric</b> 为 <b>TRUE</b> (默认), 则用 <b>cat</b> 链接对象 (平            均数和标准差) ("<b>\n</b>" 表示新行), 否则, 如果 <b>prin</b>  <b>t</b> 为 <b>TRUE</b> (非默认) 且 <b>parametric</b> 为 <b>FALSE</b> (非默            认), 则用 <b>cat</b> 链接对象 (中位数和中位差) ("<b>\n</b>"            表示新行)</p> <p>④ 定义 <b>result</b> 为包含 <b>center</b> 和 <b>spread</b> 的列表,            并定义 <b>center=center, spread=spread</b></p> <p>⑤ 函数返回 <b>result</b></p>	<p>说明下例自编函数的执行逻辑 (分别对命令 ① 至 ⑤ 进行阐述):</p> <pre> mystats &lt;- function(x, parametric=TRUE, print=FALSE) {      ①    if (parametric) {   ②      center &lt;- mean(x); spread &lt;- sd(x)    } else {      center &lt;- median(x); spread &lt;- mad(x)    }    if (print &amp; parametric) {                                     ③      cat("Mean=", center, "\n", "SD=", spread, "\n")    } else if (print &amp; !parametric) {      cat("Median=", center, "\n", "MAD=", spread, "\n")    }    result &lt;- list(center=center, spread=spread)                 ④    return(result)  ⑤  }</pre>
<p><b>y</b> 将是一个包含了均值何标准差的列表, 但并没有显示            出输出结果。</p> <p>这是因为 <b>y &lt;- mystats(x)</b> 默认 <b>parametric=TRUE</b>,  <b>print=FALSE</b>, 因此在 ② 中 <b>if-else</b> 结构中将会            执行 <b>else</b> 之前的命令, 而在 ③ 中 <b>if-else</b> 结构因            为 <b>print=FALSE</b> 而根本不会执行, 也就没有输出结            果。</p>	<p>接上例</p> <p>已知:</p> <pre> set.seed(1234)  x &lt;- rnorm(500)</pre> <p>执行语句:</p> <p><b>y &lt;- mystats(x)</b> 会得到什么? <b>mystats()</b> 是怎样执行 <b>x</b> 的? (<b>mystats</b>            为上面定义的自编函数)</p>

<sup>55</sup> 一种比较“直观粗暴”的理解: “function” 相当于 **f(x)** 中的 “f”, 只不过这里必须写成 “function”; **arg1, arg2** 可理解为 **f(x, y)** 中的 **x, y**; **statement** 可以理解为 **f(x, y)=(x+y)^2** 中的 **(x+y)^2** 或者其他各种映射关系; 而 **return** 则给了我们更灵活的方式输入想要的结果。

在 ② 中 <code>if-else</code> 结构中将会执行 <code>else</code> 之后的命令，而在 ③ 中 <code>if-else</code> 结构将会执行 <code>else</code> 之后的命令	<p>【补】接上例</p> <p>如果执行语句：</p> <pre>y &lt;- mystats(x, parametric=FALSE, print=TRUE)</pre> <p>会得到什么？</p> <p><code>mystats()</code>是怎样执行 <code>x</code> 的？</p>
---	---

<p>① 函数名为 <code>mydate</code>，变量名为 <code>type</code>（默认值为 <code>long</code>）</p> <p>② <code>switch</code> 结构，当 <code>type</code> 为 <code>long</code> 时，用 <code>Sys.time()</code> 返回当天日期（如“2010-12-01”，见 4.6 节），并用 <code>format</code>（可以用 <code>format(x, format="output_format")</code> 来输出指定格式的日期，见 4.6 节）输出“非缩写的星期名（空格）非缩写的月份（空格）数字表示的日期（空格）四位数的年份”。当 <code>type</code> 为 <code>short</code> 时，用 <code>Sys.time()</code> 返回当天日期，并用 <code>format</code> 输出“月份（00~12）- 数字表示的日期- 两位数年份”。当输入的 <code>type</code> 不匹配 <code>long</code> 或 <code>short</code> 时，输出 <code>type</code> “is not a recognized type（换行）”</p>	<p>说明下例自编函数的执行逻辑（分别对命令 ① 和 ② 进行阐述）：</p> <pre>mydate &lt;- function(type="long") {   switch(type,     long = format(Sys.time(), "%A %B %d %Y"),     short = format(Sys.time(), "%m-%d-%y"),     cat(type, "is not a recognized type\n")   ) }</pre> <p>①</p> <p>②</p>
<pre>"Wednesday December 13 2017"<sup>56</sup> "12-13-17" "Wednesday December 13 2017" medium is not a recognized type</pre>	<p>接上例</p> <p>已知：</p> <p>今日的日期为 2017 年 12 月 13 日星期三</p> <p>执行 <code>mydate("long")</code> 会输出_____</p> <p>执行 <code>mydate("short")</code> 会输出_____</p> <p>执行 <code>mydate()</code> 会输出_____</p> <p>执行 <code>mydate("medium")</code> 会输出_____</p>

## 5.6 整合与重构

### 5.6.1 转置

<code>cars &lt;- mtcars[1:5, 1:4]</code>	调用 <code>mtcars</code> 数据框的第 1 至 5 行以及第 1 至 4 列，保存在 <code>cars</code> 中
<code>t(cars)</code>	将 <code>cars</code> 转置

### 5.6.2 整合数据

<code>aggregate(x, by, FUN)</code>	_____函数可以使用一个或多个变量以及一个预先设定好的函数来折叠数据 <sup>57</sup>
<code>x</code> 是待折叠的数据对象， <code>by</code> 是一个变量名组成的列表，这些变量将被去掉以形成新的观测，而 <code>FUN</code> 则是用来计	

<sup>56</sup> 汉化后的输出是“星期三 十二月 13 2017”

<sup>57</sup> 这里的“折叠数据”可以理解为 Excel 中的数据透视表功能。`x` 作为透视表中的列变量，`by` 作为行变量，`FUN` 为函数。

算描述性统计量的标量函数，它将被用来计算新观测中的值。	该函数包含的参数有_____
<pre>options(digits=3) attach(mtcars) aggdata &lt;- aggregate(mtcars, by=list(cyl, gear), FUN=mean, na.rm=TRUE)</pre>	<p>针对 <code>mtcars</code> 数据框，根据变量 <code>cyl</code> 和 <code>gear</code> 整合其他所有变量<sup>58</sup>，返回各变量的均值，在计算均值时排除空值，结果保存在 <code>aggdata</code> 中<sup>59</sup>：</p> <p>设定 3 位有效数字：</p> <p>绑定 <code>mtcars</code> 数据框：</p> <p>整合变量至 <code>aggdata</code> 中</p>
<pre>attach(mtcars) aggregate(mtcars[, c("dis", "hp")], by=list(cyl), FUN=sum)<sup>60</sup></pre>	【补】根据 <code>cyl</code> 整合 <code>mtcars</code> 中的 <code>dis</code> 和 <code>hp</code> 变量，返回各变量的均值。
<code>table(mtcars\$cyl)</code>	【补】求各类 <code>cyl</code> 的计数值

### 5.6.3 reshape 包

reshape 包/reshape2 包 <sup>61</sup>	_____包是一套重构和整合数据集的万能工具
<pre>melt()</pre> <p>唯一的标识符—变量组合</p> <pre>cast()</pre>	大致说来，要首先将数据融合（_____函数），以使每一行都是_____。然后将数据重铸（_____函数）为你想要的任何形状。

<pre>library(reshape) md &lt;- melt(mydata, id=c("ID", "Time"))</pre>	<p>已知：</p> <pre>ID &lt;- c(1,1,2,2) Time &lt;- c(1,2,1,2) X1 &lt;- c(5,3,6,2) X2 &lt;- c(6,5,1,4) mydata &lt;- data.frame(ID,Time,X1,X2)</pre> <p>将 mydata 按 ID 和 Time 融合，将结果保存在 md 中</p> <p>若展示 md 会得到_____，其包括四个变量，分别是_____， _____, _____, _____。</p>
---	---

	ID	Time	variable	value
1	1	1	X1	5
2	1	2	X1	3
3	2	1	X1	6
4	2	2	X1	2
5	1	1	X2	6
6	1	2	X2	5
7	2	1	X2	1
8	2	2	X2	4

ID

Time

variable

value

<sup>58</sup> 若只想用某一个变量作为列变量，可以采用“`mtcars$变量名`”实现。在 6.1.3 节中提供了一个例子。

<sup>59</sup> 若采用 Excel 数据透视表的概念可以理解为把 `cyl` 和 `gear` 作为数据透视表的行变量（其中 `cyl` 作为一级行标签，`gear` 作为二级行标签），`mtcars` 中的其他变量作为透视表中的列变量，函数值采用均值。

<sup>60</sup> 注意这里即便只有一个行变量 `cyl` 也需要采用 `by=list(cyl)` 而不是 `by=cyl`。

<sup>61</sup> 原书第一版中是 `reshape` 包，第二版中是 `reshape2` 包。

<code>cast(md, ID~variable, mean)</code>	<p>(a) 重塑并整合 md，求每个 ID 的 variable 的均值（即 ID 为行标签，variable 是列变量，函数为平均值），得到下表：</p> <table><tr><td></td><td>ID</td><td>X1</td><td>X2</td></tr><tr><td>1</td><td>1</td><td>4</td><td>5.5</td></tr><tr><td>2</td><td>2</td><td>4</td><td>2.5</td></tr></table>		ID	X1	X2	1	1	4	5.5	2	2	4	2.5													
	ID	X1	X2																							
1	1	4	5.5																							
2	2	4	2.5																							
<code>cast(md, Time~variable, mean)</code>	<p>(b) 重塑并整合 md，求在不同 Time 的 variable 的均值（即 Time 为行标签，variable 是列变量，函数为平均值），得到下表：</p> <table><tr><td></td><td>Time</td><td>X1</td><td>X2</td></tr><tr><td>1</td><td>1</td><td>5.5</td><td>3.5</td></tr><tr><td>2</td><td>2</td><td>2.5</td><td>4.5</td></tr></table>		Time	X1	X2	1	1	5.5	3.5	2	2	2.5	4.5													
	Time	X1	X2																							
1	1	5.5	3.5																							
2	2	2.5	4.5																							
<code>cast(md, ID~Time, mean)</code>	<p>(c) 重塑并整合 md，求每个 ID 的 variable 在不同时间的平均值（即 ID 为行标签，Time 为列变量），得到下表：</p> <table><tr><td></td><td>ID</td><td>1</td><td>2</td></tr><tr><td>1</td><td>1</td><td>5.5</td><td>4</td></tr><tr><td>2</td><td>2</td><td>3.5</td><td>3</td></tr></table>		ID	1	2	1	1	5.5	4	2	2	3.5	3													
	ID	1	2																							
1	1	5.5	4																							
2	2	3.5	3																							
<code>cast(md, ID+Time~variable)</code>	<p>(d) 重塑 md，得到每个 ID 在不同 Time 中对应的 variable（X1 和 X2）值（即 ID 为一级行标签，Time 为二级行标签，variable 是列变量），得到下表：</p> <table><tr><td></td><td>ID</td><td>Time</td><td>X1</td><td>X2</td></tr><tr><td>1</td><td>1</td><td>1</td><td>5</td><td>6</td></tr><tr><td>2</td><td>1</td><td>2</td><td>3</td><td>5</td></tr><tr><td>3</td><td>2</td><td>1</td><td>6</td><td>1</td></tr><tr><td>4</td><td>2</td><td>2</td><td>2</td><td>4</td></tr></table>		ID	Time	X1	X2	1	1	1	5	6	2	1	2	3	5	3	2	1	6	1	4	2	2	2	4
	ID	Time	X1	X2																						
1	1	1	5	6																						
2	1	2	3	5																						
3	2	1	6	1																						
4	2	2	2	4																						
<code>cast(md, ID+variable~Time)</code>	<p>(e) 重塑 md，得到每个 ID 在不同 variable 中对应的 Time 值（即 ID 为一级行标签，variable 为二级行标签，Time 是列变量），得到下表：</p> <table><tr><td></td><td>ID</td><td>variable</td><td>1</td><td>2</td></tr><tr><td>1</td><td>1</td><td>X1</td><td>5</td><td>3</td></tr><tr><td>2</td><td>1</td><td>X2</td><td>6</td><td>5</td></tr><tr><td>3</td><td>2</td><td>X1</td><td>6</td><td>2</td></tr></table>		ID	variable	1	2	1	1	X1	5	3	2	1	X2	6	5	3	2	X1	6	2					
	ID	variable	1	2																						
1	1	X1	5	3																						
2	1	X2	6	5																						
3	2	X1	6	2																						

	4	2	X2	1	4
cast(md, ID~variable+Time)	(f) 重塑 md, 得到每个 ID 的 X1 和 X2 在不同 Time 中的值是多少 (即 ID 为行标签, variable 和 Time 是列变量), 得到下表:				
	ID	X1_1	X1_2	X2_1	X2_2
1	1	5	3	6	5
2	2	6	2	1	4

<pre> name year test math english 1    a 2017    1    1      2 2    a 2017    2    5      9 3    a 2017    3    8      8 4    a 2018    1    6      5 5    a 2018    2    4      3 6    a 2018    3    2      2 7    b 2017    1    7      7 8    b 2017    2    8      1 9    b 2017    3    5      4 10   b 2018    1    4      9 11   b 2018    2    6      5 12   b 2018    3    7      7                     </pre>	<p>【补】如果觉得上面的示例的演示不够直观, 可以尝试下面这个例子。</p> <p>已知: a 和 b 参加了在 2017 年和 2018 年分别组织的 3 次考试, 取得了 math 和 english 的成绩。</p> <pre> name &lt;- c("a","a","a","a","a","a","b","b","b","b","b","b") year&lt;-c(2017,2017,2017,2018,2018,2018,2017,2017,2017, 2018,2018,2018) test &lt;- c(1,2,3,1,2,3, 1,2,3,1,2,3) math &lt;- c(1,5,8,6,4,2,7,8,5,4,6,7) english &lt;-c(2,9,8,5,3,2,7,1,4,9,5,7) mydata &lt;- data.frame(name,year,test,math,english)                     </pre>
<pre> library(reshape) md&lt;-melt(mydata, id=(c("name","year","test")))                     </pre>	融合 mydata, 保存在 md 中
cast(md, year~variable, mean)	重塑并整合 mydata, 求每年 math 和 english 的平均分
cast(md, name+year~variable, sum)	重塑并整合 mydata, 求每人每年 math 和 english 的总分
cast(md, name~year+test, sum)	重塑并整合 mydata, 求每人(行标签)在每年每次考试(列标签)中的总分
cast(md, name~year+test+variable)	重塑 mydata, 得到每人(行标签)在每年每次考试(列标签)中的分数
cast(md, year+test~name+variable)	重塑 mydata, 得到每年每次(行标签)每人每次考试(列标签)的分数

## 第 6 章 基本图形

### 6.1 条形图

#### 6.1.1 简单的条形图

<pre>barplot(height) horiz=TRUE main xlab 和 ylab</pre>	<p>绘制条形图的函数是_____</p> <p>参数_____则会生成一幅水平条形图。参数_____可添加一个图形标题,而参数_____和_____则会分别添加 x 轴和 y 轴标签。</p>
<pre>library(vcd) counts &lt;- table(Arthritis\$Improved) barplot(counts, main="Horizontal Bar Plot", xlab="Frequency", ylab="Improvement", horiz=TRUE)</pre>	<p>载入 vcd 包;</p> <p>对 Arthritis 数据集的 Improved 变量做频数统计,保存在 counts 中;针对 counts 绘制水平条形图,图形名称为 Horizontal Bar Plot,横轴名称为 Frequency,纵轴名称为 Improvement,</p>
<pre>plot((Arthritis\$Improved))</pre> <p>使用 table() 函数将其表格化(频数统计)</p>	<p>若 Arthritis\$Improved 是一个因子或有序型因子,可以使用函数_____快速创建一幅垂直条形图,而无需_____</p>

#### 6.1.2 堆砌条形图和分组条形图<sup>62</sup>

<p>堆砌条形图</p> <p>分组条形图</p>	<p>如果 barplot(height)中的 height 是一个矩阵而不是一个向量,则绘图结果将是一幅_____或_____。</p>
<pre>beside=FALSE beside=TRUE</pre>	<p>若_____ (默认值),则将绘制堆砌条形图。(矩阵中的每一列都将生成图中的一个条形,各列中的值将给出堆砌的“子条”的高度)</p> <p>若_____,则将绘制分组条形图(矩阵中的每一列都表示一个分组,各列中的值将并列而不是堆砌)。</p>
<pre>library(vcd) counts &lt;- table(Arthritis\$Improved, Arthritis\$Treatment)  barplot(counts, col=c("red", "yellow", "green"), legend=rownames(counts))<sup>63</sup>  barplot(counts, col=c("red", "yellow", "green"), legend=rownames(counts), beside=TRUE)</pre>	<p>载入 vcd 包;</p> <p>根据 Arthritis 数据集中的 Improved 变量和 Treatment 变量作列联表,其中 Improved 为行变量, Treatment 为列变量,将结果保存在 counts 中;</p> <p>作 counts 的堆砌条形图,颜色为红、黄、绿,图例为 counts 的行名称;</p> <p>作 counts 的分组条形图,颜色为红、黄、绿,图例为 counts 的行</p>

<sup>62</sup> 实际上堆砌条形图和分组条形图都是在描绘列联表各列变量中行变量的频次分布。

<sup>63</sup> 由于设定图形名称,横轴名称,纵轴名称的参数同质化严重,为了精简命令,本手册删去了绝大部分涉及 main=, xlab=, ylab=参数的练习,或者仅以名称的首字母缩写代替名称。

	名称
--	----

### 6.1.3 均值条形图

<pre>means &lt;- aggregate(states\$Illiteracy, by = list(state.region), FUN=mean)  means &lt;- means[order(means\$x), ]  barplot(means\$x, names.arg=means\$Group. 1)</pre>	<p>已知:</p> <p><code>states &lt;- data.frame(state.region, state.x77)</code><sup>64</sup></p> <p>将 <code>states</code> 数据集的 <code>Illiteracy</code> 变量按 <code>state.region</code> 整合，函数为平均值，将结果保存在 <code>means</code> 中；</p> <p>将 <code>means</code> 中的变量 <code>x</code>（即 <code>Illiteracy</code> 变量按 <code>state.region</code> 整合后的各均值）从小到大排序；</p> <p>绘制 <code>means\$x</code> 的条形图，各条形位于横轴的标签为 <code>means\$Group.1</code></p>
---	--

### 6.1.4 条形图的微调

<code>par(mar=c(5, 8, 4, 2))</code>	增加图形的边界大小（从原来的 <code>c(5,4,4,2)</code> 变为 <code>c(5,8,4,2)</code> ）
<code>par(las=2)</code>	旋转条形的标签，使标签垂直于坐标轴
<pre>barplot(counts, main="T0", horiz=TRUE, cex.names=0.8, names.arg=c("NI", "SI", "MI"))</pre>	<p>已知:</p> <p><code>counts &lt;- table(Arthritis\$Improved)</code></p> <p>绘制 <code>counts</code> 的水平条形图，图形名称为 <code>T0</code>，标签字号为 <code>0.8</code>，各条形的标签依次为 <code>NI</code>，<code>SI</code>，<code>MI</code></p>

### 6.1.5 棘状图

<pre>library(vcd) attach(Arthritis) counts &lt;- table(Treatment, Improved) spine(counts, main="SE") detach(Arthritis)</pre>	<p>载入 <code>vcd</code> 包；</p> <p>绑定 <code>Arthritis</code> 数据集；</p> <p>根据 <code>Improved</code> 变量和 <code>Treatment</code> 变量作列联表，其中 <code>Improved</code> 为行变量，<code>Treatment</code> 为列变量，将结果保存在 <code>counts</code> 中；</p> <p>作 <code>counts</code> 的棘状图，图形名称为 <code>SE</code>；</p> <p>解除绑定</p>
--	--

## 6.2 饼图

<pre>pie(x, labels)</pre> <p><code>x</code> 是一个非负数值向量，表示每个扇形的面积</p> <p><code>labels</code> 则是表示各扇形标签的字符型向量</p>	<p>饼图可由_____函数创建。</p> <p>解释其参数的含义</p>
--	---------------------------------------

<sup>64</sup> `state.region` 是 50 行\*1 列的因子型。`state.x77` 是一个 50 行\*8 列的矩阵，且带有行名称。

	<p>已知:</p> <pre>slices &lt;- c(10, 12, 4, 16, 8) lbls &lt;- c("US", "UK", "Australia", "Germany", "France")</pre>
<code>par(mfrow=c(2, 2))</code>	创建 2*2 的图形矩阵
<code>pie(slices, labels = lbls)</code>	<p>图 1:</p> <p>作 slices 的饼图, 以 lbls 为标签</p>
<pre>pct &lt;- round(slices/sum(slices)*100) lbls2 &lt;- paste(lbls, " ", pct, "%", sep="") pie(slices, labels=lbls2, col=rainbow(length(lbls2)))</pre>	<p>图 2:</p> <p>将 slices 转化为比例值, 保存在 pct 中 (比例值四舍五入至整数);</p> <p>将 lbls 和 pct 链接起来, 中间空一格, 并在 pct 的值后面加上百分号 “%”, 分隔符为空, 保存在 lbls2 中;</p> <p>作 slices 的饼图, 标签为 lbls2, 有几个子块就为图形添加几色彩虹色</p>
<pre>library(plotrix) pie3D(slices, labels=lbls, explode=0.1)</pre>	<p>图 3:</p> <p>载入创建三维饼图的包;</p> <p>做 slices 的三维饼图, 以 lbls 为标签, 各字块的分散程度为 0.1</p>
<pre>mytable &lt;- table(state.region) lbls3 &lt;- paste(names(mytable), "\n", mytable, sep="") pie(mytable, labels = lbls3)</pre>	<p>图 4:</p> <p>将 state.region 作频数统计, 保存在 mytable 中;</p> <p>生成包含 mytable 中所有变量名称的字符型向量, 保存在 name 中;</p> <p>将 name 和 mytable 连起来, 中间另起一行, 分隔符为空, 保存在 lbls3 中;</p> <p>作 mytable 的饼图, 标签为 lbls3</p>

<code>library(plotrix)</code>	载入绘制扇形的包
<code>fan.plot(slices, labels = lbls)</code>	<p>已知:</p> <pre>slices &lt;- c(10, 12, 4, 16, 8) lbls &lt;- c("US", "UK", "Australia", "Germany", "France")</pre> <p>绘制 slices 的扇形图, 标签为 lbls</p>

## 6.3 直方图

<pre>hist(x) freq=FALSE</pre>	使用_____函数创建直方图
-------------------------------	----------------



breaks	参数_____表示根据概率密度而不是频数绘制图形 参数_____用于控制组的数量
--------	---

par(mfrow=c(2, 2))	创建 2*2 的图形矩阵
hist(mtcars\$mpg)	图 1: 作 mtcars 数据集中 mpg 变量的频数直方图
hist(mtcars\$mpg, breaks=12, col="red", xlab="MPG", main="CH")	图 2: 作 mtcars 数据集中 mpg 变量的频数直方图, 将组数指定为 12, 颜色为红色, 横轴名称为 MPG, 图形名称为 CH
hist(mtcars\$mpg, freq=FALSE, breaks=12, col="red", xlab="MPG", main="HR") rug(jitter(mtcars\$mpg, amount=0.01)) lines(density(mtcars\$mpg), col="blue", lwd=2)	图 3; 根据概率密度作 mtcars 数据集中 mpg 变量的直方图, 将组数指定为 12, 颜色为红色, 横轴名称为 MPG, 图形名称为 HR; 添加 mpg 的轴须图, 并按 0.01 将结打散 <sup>65</sup> ; 添加 mpg 的密度曲线, 蓝色, 双倍默认宽度
h<- hist(x, breaks=12, col="red", xlab="MPG", main="HW") xfit<- seq(min(x), max(x), length=40) yfit<- dnorm(xfit, mean=mean(x), sd=sd(x)) mid <- h\$mids[1:2] yfit <- yfit*diff(mid)*length(x) lines(xfit, yfit, col="blue", lwd=2) box()	图 4: 已知: x <- mtcars\$mpg  作 x 的频数直方图, 组数为 12, 颜色为红色, 横轴名称为 MPG, 图名称为 HW, 保存在 h 中; 生成序列 xfit, 该序列最小值为 x 的最小值, 最大值为 x 的最大值, 中间各值为最小值至最大值间的等差序列, 共 40 个值; 令 yfit 为 xfit 的正态分布密度函数, 均值为 x 的均值, 标准差为 x 的标准差 <sup>66</sup> ; 令 mid <- h\$mids[1:2] <sup>67</sup>

<sup>65</sup> 轴须图用对应的小竖线表示 mpg 中各值在横轴上的位置, 某一临近区域内的小竖线越密集说明该区域的取值比较多。当同一个位置上有多个相同取值时, 由于这些值的小竖线都重叠在一起, 看不出来有多少个, 因此用 amount=0.01 将这些值在±0.01中均匀分布随机数。例如, 原先的值为 (7, 7, 7, 7), 其小竖线都重合在一起, 现在将其打散为 (7.002, 6.995, 7.008, 6.992), 其竖线就不会重合在一起了。

<sup>66</sup> 若此处不定义均值和标准差, 将默认均值为 0, 标准差为 1。

<sup>67</sup> h\$mids 表示各组切分点的中间值。例如 h 的 13 个切分点(即切分成 12 组)分别是 10,12,14,16,18,20,22,24,26,28,30,32,34, 那么 h\$mids 就是 11,13,15,17,19,21,23,25,27,29,31,33。而 h\$mids[1:2]取 h\$mids 的前 2 个值, 也就是 11 和 13。

	<p>将 <code>yfit</code> 乘以 <code>mid</code> 所表示的值域大小，再乘以 <code>x</code> 中的数量，将结果保存在 <code>yfit2</code> 中<sup>68</sup>；</p> <p>在原图上叠加正态曲线，蓝色，宽度为默认值的 2 倍；</p> <p>生成围绕图形的框线</p>
--	--

## 6.4 核密度图

<code>plot(density(x))</code>	_____命令可绘制核密度图（不叠加到另一幅图）
-------------------------------	--------------------------

<code>par(mfrow=c(2, 1))</code>	创建 2*1 的图形矩阵
<pre>d &lt;- density(mtcars\$mpg) plot(d)</pre>	<p>图 1:</p> <p>绘制 <code>mtcars</code> 数据集中 <code>mpg</code> 变量的核密度图</p>
<pre>d &lt;- density(mtcars\$mpg) plot(d, main="KD") polygon(d, col="red", border="blue")<sup>69</sup> rug(mtcars\$mpg, col="brown")</pre>	<p>图 2:</p> <p>绘制 <code>mtcars</code> 数据集中 <code>mpg</code> 变量的核密度图，图名称为 <code>KD</code>；</p> <p>将上核密度图内填充红色，并将曲线修改为红色；</p> <p>添加棕色的轴须图</p>

<p><b>sm 包</b></p> <p><code>sm.density.compare(x, factor)</code></p>	<p>_____包中的 _____函数可向图形叠加两组或更多的核密度图<sup>70</sup></p>
<pre>par(lwd=2) library(sm) attach(mtcars) cyl.f &lt;- factor(cyl, levels= c(4, 6, 8), labels = c("4 cylinder", "6 cylinder", "8 cylinder")) sm.density.compare(mpg, cyl, xlab="MPG") title(main="MPGDCC") colfill &lt;- c(2: (1+length(levels(cyl.f))))<sup>71</sup></pre>	<p>设定线宽为 2；</p> <p>载入可绘制可比较的核密度图的包；</p> <p>绑定 <code>mtcars</code> 数据集；</p> <p>将 <code>cyl</code> 变量分组创建因子型，各因子的实际值依次为 4,6,8，为各因子依次添加标签 4 cylinder, 6 cylinder,8 cylinder，将结果保存在 <code>cyl.f</code> 中；</p> <p>绘制不同等级 <code>cyl</code> 汽车的 <code>mpg</code> 的核密度图，横轴为 <code>MPG</code>；</p>

<sup>68</sup> 这一步的目的是：`yfit` 作为密度函数，其曲线下方的面积为 1，如果把这根曲线添加到前一步绘制的频数分布的图的坐标系里，这根密度函数的曲线就太低矮了（原图的纵轴高度为 7，而 `yfit` 的各元素取值在 0.004 至 0.07 之间）。因此，这一步在于把 `yfit` 的曲线“拉高”。

<sup>69</sup> `polygon()` 根据 `x` 和 `y` 作为多边形的顶点来绘制多边形

<sup>70</sup> `sm.density.compare(x, factor)` 并不是单纯的把几幅没有关联的核密度图叠加在一起，更准确的描述应该是其可以叠加绘制 `x` 在不同 `factor` 等级中的核密度图。

<sup>71</sup> 这里用了一个比较复杂的方式生成了 3 色颜色值组成的向量，即 `c(2: (1+length(levels(cyl.f))))` 的值为 `c(2, 3, 4)`，其目的是得到和 `sm.density.compare()` 生成的三条曲线对应的颜色。也许你已经注意到我们并没有在 `sm.density.compare()` 设定颜色参数，但其为了区分线条，自动从色号 2 开始为曲线赋颜色。这也是为什么 `colfill <- c(2: (1+length(levels(cyl.f))))` 要从 2 开始取色号，因为色号 1 是黑色，为了生成彩色图像，`sm.density.compare()` 从色号 2 开始取色。为了生成的图形不要那么花哨，本人还是建议用黑色，但用不同线型区分曲线。例如：`sm.density.compare(mpg, cyl, col=(1, 1, 1), lty=c(1, 2, 3))`。

<pre>legend(locator(1), levels(cyl.f), fill= colfill) detach(mtcars)</pre>	<p>给上图添加图名 MPGDCC;</p> <p>创建颜色向量 colfill, 使其和生成的颜色对应上图中三条密度图曲线的颜色;</p> <p>用鼠标单击添加图例, 图例的标签为 cyl.f 的各水平 (字符向量), 用颜色向量 colfill 为图例创建 3 色填充的盒型图例;</p> <p>解除绑定</p>
--	--

## 6.5 箱线图

<pre>boxplot(mtcars\$mpg, main="BP", ylab="MPG")</pre>	<p>绘制 mtcars 数据集中 mpg 变量的箱线图, 图名称为 BP, 纵轴名称为 MPG</p>
<pre>boxplot.stats(mtcars\$mpg)</pre>	<p>输出用于构建上图的统计量</p>

### 6.5.1 使用并列箱线图进行跨组比较

<pre>boxplot(formula, data=dataframe)</pre>	<p>箱线图可以展示单个变量或分组变量, 其使用格式为_____</p>
<pre>y ~ A y ~ A*B</pre>	<p>公式_____将为类别型变量 A 的每个值并列地生成数值型变量 y 的箱线图</p> <p>公式_____将为类别型变量 A 和 B 所有水平的两两组合生成数值型变量 y 的箱线图</p>
<pre>varwidth=TRUE</pre>	<p>参数_____使箱线图的宽度与其样本大小的平方根成正比</p>
<pre>horizontal=TRUE</pre>	<p>参数_____可以反转坐标轴的方向。</p>

<pre>boxplot(mpg ~ cyl, data=mtcars, main="CMD", xlab="NC", ylab="MPG")</pre>	<p>针对 mtcars 数据集, 按气缸数 (cyl) 分组绘制每加仑汽油行驶的英里数 (mpg) 的箱线图, 图名称为 CMD, 横轴名称为 NC, 纵轴名称为 MPG</p>
---	--

<pre>notch=TRUE</pre>	<p>通过添加参数_____, 可以得到含凹槽的箱线图 (若两个箱的凹槽互不重叠, 则表明它们的中位数有显著差异)</p>
-----------------------	---

<pre>mtcars\$cyl.f &lt;- factor(mtcars\$cyl,   levels=c(4, 6, 8),   labels=c("4", "6", "8"))</pre>	<p>将 mtcars 数据集中的 cyl 变量因子化, 各因子水平为 4,6,8, 分别赋标签 4,6,8, 保存在 mtcars\$cyl.f 中</p>
--	---

(其实 sm.density.compare 的默认线型就是冲 1 开始取的, 这里写出 lty=c(1, 2, 3) 表示强调。) 同时设定图例为: legend(locator(1), levels(cyl.f), lty=c(1, 2, 3))。

<pre>mtcars\$am.f &lt;- factor(mtcars\$am,   levels=c(0, 1),   labels=c("auto", "standard"))</pre>	<p>将 mtcars 数据集中的 am 变量因子化，各因子水平为 0,1，分别赋标签 auto,standard，保存在 mtcars\$am.f 中</p>
<pre>boxplot(mpg ~ am.f *cyl.f,   data=mtcars,   varwidth=TRUE)</pre>	<p>针对 mtcars 数据集，按气缸数（cyl）和换挡方式（am）分组绘制每加仑汽油行驶的英里数（mpg）的箱线图，使箱线图的宽度与其样本大小的平方根成正比</p>

### 6.5.2 小提琴图

<pre>library(violplot) x1 &lt;- mtcars\$mpg[mtcars\$cyl==4] x2 &lt;- mtcars\$mpg[mtcars\$cyl==6] x3 &lt;- mtcars\$mpg[mtcars\$cyl==8] violplot(x1, x2, x3,   names=c("4 cyl", "6 cyl", "8 cyl"),   col="gold") title("VP")</pre>	<p>载入绘制小提琴图的包；</p> <p>提取 mtcars 数据集中 cyl 为 4 的 mpg，保存在 x1 中；</p> <p>提取 mtcars 数据集中 cyl 为 6 的 mpg，保存在 x2 中；</p> <p>提取 mtcars 数据集中 cyl 为 8 的 mpg，保存在 x3 中；</p> <p>绘制 x1,x2,x3 的小提琴图，指定横轴标签分别为 4 cyl, 6 cyl, 8 cyl，颜色为金色；</p> <p>为图像添加标题 VP</p>
--	--

## 6.6 点图

<pre>dotchart(mtcars\$mpg,   labels=row.names(mtcars), cex=0.7,   main="GMCM",   xlab="MPG")</pre>	<p>作 mtcars 数据集中 mpg 变量的点图，各点对应的纵轴标签为 mtcars 的行名称，标签字号为默认值的 0.7 倍，图标题为 GMCM，横轴标题为 MPG</p>
--	---

<pre>x &lt;- mtcars[order(mtcars\$mpg), ] x\$cyl &lt;- factor(x\$cyl) x\$color[x\$cyl==4] &lt;- "red" x\$color[x\$cyl==6] &lt;- "blue" x\$color[x\$cyl==8] &lt;- "darkgreen" dotchart(x\$mpg,   labels = row.names(x),   cex=.7,   groups = x\$cyl,   gcolor = "black",   color = x\$color,   pch=19)</pre>	<p>将 mtcars 按 mpg 由小到大排序，保存到 x 中；</p> <p>将 x 中的 cyl 变量因子化；</p> <p>将字符型变量 color 添加到 x 中，其中，当 cyl 为 4 时，color 为“red”，当 cyl 为 6 时，color 为“blue”，当 cyl 为 8 时，color 为“darkgreen”；</p> <p>作 x 数据集中 mpg 变量的点图，各点对应的纵轴标签为 x 的行名称，标签字号为默认值的 0.7 倍，按 cyl 分组，数字 4,6,8 以黑色显示<sup>72</sup>，点和标签的颜色由向量 color 决定，点以填充的圆圈表示</p>
---	---

<sup>72</sup> 即图形最左侧纵轴标签中用于区分各组的数字 4,6,8。

## 第 7 章 基本统计分析

### 7.1 描述性统计分析

<code>head(mtcars[vars])</code>	<p>已知:</p> <pre>vars &lt;- c("mpg", "hp", "wt")</pre> <p>获取 mtcars 数据集中由 vars 所涵盖的变量的开始部分（前 6 行）</p>
---------------------------------	--

#### 7.1.1 方法云集

用 `summary()` 计算描述性统计

<code>summary(mtcars[vars])</code>	用 <code>summary</code> 得到 <code>mtcars[vars]</code> 的描述性统计
------------------------------------	--

用 `sapply()` 计算描述性统计

<p><code>sapply(x, FUN, options)</code></p> <p><b>x</b> 是数据框（或矩阵）</p> <p><b>FUN</b> 为一个任意的函数（典型函数有 <code>mean()</code>、<code>sd()</code>、<code>var()</code>、<code>min()</code>、<code>max()</code>、<code>median()</code>、<code>length()</code>、<code>range()</code> 和 <code>quantile()</code>）</p> <p><b>option</b> 是可影响 <b>FUN</b> 且被传递到 <b>FUN</b> 的参数</p>	<p><code>sapply()</code> 函数的语法格式为 _____</p> <p>解释各参数</p>
---	--

通过 `sapply()` 和自定义的函数计算描述性统计（包含峰度和偏度）

<pre>mystats &lt;- function(x, na.omit=FALSE){   if (na.omit)     x &lt;- x[!is.na(x)]   m &lt;- mean(x)   n &lt;- length(x)   s &lt;- sd(x)   skew &lt;- sum((x-m)^3/s^3)/n   kurt &lt;- sum((x-m)^4/s^4)/n - 3   return(c(n=n, mean=m, stdev=s, skew= skew, kurtosis=kurt)) }</pre>	<p>创建自编函数 <code>mystats</code>，求变量 <code>x</code> 的均值（用 <code>mean</code> 表示），观测量（用 <code>n</code> 表示），标准差（用 <code>stdev</code> 表示），偏度（用 <code>skew</code> 表示）<sup>73</sup>，峰度（用 <code>kurtosis</code> 表示）<sup>74</sup>，并返回这些统计量。要求：当想忽略可能存在的缺失值时，该函数能够通过定义参数 <code>na.omit=TRUE</code> 实现。<sup>75</sup></p>
<code>sapply(bl, mystats)</code>	已知:

<sup>73</sup> 偏度大于 0 表示数据较正态分布向右偏，偏度小于 0 表示数据较正态分布向左偏。

<sup>74</sup> 峰度大于 0 表示数据较正态分布更陡峭，峰度小于 0 表示数据较正态分布更平坦。

<sup>75</sup> 此处函数中使用的 `na.omit` 具有一定的迷惑性。其实这里的 `na.omit` 只是自定义的一个参数的名称，和 `na.omit()` 函数完全是两个概念。在这里用任意一个名称代替 `na.omit` 都可以。这段代码的意思是在默认情况下，即不明确给出 `na.omit` 的取值时，即便数据中有缺失值 `NA`，也该怎么计算就怎么计算。此时，如果数据中有缺失值 `NA`，那么 `m`，`stdev`，`skew`，`kurtosis` 都无法计算，会显示 `NA`。而当给出 `na.omit=TRUE` 时，即激活 `if(na.omit)` 语句，此时 `x` 只保留非 `NA` 值，原先无法计算的统计指标便可以正常计算了。在原文对这个函数进行阐述时说了这么一句话：“请注意，如果你只希望单纯地忽略缺失值，那么应当使用 `sapply(mtcars[myvars], mystats, na.omit=TRUE)`。”这句话也具有迷惑性，仿佛是在暗示 `na.omit=TRUE` 是 `sapply` 自带的参数，但其实是我们在构建函数时定义的。

<code>sapply(bl, mystats, na.omit=TRUE)</code>	<pre>vars &lt;- c("mpg", "hp", "wt") bl &lt;- mtcars[vars]</pre> <p>通过 <code>sapply</code> 和上面自定义的函数求 <code>bl</code> 的描述性统计量；</p> <p>通过 <code>sapply</code> 和上面自定义的函数求 <code>bl</code> 的描述性统计量，并要求忽略缺失值</p>
--	--

Hmisc 包中的 <code>describe()</code> 函数	_____包的_____函数可返回变量和观测的数量、info 值 <sup>76</sup> 、缺失值和唯一值的数目、平均值、分位数，以及五个最大的值和五个最小的值
<code>library(Hmisc)</code> <code>describe(bl)</code>	通过上述包及相应函数获得 <code>bl</code> 的描述性统计

<p><code>pastecs</code> 包中有一个名为 <code>stat.desc()</code> 的函数</p> <pre>stat.desc(x, basic=TRUE, desc=TRUE, norm=FALSE, p=0.95)</pre> <p><code>x</code> 是一个数据框或时间序列</p> <p>若 <code>basic=TRUE</code>（默认值），则计算其中所有值、空值、缺失值的数量，以及最小值、最大值、值域，还有总和。</p> <p>若 <code>desc=TRUE</code>（同样也是默认值），计算中位数、平均数、平均数的标准误、平均数置信度为 <b>95%</b> 的置信区间、方差、标准差以及变异系数。</p> <p>若 <code>norm=TRUE</code>（不是默认的），则返回正态分布统计量，包括偏度和峰度（以及它们的统计显著程度）和 <b>Shapiro-Wilk</b> 正态检验结果。</p> <p><code>p</code> 值来计算平均数的置信区间（默认置信度为 <b>0.95</b>）。</p>	<p>_____包的_____函数可以计算种类繁多的描述性统计量，其语法格式为_____</p> <p>解释其参数用法</p>
---	---

<sup>76</sup> 当前，Hmisc 包的 `describe()` 函数会报一个 `info` 值，该值衡量了一个序列的值的连续程度，可以近似理解为序列中相同值多不多。当不同的值越多时，相同值带来的影响越小。`info` 在 0 到 1 之前取值。一个序列中的值全部相同时，`info` 值为 0，如果全部不同的话 `info` 值为 1。`info` 的公式为：

$$\text{info} = \frac{1 - \sum (\text{各观测值的相对频率})^3}{1 - (\frac{1}{\text{样本量}})^2}$$

例如，当 `a<-c(0,1)` 时，`a` 的 `info` 值为 1。当 `b<-c(1,1,1,1)` 时，`b` 的 `info` 值为 0。注意各序列中的值的绝对大小不影响 `info`，例如 `d<-c(10,10,10,10)` 的 `info` 值仍然为 0。而若 `e<-c(1,1,0)`，`f<-c(1,1,1,0)`，前者的 `info` 为 0.75，后者的 `info` 为 0.6。

<code>library(pastecs)</code> <code>stat.desc(bl)</code> <sup>77</sup>	通过上述包及相应函数获得 <code>bl</code> 的描述性统计
---	-------------------------------------

<code>psych</code> 包的 <code>describe()</code> 的函数	_____包的_____函数可以计算非缺失值的数量、平均数、标准差、中位数、截尾均值 <sup>78</sup> 、绝对中位差、最小值、最大值、值域、偏度、峰度和平均值的标准误。
<code>library(psych)</code> <code>describe(bl)</code>	通过上述包及相应函数获得 <code>bl</code> 的描述性统计

R 会优先使用最后载入的包	<code>psych</code> 包和 <code>Hmisc</code> 包均提供了名为 <code>describe()</code> 的函数。R 如何知道该使用哪个呢？
<code>Hmisc::describe()</code>	如果明确想调用 <code>Hmisc</code> 包中的 <code>describe()</code> 函数，可以键入_____

### 7.1.2 分组计算描述性统计量

使用 `aggregate()` 分组获取描述性统计量

<code>aggregate(bl, by=list(am=mtcars\$am), mean)</code>	已知：  <code>vars &lt;- c("mpg", "hp", "wt")</code>  <code>bl &lt;- mtcars[vars]</code>  使用 <code>aggregate()</code> 函数，依据换挡方式（自动挡/手动挡，即 <code>am</code> 变量）分组求 <code>bl</code> 中各变量的均值
<code>aggregate(bl, by=list(am=mtcars\$am), sd)</code>	接上例， 使用 <code>aggregate()</code> 函数，依据换挡方式（自动挡/手动挡，即 <code>am</code> 变量）分组求 <code>bl</code> 中各变量的标准差
返回的结果中的 <code>am</code> 列将被标注为 <code>Group. 1</code> 而不是 <code>am</code>	接上例， 如果使用的是 <code>list(mtcars\$am)</code> 而不是 <code>list(am=mtcars\$am)</code> 会怎样？
<code>aggregate(bl, by=list(am=mtcars\$am, cyl=mtcars\$cyl), mean)</code>	【补】接上例， 使用 <code>aggregate()</code> 函数，同时依据换挡方式（自动挡/手动挡，即 <code>am</code> 变量）和气缸数（即 <code>cyl</code> ）分组求 <code>bl</code> 中各变量的均值
<code>aggregate()</code> 仅允许在每次调用中使用平均数、标准差这样的单返回值函数。它无法一次返回若干个统计量。	使用 <code>aggregate()</code> 函数分组获得描述性统计在功能上的缺陷是什么？可以采用_____函数弥补，其格式为_____

<sup>77</sup> 在代码清单 7-4 生成的结果中（第一版 132 页，第二版 133 页）`SE.mean` 是平均数的标准误，`CI.mean.0.95` 是平均数的置信区间，`var` 是方差，`std.dev` 是标准差，`coef.var` 是变异系数。

<sup>78</sup> 默认 `trim=0.1`，即首尾各去掉 10%，这一项可自定义。

<p>可以使用 <code>by()</code> 函数完成这项任务。格式为：</p> <p><code>by(data, INDICES, FUN)</code></p> <p>其中 <code>data</code> 是一个数据框或矩阵，<code>INDICES</code> 是一个因子或因子组成的列表，定义了分组，<code>FUN</code> 是任意函数</p>	
--	--

使用 `by()` 分组计算描述性统计量 <sup>79</sup>

<pre>dstats &lt;- function(x) sapply(x, mystats)  by(bl, list(am=mtcars\$am, cyl=mtcars\$cyl), dstats) <sup>80</sup></pre>	<p>已知：</p> <p><code>mystats</code> 为 7.1.1 节中创建的自编函数</p> <pre>vars &lt;- c("mpg", "hp", "wt")  bl&lt;-mtcars[vars]</pre> <p>同时依据换挡方式（自动挡/手动挡，即 <code>am</code> 变量）和气缸数（即 <code>cyl</code>）分组求 <code>bl</code> 中各变量的描述性统计量（由 <code>mystats</code> 而定）：</p> <p>生成自编函数 <code>dstats</code>，其可将 <code>mystats</code> 函数应用在数据框 <code>x</code> 的每个变量上；</p> <p>用 <code>by()</code> 分组计算描述性统计量</p>
--	--

`doBy` 包

<pre>summaryBy()  summaryBy(formula, data=dataframe, FUN= function)  formula 接受以下的格式： var1 + var2 + var3 + ... + varN ~ groupvar1 + groupvar2 + ... + groupvarN</pre> <p>在~左侧的变量是需要分析的数值型变量，而右侧的变量是类别型的分组变量</p>	<p><code>doBy</code> 包中的_____函数可以分组计算概述统计量。其语法格式为_____，其中 <code>formula</code> 的格式为_____</p>
<pre>library(doBy) summaryBy(mpg+hp+wt~am+cyl, data=mtcars, FUN=mystats)</pre>	<p>已知：</p> <p><code>mystats</code> 为 7.1.1 节中创建的自编函数</p> <p>使用 <code>doBy</code> 包中的 <code>summaryBy()</code> 同时依据换挡方式(自动挡/手动挡，</p>

<sup>79</sup> 此例在原书第二版内容的基础上改编。

<sup>80</sup> 可能会存在疑问为什么不直接写成 `by(bl, list(am=mtcars$am, cyl=mtcars$cyl), mystats)`，而需要先生成 `dstats <- function(x) sapply(x, mystats)`，再在 `by()` 中引用 `dstats`。原因在于 `mystats` 只能应用在某个向量上，如果想要将其应用在一个数据框的每个变量上，需要用 `sapply(x, mystats)` 实现。

个人认为用 `by()` 求分组描述性统计简单直观，这里统计函数还可以采用 7.1.1 节中的 `summary()` 以及 `Hmisc` 包中的 `describe()` 等。



	即 <code>am</code> 变量) 和气缸数 (即 <code>cyl</code> ) 分组求 <code>mtcars</code> 数据框中 <code>mpg</code> , <code>hp</code> , <code>wt</code> 变量的描述性统计量 (由 <code>mystats</code> 而定) <sup>81</sup>
--	--

## psych 包

<pre>library(psych) describe.by(bl, mtcars\$am)</pre>	<p>已知:</p> <pre>vars &lt;- c("mpg", "hp", "wt") bl&lt;-mtcars[vars]</pre> <p>使用 <code>psych</code> 包中的 <code>describe.by()</code> 函数, 依据换挡方式 (自动挡/手动挡, 即 <code>am</code> 变量) 分组求 <code>bl</code> 中各变量描述性统计量</p>
<p>1. 不允许指定任意函数</p> <p>2. 仅在分组变量交叉后不出现空白单元时有效</p>	<p><code>describe.by()</code> 函数的两个缺陷是什么?</p>

reshape 包	可以通过_____包的融合和重铸功能来获取分组描述性统计
<pre>dfm &lt;- melt(dataframe, measure.vars=y, id.vars=g)</pre> <p><code>dataframe</code> 是数据框</p> <p><code>y</code> 为一个向量, 指明了要进行概述的数值型变量 (默认使用所有变量)</p> <p><code>g</code> 是一个或多个分组变量组成的向量</p>	<p>进行融合的语法结构为_____</p> <p>解释其参数含义</p>
<pre>cast(dfm, groupvar1+groupvar2+...+variable~., FUN)</pre> <p>分组变量以+号分隔</p> <p><code>variable</code> 表示重铸后数据框中的变量 <code>variable</code><sup>82</sup></p> <p><code>FUN</code> 为任意函数</p>	<p>进行重铸语法结构为_____</p>
<pre>library(reshape) dstats &lt;- function(x) (c(n=length(x), mean=mean(x), sd=sd(x))) dfm &lt;- melt(mtcars, measure.vars = c("mpg", "hp", "wt"), id.vars = c("am", "cyl")) cast(dfm, am + cyl + variable ~ ., dstats)</pre>	<p>载入融合、重铸的包;</p> <p>自编函数 <code>dstats</code>, 使其求变量 <code>x</code> 的观测值数量 (用 <code>n</code> 表示), 均值 (用 <code>mean</code> 表示), 标准差 (用 <code>sd</code> 表示);</p> <p>融合 <code>mtcars</code> 数据集, 要进行概述的数值型变量为 <code>mpg</code>, <code>hp</code>, <code>wt</code>,</p> <p>分组变量为 <code>am</code> 和 <code>cyl</code>, 保存在 <code>dfm</code> 中;</p> <p>融合数据, 函数为 <code>dstats</code></p>

## 7.1.5 结果的可视化

## 7.2 频数表和列联表

<code>library(vcd)</code>	载入 <code>vcd</code> 包;
---------------------------	------------------------

<sup>81</sup> 此例在原书内容上有改动。

<sup>82</sup> 也可以理解成融合后数据框中的变量 `variable`

<code>head(Arthritis)</code>	查看数据集 Arthritis 的前 6 行
------------------------------	------------------------

### 7.2.1 生成频数表

#### 一维列联表

<code>mytable &lt;- with(Arthritis, table(Improved))</code>	使用 <code>table()</code> 生成 Arthritis 数据框中 Improved 变量的列联表，并保存在 mytable 中
<code>prop.table(mytable)</code>	使用_____命令可将频数转变为比例值
<code>prop.table(mytable)*100</code>	使用_____命令可将频数转变为百分比

#### 二维列联表

<code>mytable &lt;- with(Arthritis, table(Treatment, Sex))</code>	使用 <code>table()</code> 生成 Arthritis 数据集中 Treatment 和 Sex 变量的二维列联表，并保存在 mytable 中，其中 Treatment 作行变量，Sex 作列变量
---	--

<code>mytable &lt;- xtabs(~ A + B, data=mydata)</code> 若某个变量写在公式的左侧，则其为一个频数向量 <sup>83</sup>	<code>xtabs()</code> 生成列联表的语法格式为_____，其中 A 为行变量，B 为列变量 若某个变量出现在~的左侧，表示什么？
<code>mytable &lt;- xtabs(~ Treatment+Improved, data=Arthritis)</code>	使用 <code>xtabs()</code> 生成 Arthritis 数据框中 Treatment 和 Sex 变量的二维列联表，其中 Treatment 作行变量，Sex 作列变量，结果保存在 mytable 中

<code>margin.table(mytable, 1)</code>	生成 mytable 的各行和（代 <code>xtabs()</code> 语句中~右边的第一个变量）
<code>prop.table(mytable, 1)</code>	生成 mytable 的各行比例
<code>margin.table(mytable, 2)</code>	生成 mytable 的各列和（代 <code>xtabs()</code> 语句中~右边的第二个变量）
<code>prop.table(mytable, 2)</code>	生成 mytable 的各列比例
<code>prop.table(mytable)</code>	生成 mytable 的各单元格比例

<sup>83</sup> 例如，`f<-data.frame(a=c(0.2,0.1,0.2,0.3,0.4),b=c("M", "F", "F", "M", "M"),e=c("A", "B", "A", "B", "A"))`。命令 `mytable<-xtabs(~b+e,data=f)`，得到：

```
e
b  A B
F  1 1
M  2 1
```

命令 `mytable<-xtabs(a~b+e,data=f)` 得到：

```
e
b  A  B
F  0.2 0.1
M  0.6 0.3
```

<code>addmargins(mytable)</code>	为 <code>mytable</code> 添加各行和以及各列和
<code>addmargins(prop.table(mytable))</code>	生成 <code>mytable</code> 的各单元格比例, 并为其添加各单元格比例的各行和以及各列和
<code>addmargins(mytable, 2)</code> <sup>84</sup>	为 <code>mytable</code> 添加各行和
<code>addmargins(mytable, 1)</code>	为 <code>mytable</code> 添加各列和

<code>gmodels</code> 包中的 <code>CrossTable()</code> 函数	使用_____包中的_____函数可以创建二维列联表
<code>library(gmodels)</code> <code>CrossTable(vcd::Arthritis\$Treatment, vcd::Arthritis\$Improved)</code> <sup>85</sup>	使用上述函数生成 <code>Arthritis</code> 数据集中 <code>Treatment</code> 和 <code>Improved</code> 变量的二维列联表

### 三维列联表

<code>mytable &lt;- xtabs(~ Treatment+Sex+Improved, data= vcd::Arthritis)</code>	用 <code>xtabs()</code> 生成 <code>Arthritis</code> 数据集中 <code>Treatment</code> , <code>Sex</code> , <code>Improved</code> 三个变量的三维列联表, 保存在 <code>mytable</code> 中
<code>fTable(mytable)</code>	用命令_____把 <code>mytable</code> 变成一种更紧凑的输出形式
<code>margins.table(mytable, 1)</code>	根据 <code>xtabs()</code> 中~右边的第一个变量(即 <code>Treatment</code> )生成 <code>mytable</code> 的边际频数
<code>margins.table(mytable, 2)</code>	根据 <code>xtabs()</code> 中~右边的第二个变量(即 <code>Sex</code> )生成 <code>mytable</code> 的边际频数
<code>margins.table(mytable, 3)</code>	根据 <code>xtabs()</code> 中~右边的第三个变量(即 <code>Improved</code> )生成 <code>mytable</code> 的边际频数
<code>margins.table(mytable, c(1, 3))</code>	根据 <code>xtabs()</code> 中~右边的第一个和第三个变量(即 <code>Treatment</code> 和 <code>Improved</code> )生成 <code>mytable</code> 的边际频数
<code>fTable(prop.table(mytable, c(1, 2)))</code>	根据 <code>xtabs()</code> 中~右边的第一个和第二个变量(即 <code>Treatment</code> 和 <code>Sex</code> )生成 <code>mytable</code> 的比例, 并用 <code>fTable()</code> 输出
<code>fTable(addmargins(prop.table(mytable, c(1, 2)), 3)) * 100</code>	根据 <code>xtabs()</code> 中~右边的第一个和第二个变量(即 <code>Treatment</code> 和 <code>Sex</code> )生成 <code>mytable</code> 的比例; 在上一步的基础上, 为~右边的第三个变量添加边际和; 用 <code>fTable()</code> 输出以上内容; 得到百分比

<sup>84</sup> 注意区分“生成各行和”与“添加各列和”使用的是数字 1, 而“生成各列和”与“添加各行和”使用的是数字 2。

<sup>85</sup> 原书此处的代码没有指定 `Arthritis` 数据集出自 `vcd` 包, 因而在 `gmodels` 包中找不到, 可以手动指定 `vcd` 包。

`CrossTable()`所生成的表中的各单元格的内容自上而下分别是: 频数, 卡方贡献, 行比例, 列比例, 各单元格比例。关于卡方贡献的定义可参考 Performing experiments using FTFs (UCL): <http://www.ucl.ac.uk/english-usage/resources/ftfs/experiment.htm>

## 7.2.2 独立性检验

### 卡方独立性检验

<code>chi sq. test()</code> 函数	_____函数能对二维表的行变量和列变量进行卡方独立性检验
<code>library(vcd)</code> <code>mytable &lt;- xtabs(~Treatment+Improved, data=Arthritis)</code> <code>chi sq. test(mytable)</code>	对 Arthritis 数据集中 Treatment 和 Improved 变量做卡方独立性检验
p 值表示从总体中抽取的样本行变量与列变量是相互独立的概率 <sup>86</sup>	卡方检验所报告的 p 值表示什么？
二维列表的行变量与列变量是相互独立的	卡方检验的原假设是什么？

### Fisher 精确检验

<code>fisher.test(mytable)</code> 其中 <code>mytable</code> 是一个二维列联表	可以使用_____函数进行 Fisher 精确检验
边界固定的列联表中行和列是相互独立的	Fisher 精确检验的原假设是什么？
<code>mytable &lt;- xtabs(~Treatment+Improved, data=vcd::Arthritis)</code> <code>fisher.test(mytable)</code>	对 Arthritis 数据集中 Treatment 和 Improved 变量做 Fisher 精确检验
<code>fisher.test()</code> 函数可以在任意行列数大于等于 2 的二维列联表上使用，但不能用于 $2 \times 2$ 的列联表	R 中，Fisher 精确检验的局限是什么？

### Cochran-Mantel-Haenszel 检验

两个名义变量在第三个变量的每一层中都是条件独立的	Cochran-Mantel-Haenszel 检验的原假设是什么？
<code>mytable &lt;- xtabs(~Treatment+Improved+Sex, data=Arthritis)</code> <code>mantel haen. test(mytable)</code>	检验治疗情况 (Treatment) 和改善情况 (Improved) 在性别 (Sex) 的每一水平下是否独立

## 7.2.3 相关性的度量

vcd 包的 <code>assocstats()</code> 函数	_____包中的_____函数可以用来计算二维列联表的 phi 系数、列联系数和 Cramer's V 系数
<code>library(vcd)</code> <code>mytable &lt;- xtabs(~Treatment+Improved, data=Arthritis)</code> <code>assocstats(mytable)</code> <sup>87</sup>	用上述函数求 Arthritis 数据集中 Treatment 和 Improved 二维列联表的相关性度量

## 7.2.4 结果的可视化

### 7.2.5 将表转换为扁平格式<sup>88</sup>

<code>treatment &lt;- rep(c("Placebo", "Treated"), 3)</code> <code>improved &lt;- rep(c("None", "Some", "Marked"), 3)</code>	已知：  某列联表为
---	------------------

<sup>86</sup> 例如，若 p 很小 ( $p=0.0001$ )，我们认为变量间不独立。一般  $p<0.01$  时我们认为变量不独立。

<sup>87</sup> phi 系数和 Cramer's V 系数都在  $[0, 1]$  区间取值，越接近 1 越相关。

<sup>88</sup> 第二版中删除了这一小节。

<pre>d"), each = 2) Freq &lt;- c(29, 13, 7, 7, 7, 21) <sup>89</sup> mytable &lt;- as.data.frame(cbind(treatment, improved, Freq)) mydata &lt;- table2flat(mytable) head(mydata)</pre>	治疗情况	改善情况 improved		
	treatment	无改善 None	一定程度的改善 Some	显著改善 Marked
	安慰剂治疗 Placebo	29	7	7
	用药治疗 Treated	13	17	21

将此列联表转换为扁平格式

生成 treatment 向量，保存在 treatment 中；

生成 improved 向量，保存在 improved 中；

生成频数向量，保存在 Freq 中；

将 treatment, improved, Freq 组成数据框 mytable；

使用 table2flat( ) 函数（table2flat 是一个自编函数，参见第一版第 145 页）将数据变成扁平格式，替换 mytable；

展示 mytable 的前 6 行

## 7.3 相关

### 7.3.1 相关的类型

Pearson、Spearman 和 Kendall 相关系数矩阵

<pre>cor() cor(x, use= , method= )</pre>	<p>函数可以计算 Pearson、Spearman 和 Kendall 相关系数矩阵，其语法格式为_____</p>
<pre>use= all.obs (报错) everything (报 missing) <sup>90</sup> complete.obs (行删除) pairwise.complete.obs (成对删除)</pre>	<p>上述函数中指定缺失数据的处理方式的参数是_____。可选的方式为：</p> <p>_____（假设不存在缺失数据——遇到缺失数据时将报错）；</p> <p>_____（遇到缺失数据时，相关系数的计算结果将被设为 missing）；</p> <p>_____（行删除）；</p> <p>_____（成对删除，pairwise deletion）<sup>91</sup></p>

<sup>89</sup> 注意此向量各元素的排列与原列联表的对应关系。

<sup>90</sup> 即显示 NA。

<sup>91</sup> 下面举例说明行删除和成对删除。假设数据集 f 和 g 分别为（g 中补出了 f 中的缺失值）：

a	b	c
1	2	2
2	2	1
1	4	2
2	4	1
1	NA	2

a	b	c
1	2	2
2	2	1
1	4	2
2	4	1
1	2	2

当使用行删除时，会直接删除 f 的最后一行。使用成对删除时，只会在计算 b 和 a 的相关系数时去掉 NA 和 1 这对数据，计算 b 和 c 的相关系数时去掉 NA 和 2 这对数据，而计算 a 和 c 的相关系数时不拿掉任何数据。可以尝试一下当选择成对删除时，f 的相关系数矩

<code>pearson</code> 、 <code>spearman</code> 或 <code>kendall</code>	上述函数中指定相关系数的类型的参数是_____, 可选类型为_____, _____ 或_____
<code>use="everything"</code> 和 <code>method="pearson"</code>	上述函数的默认参数为 <code>use=_____</code> , <code>method=_____</code>
<code>cor(states, method="spearman")</code>	计算 <code>states</code> 的 Spearman 相关系数方阵

<code>cov()</code>	_____函数可以计算协方差矩阵
<pre>x &lt;- states[, c("Population", "Income", "Illiteracy", "HS Grad")] y &lt;- states[, c("Life Exp", "Murder")] cor(x, y)</pre>	<p>已知:</p> <p><code>states&lt;- state.x77[,1:6]</code>, 且 <code>states</code> 为矩阵, 其中包含的 6 个变量为 Population, Income, Illiteracy, Life Exp, Murder, HS Grad。</p> <p>计算 Population, Income, Illiteracy, HS Grad 与 Life Exp, Murder 间的相关系数矩阵</p>

## 偏相关

<p><code>ggm</code> 包的 <code>pcor()</code> 函数</p> <p><code>pcor(u, S)</code></p> <p><code>u</code> 是一个数值向量, 前两个数值表示要计算相关系数的变量下标, 其余的数值为条件变量 (即要排除影响的变量) 的下标</p> <p><code>S</code> 为变量的协方差阵</p>	<p>_____包的_____函数可以计算偏相关系数</p> <p>函数的语法格式为_____, 解释其中的参数</p>
<pre>library(ggm) colnames(states) pcor(c(1, 5, 2, 3, 6), cov(states))</pre>	<p>计算控制了收入 (Income, 2)、文盲率 (Illiteracy, 3) 和高中毕业率 (HS Grad, 6) 的影响时, 人口 (Population, 1) 和谋杀率 (Murder, 5) 之间的相关系数 (括号中的数字为各变量在 <code>states</code> 数据集中位置的编号)</p> <p>载入包;</p> <p>展示各列变量, 以决定变量所处位置的编号<sup>92</sup>;</p> <p>计算偏相关系数</p>

## 7.3.2 相关性的显著性检验

<p><code>cor.test()</code></p> <p><code>cor.test(x, y, alternative =, method =)</code></p>	<p>_____函数可以对单个的 Pearson、Spearman 和 Kendall 相关系数进行检验, 其语法格式为_____</p>
要检验相关性的变量	上述函数中 <code>x</code> 和 <code>y</code> 为_____

阵中 `a` 和 `b` 的相关系数与 `g` 的相关系数矩阵中 `a` 和 `b` 的相关系数是相同的。

成对删除在第一版 15.8.1 节, 第二版 18.8.1 节中有更详细的阐述。原书作者并不推荐使用成对删除。

<sup>92</sup> 这条命令是在第二版中新增的。

进行双侧检验或单侧检验 <sup>93</sup> "two. side"、"less"或"greater"	alternative 则用来指定_____（取值为"_____","_____"或"_____")
要计算的相关类型 "pearson"、"kendall"或"spearman"	method 用以指定_____（取值为"_____","_____"或"_____")
alternative="less" alternative="greater" alternative="two. side"side（总体相关系数不等于0）	当研究的假设为总体的相关系数小于0时，使用_____ 在研究的假设为总体的相关系数大于0时，使用_____ 默认情况下，假设为_____（表示_____）

cor.test(states[, 3], states[, 5])	检验文盲率（Illiteracy, 3）和谋杀率（Murder, 5）之间的相关系数的显著性，采用双侧检验 <sup>94</sup>
------------------------------------	---

psych包的cor.test() cor.test(x, use=, method=)	_____包的_____函数可以为Pearson、Spearman或Kendall相关计算相关矩阵，同时报告显著性水平 其语法格式为_____
"pairwise"或"complete" 成对删除或行删除	上述函数中参数use=的取值可为_____或_____（分别表示对缺失值执行_____或_____）
"pearson"（默认值）、"spearman"或"kendall"。	上述函数中参数method=的取值可为_____、_____或_____

#### 其他显著性检验

ggm包的pcor.test() pcor.test(r, q, n) 其中的r是由pcor()函数计算得到的偏相关系数 q为要控制的变量数（以数值表示位置） n为样本大小	_____包的_____函数可以用来检验在控制一个或多个额外变量时两个变量之间的条件独立性 其语法格式为_____
library(ggm) r=pcor(c(1, 5, 2, 3, 6), cov(states)) pcor.test(r, c(2, 3, 6), 50)	【补】已知样本量为50，针对states数据集，计算控制了收入（Income, 2）、文盲率（Illiteracy, 3）和高中毕业率（HS Grad, 6）的影响时，人口（Population, 1）和谋杀率（Murder, 5）之间的相关系数，并做显著性检验

#### 7.3.3 相关关系的可视化

### 7.4 t 检验

#### 7.4.1 独立样本的t检验

<sup>93</sup> 单侧或双侧检验的更多内容可参考伍德里奇《计量经济学导论》C.6节的“假设检验”。

<sup>94</sup> 此处原文有误，states[, 3]是文盲率（Illiteracy）而不是预期寿命。在第二版中此处错误没有得到更正。

两个总体的均值相等的假设 两组数据是独立的，并且是从正态总体中抽得	一个针对两组的独立样本 t 检验可以用于检验____，这里假设____
<code>t.test()</code>  <code>t.test(y ~ x, data)</code> <code>y</code> 是一个数值型变量， <code>x</code> 是一个二分变量  <code>t.test(y1, y2)</code> <sup>95</sup> <code>y1</code> 和 <code>y2</code> 为数值型向量（即各组的结果变量）	____函数可以用来做 t 检验，其可调用的语法格式有两种，分别是____和____  解释上述函数的参数
<code>var.equal=TRUE</code>	上述函数可以添加一个参数____以假定方差相等
<code>alternative="less"</code> <code>alternative="greater"</code>	可以添加参数____或____来进行有方向的检验

<code>library(MASS)</code> <code>t.test(Prob ~ So, data=UScrime)</code> <sup>96</sup>	已知：  MASS 包中的 UScrime 数据集包含了 1960 年美国 47 个州的刑罚制度对犯罪率影响的信息。其中包含的变量为 Prob（监禁的概率）；U1（14~24 岁年龄段城市男性失业率）；U2（35~39 岁年龄段城市男性失业率）；类别型变量 So（指示该州是否位于南方的指示变量），作为分组变量使用  问如果在美国的南方犯罪，是否更有可能被判监禁？
--	--

## 7.4.2 非独立样本的 t 检验

组间的差异呈正态分布 <code>t.test(y1, y2, paired=TRUE)</code> <code>y1</code> 和 <code>y2</code> 为两个非独立组的数值向量	非独立样本的 t 检验假定____  检验的调用函数语法格式为____
<code>library(MASS)</code> <code>apply(UScrime[c("U1", "U2")], function(x) (c(mean=mean(x), sd=sd(x))))</code> <sup>97</sup> <code>with(UScrime, t.test(U1, U2, paired=TRUE))</code>	载入 MASS 包；  自编函数，计算 UScrime 数据集中 U1 和 U2 变量的均值和标准差，均值用 mean 表示，标准差用 sd 表示；  做非独立样本的 t 检验，检验 U1 和 U2 均值是否相同

## 7.4.3 多于两组的情况

如果能够假设数据是从正态总体中独立抽样而得的，那么可以使用方差分析（ANOVA）	在多于两个的组之间进行比较，应该怎么做？
--	----------------------

<sup>95</sup> 这里的 `y1` 和 `y2` 就相当于已经按格式 `t.test(y ~ x, data)` 中的 `x` 分好类的 `y`。

<sup>96</sup> 或者使用 `a1<-UScrime[which(UScrime$So==1),14]` #14 是 Prob 在 UScrime 所处的列编号，`a2<-UScrime[which(UScrime$So==0),14]`，`t.test(a1,a2)`，其得到的结果和 `t.test(Prob ~ So, data=UScrime)` 的结果是一样的。

<sup>97</sup> 这条命令适用性较强，值得熟练掌握。



## 7.5 组间差异的非参数检验

非参数检验	若结果变量在本质上就严重偏倚或呈现有序关系，可以采用_____检验
-------	-----------------------------------

### 7.5.1 两组的比较

Wilcoxon 秩和检验	若两组数据独立，可以使用_____检验来评估观测是否是从相同的概率分布中抽得的（即，在一个总体中获得更高得分的概率是否比另一个总体要大）
<code>wilcox.test()</code>  <code>wilcox.test(y ~ x, data)</code> 其中的 <b>y</b> 是数值型变量，而 <b>x</b> 是一个二分变量  <code>wilcox.test(y1, y2)</code> 其中的 <b>y1</b> 和 <b>y2</b> 为各组的结果变量	_____函数可以用来做 Wilcoxon 秩和检验，其可调用的语法格式有两种，分别是_____和_____  解释上述函数的参数
<code>with(UScrime, by(Prob, So, median))</code> <code>wilcox.test(Prob ~ So, data=UScrime)</code>	针对 UScrime 数据集，对 Prob 变量按 So 分组求中位数；  针对 UScrime 数据集，做 Wilcoxon 秩和检验，检验南方州和非南方州（So）的监禁概率（Prob）是否相同
<code>sapply(UScrime[c("U1", "U2")], median)</code> <code>with(UScrime, wilcox.test(U1, U2, paired=TRUE))</code>	针对 UScrime 数据集，求 U1 和 U2 的中位数；  针对 UScrime 数据集，做 Wilcoxon 秩和检验，检验 U1（14~24 岁年龄段城市男性失业率）和 U2（35~39 岁年龄段城市男性失业率）是否相同，U1 和 U2 非独立

### 7.5.2 多于两组的比较

Kruskal - Wallis 检验（独立）  Friedman 检验（不独立）	如果需要比较的组数多于两个，且无法满足 ANOVA 设计的假设，那么可以使用非参数方法来评估组间的差异  如果各组独立，则可以使用_____检验  如果各组不独立（如重复测量设计或随机区组设计），则可以使用_____检验
---	--

<code>kruskal.test(y ~ A, data)</code> <b>y</b> 是一个数值型结果变量 <b>A</b> 是一个拥有两个或更多水平的分组变量	Kruskal-Wallis 检验的语法式为_____
Wilcoxon 秩和检验	若有 A 有两个水平，则该函数与_____等价

<b>friedman.test(y ~ A   B, data)</b> <b>y</b> 是数值型结果变量 <b>A</b> 是一个分组变量 <b>B</b> 是一个用以认定匹配观测的区组变量 <sup>98</sup>	Friedman 检验的语法式为_____
---	-----------------------

<b>kruskal.test(Illiteracy ~ state.region, data=states)</b>	已知: <b>states &lt;- data.frame(state.region, state.x77)</b>  针对 <b>states</b> 数据集, 做 Kruskal-Wallis 检验, 问不同地区 ( <b>state.region</b> ) 的文盲率 ( <b>Illiteracy</b> ) 是否不同?
哪些地区显著地与其他地区不同 <b>wmc()</b>  <b>wmc(y ~ A, data=, method=)</b> <b>y</b> 是数值输出变量 <b>A</b> 是分组变量 <b>data</b> 是包含这些变量的数据框 <b>method</b> 指定限制 I 类误差的方法	上诉检验只能回答不同地区的文盲率是否相同, 但不能回答 _____ 要实现这一点可以使用作者编写的函数_____可以实现这一目的, 它每次用 Wilcoxon 检验比较两组, 并通过 <b>p.adj()</b> 函数调整概率值 <sup>99</sup> 该函数的语法格式为_____, 解释其中的参数
<b>source("http://www.statmethods.net/RiA/wmc.txt")</b> <b>states &lt;- data.frame(state.region, state.x77)</b> <b>wmc(Illiteracy ~ state.region, data=states, method="holm")</b>	已知: 可以从 <a href="http://www.statmethods.net/RiA/wmc.txt">www.statmethods.net/RiA/wmc.txt</a> 上下载到一个包含 <b>wmc()</b> 函数的文本文件  <b>states &lt;- data.frame(state.region, state.x77)</b>  使用作者的自编函数, 针对 <b>states</b> 数据集, 用 Wilcoxon 秩和检验对不同地区 ( <b>state.region</b> ) 的文盲率 ( <b>Illiteracy</b> ) 做两两比较

## 7.6 组间差异的可视化

<sup>98</sup> 这里改编“4-3 随机区组设计的两因素方差分析”(北京大学, 何平平)的案例来解释 **friedman.test** 中的 **y**, **A**, **B**。将同品系, 同体重的 9 只小白鼠随机分到 **x**, **y**, **z** 三组中, 每组分别 3 只, 每组各只白鼠分别被喂 **a**, **b**, **c** 三种不同的营养液, 三周后各自的增重为:

区组 (即 <b>friedman.test()</b> 中的 <b>B</b> )	营养液 (即 <b>friedman.test()</b> 中的 <b>A</b> )	增重 (即 <b>friedman.test()</b> 中的 <b>y</b> )
x	a	49
x	b	50
x	c	48
y	a	50
y	b	51
y	c	45
z	a	48
z	b	49
z	c	46

问: 不同营养液喂养的白鼠, 其增重有没有区别? 就可以用 **friedman.test(y ~ A | B, data)** 来检验。

<sup>99</sup> 此处使用了第二版的方法。在第一版中, 作者建议使用 **npmc** 包中的 **npmc()** 函数来实现成对组间比较, 但是 R 的最近新版本已经不支持 **npmc** 包。于是在第二版中作者推荐了其自编的函数 **wmc()**。



## 第 8 章 回归

### 8.1 回归的多面性<sup>100</sup>

<sup>100</sup> 在这里（也许不合时宜，但很实用地）补充如何在 R 中获得回归结果后工整地导出结果至 Excel。

方法 1：例如，已知  $\text{fit} \leftarrow \text{lm}(y \sim x)$ 。

`library(stargazer)` # 载入 stargazer 包（此包中的 `stargazer()` 函数能够方便又工整地展现、导出回归结果，首次使用前需用 `install.packages("stargazer")` 安装包。）

`stargazer(fit, type="text")` # 工整地展示回归结果（注意系数下的括号中是标准误。):（这里展现一个例子。）

```
=====
Dependent variable:
-----
y
-----
x                2.650***
                  (0.343)

Constant         2.014***
                  (0.739)

-----
Observations      4,268
R2                0.014
Adjusted R2       0.014
Residual Std. Error 14.960 (df = 4266)
F Statistic       59.649*** (df = 1; 4266)
=====
```

Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

如果想要将结果保存成表格格式，并在 Word 中打开，可以用下述命令：`stargazer(fit, type = "text", no.space = TRUE, out = "fit.html")`。再在“Document”或“我的文档”中找到 `fit.html` 文件并用 Word 打开。

如果想在每个估计系数下展示 p 值而不是标准误，可以用下述命令：`stargazer(fit, type="text", report=('vc*p'))`。我们将得到：

```
=====
Dependent variable:
-----
y
-----
x                2.650***
                  p = 0.000

Constant         2.014***
                  p = 0.007

-----
Observations      4,268
R2                0.014
Adjusted R2       0.014
Residual Std. Error 14.960 (df = 4266)
F Statistic       59.649*** (df = 1; 4266)
=====
```

Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

如果想删除上表中的“p=”，可以在 Word 中查找-替代删除。

如果只想在最末尾报告样本量和  $R^2$ ，可以用：`stargazer(fit, type="text", keep.stat=c("n", "req"))`

如果想对结果中的数字保留 n 位小数，可以加入 `digits = n` 这一参数。

如果想一次导出多个回归结果（例如有 `fit1`, `fit2`, `fit3`）可以用 `stargazer(fit1, fit2, fit3, type="text")`

方法 2（笨办法，可忽略）：例如，已知  $a \leftarrow \text{lm}(y \sim x)$ ;  $b \leftarrow \text{summary}(a)$ 。使用 `broom` 包的 `tidy()` 函数整理回归结果 `b` 至一个数据框 `c` 中（`library(broom); c <- tidy(b)`）。再通过函数为数据框 `c` 的 `p.value` 列旁添加星号（例如，按照计量经济学的一般标准

```
c=within(c,{
  significance <- NA
  significance[p.value<=0.01] <- "****"
  significance[p.value>0.01 & p.value<=0.05] <- "***"
  significance[p.value>0.05 & p.value<=0.1] <- "**"
  significance[p.value>0.1] <- ""
})
```

## 8.2 OLS 回归

### 8.2.1 用 lm() 拟合回归模型

<code>lm()</code> <code>myfit &lt;- lm(formula, data)</code> <sup>101</sup>  <code>formula</code> 指要拟合的模型形式, 形式如下: $Y \sim X_1 + X_2 + \dots + X_k$ ~左边为响应变量, 右边为各个预测变量, 预测变量之间用+符号分隔  <code>data</code> 是一个数据框,	在 R 中, 拟合线性模型最基本的函数是_____  其语法格式为_____
~ $y \sim x + z + w$	_____表示分隔符号, 左边为响应变量, 右边为解释变量 要通过 $x$ 、 $z$ 和 $w$ 预测 $y$ , 代码为_____
+	分隔预测变量
: $y \sim x + z + x:z$	_____表示预测变量的交互项 要通过 $x$ 、 $z$ 及 $x$ 与 $z$ 的交互项预测 $y$ , 代码为_____
* $y \sim x * z * w$	_____表示所有可能交互项的简洁方式 代码_____可展开为 $y \sim x + z + w + x:z + x:w + z:w + x:z:w$
^ $y \sim (x + z + w)^2$	_____表示交互项达到某个次数 代码_____可展开为 $y \sim x + z + w + x:z + x:w + z:w$
. $y \sim .$	_____表示包含除因变量外的所有变量 若一个数据框包含变量 $x$ 、 $y$ 、 $z$ 和 $w$ , 代码_____可展开为 $y \sim x + z + w$
- $y \sim (x + z + w)^2 - x:w$	_____表示从等式中移除某个变量。 _____可展开为 $y \sim x + z + w + x:z + z:w$
- 1 $y \sim x - 1$	_____表示删除截距项 表达式_____拟合 $y$ 在 $x$ 上的回归, 并强制直线通过原点
I() $y \sim x + I((z + w)^2)$	从算术的角度来解释括号中的元素。

`})`  
`),`  
以及界定需要保留的小数点(例如, 若想保留 4 位小数可以令 `d <- round(c[,c(2:5)], digits = 4)`, 再 `cbind(c$term, d)`)。最后, 通过 `write.table(c, "sample.csv", sep = ",")` 导出至名为 `sample.csv` 的 Excel 可查看的文件。也可用 `view(c)` 把整理好的结果显示出来, 再复制进 Excel。  
一个偷懒的方法是, 直接把 `b <- summary(a)` 所显示的 `Coefficients` 项的内容复制粘贴至 Excel 中, 此时所有内容会呈现在一列中。再使用“数据” - “分列” - “固定列宽分列”把各项分开。

<sup>101</sup> 如果只想回归满足某个条件的子集, 可以用 `subset` 参数, 例如 `lm(a~b+c, subset=(d==1))`,  $a$ ,  $b$ ,  $c$ ,  $d$  均是变量名。

	例如, $y \sim x + (z + w)^2$ 将展开为 $y \sim x + z + w + z:w$ , 而代码_____将展开为 $y \sim x + h$ , $h$ 是一个由 $z$ 和 $w$ 的平方和创建的新变量
$\log(y) \sim x + z + w$	可以在表达式中用的数学函数。例如, _____表示通过 $x$ 、 $z$ 和 $w$ 来预测 $\log(y)$

	已知某次拟合线性模型的结果保存在 <code>myfit</code> 中, 针对 <code>myfit</code> , 应用函数以达到以下目的:
<code>summary()</code>	展示拟合模型的详细结果
<code>coefficients()</code>	列出拟合模型的模型参数 (截距项和斜率)
<code>confint()</code>	提供模型参数的置信区间 (默认 95%)
<code>fitted()</code>	列出拟合模型的预测值
<code>residuals()</code>	列出拟合模型的残差值
<code>anova()</code>	生成一个拟合模型的方差分析表, 或者比较两个或更多拟合模型的方差分析表
<code>vcov()</code>	列出模型参数的协方差矩阵
<code>AIC()</code>	输出赤池信息统计量
<code>plot()</code>	生成评价拟合模型的诊断图
<code>predict()</code>	用拟合模型对新的数据集预测响应变量值

### 8.2.2 简单线性回归

<code>fit &lt;- lm(weight ~ height, data=women)</code>	已知: 数据集 <code>women</code> 包含了 15 个年龄在 30~39 岁间女性的身高( <code>height</code> )和体重 ( <code>weight</code> ) 信息  针对 <code>women</code> 数据集, 做体重 ( <code>weight</code> ) 对身高 ( <code>height</code> ) 的回归, 并保存在 <code>fit</code> 中
<code>fitted(fit)</code>	展示回归 <code>fit</code> 的各观测的预测值, 即 $\hat{y}$
<code>residuals(fit)</code>	展示回归 <code>fit</code> 的各观测的残差, 即 $\hat{\varepsilon} = y - \hat{y}$
<code>plot(women\$height, women\$weight)</code> <code>abline(fit)</code>	针对 <code>women</code> 数据集, 绘制横轴变量为体重 ( <code>weight</code> ), 纵轴变量为身高 ( <code>height</code> ) 的散点图; 把回归拟合线添加进上图中

### 8.2.3 多项式回归

<code>fit2 &lt;- lm(weight ~ height + I(height^2), data=women)</code>	针对 women 数据集，做 weight 对 height 和 height 平方的回归，并保存在 fit2 中
<code>plot(women\$height, women\$weight)</code>	绘制横轴为体重（weight），纵轴为身高（height）的散点图
<code>lines(women\$height, fitted(fit2))</code>	为上图添加 fit2 的回归曲线

car 包的 <code>scatterplot()</code> 函数  <pre>library(car) scatterplot(weight ~ height,              data=women,              spread=FALSE,              lty.smooth=2,              pch=19)</pre>	使用_____包的_____函数绘制 weight 与 height 的散点图、线性拟合曲线、平滑拟合曲线、并在相应的边界展示每个变量的箱线图。（删除残差正负均方根在平滑曲线上的展开和非对称信息；平滑拟合曲线为虚线；点为实心圆）
--	--

### 8.2.4 多元线性回归

<code>cor()</code>	_____函数提供了二变量之间的相关系数
car 包中 <code>scatterplotMatrix()</code> 函数	_____包中_____函数默认在非对角线区域绘制变量间的散点图，并添加平滑和线性拟合曲线，在对角线区域绘制每个变量的密度图和轴须图
<pre>cor(states) library(car) scatterplotMatrix(states, spread=FALSE,                   lty.smooth=2)</pre>	已知：  <pre>states &lt;- as.data.frame(state.x77[,c("Murder", "Population",                                      "Illiteracy", "Income", "Frost")])</pre> 展现 states 数据集中各变量间的相关系数； 绘制因变量与自变量的散点图矩阵。（包含线性和平滑拟合曲线，以及相应的边际分布（核密度图和轴须图））

<pre>fit &lt;- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states)</pre>	已知：  <pre>states &lt;- as.data.frame(state.x77[,c("Murder", "Population",                                      "Illiteracy", "Income", "Frost")])</pre> 针对 states 数据集，做 Murder 对 Population, Illiteracy, Income, Frost 的回归，保存在 fit 中
---	---

## 8.2.5 有交互项的多元线性回归

<code>fit &lt;- lm(mpg ~ hp + wt + hp:wt, data=mtcars)</code>	针对 mtcars 数据集，做每加仑行驶的英里数 (mpg) 对 hp (马力)，车重 (wt)，以及 hp (马力) 和车重 (wt) 交互项的回归，保存在 fit 中
因变量与其中一个自变量的关系依赖于另外一个自变量的水平	若两个自变量的交互项显著说明_____
每加仑汽油行驶英里数 (mpg) 与汽车马力 (hp) 的关系依车重 (wt) 不同而不同	就上例而言，hp (马力) 和车重 (wt) 的交互项显著说明_____

<code>effects</code> 包中的 <code>effect()</code> 函数  <code>library(effects)</code> <code>plot(effect("hp:wt", fit,</code> <code>      list(wt=c(2.2, 3.2, 4.2))),</code> <code>      multiline=TRUE)</code>	已知：  wt 的均值为 3.2，少于均值一个标准差和多于均值一个标准差的值分别为 2.2 和 4.2，因此令 <code>wt &lt;- c(2.2, 3.2, 4.2)</code>  通过_____包的_____函数以图形的形式展示 fit 中 hp 与 wt 的交互项
--	---

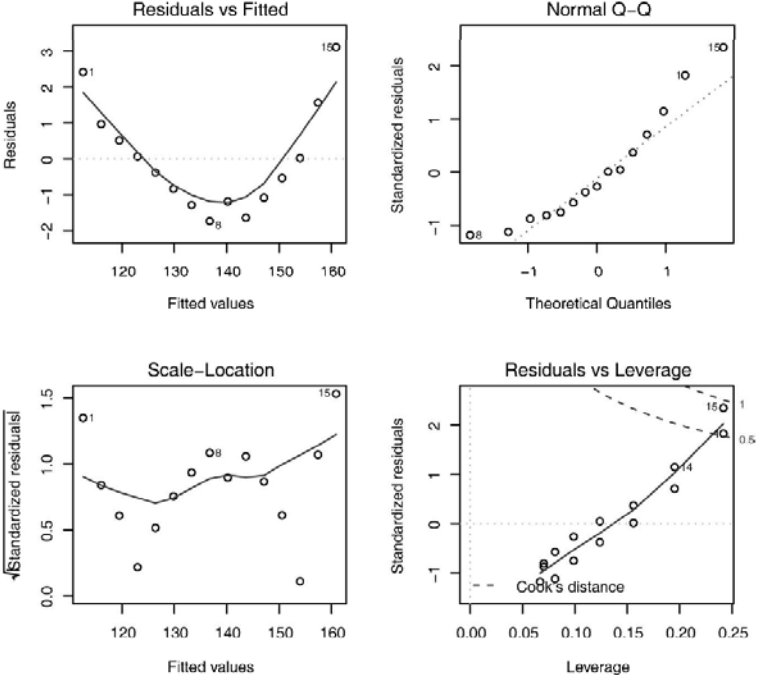
## 8.3 回归诊断

### 8.3.1 标准方法

<code>plot()</code> 函数	R 检验回归分析中统计假设的常见方法之一是对 <code>lm()</code> 函数返回的对象使用_____函数以可以生成评价模型拟合情况的四幅图形
------------------------	---

<code>par(mfrow=c(2, 2))</code> <code>plot(fit)</code>	已知：  <code>fit &lt;- lm(weight ~ height, data=women)</code>  检验 fit 中的统计假设  将四幅图组合在 2*2 的图形矩阵中
---	--



<p>残差值与拟合值的关联情况； 需要对回归模型增加一个二次项； <b>Residual</b> 和 <b>fit</b> 无关</p>	 <p>Residuals VS Fitted（左上图）描述了_____的情况； 若图中有一个明显的曲线关系，这可能暗示_____； 该图的理想状态是_____</p>
<p>在正态分布对应的值下，标准化残差的概率图； 图上的点应该落在呈 45 度角的直线上（即残差值满足是一个均值为 0 的正态分布。）</p>	<p>Normal Q-Q（右上图）描述了_____； 该图的理想状况是_____</p>
<p>同方差性 水平线周围的点应该随机分布</p>	<p>Scale-Location（左下图）描述了_____； 若满足同方差假设，那么在图中_____；</p>
<p>离群点，高杠杆值点和强影响点<sup>102</sup></p>	<p>Residuals VS Leverage（右下图）可以鉴别出_____</p>

<pre>newfit &lt;- lm(weight~ height + I(height^2), data=women[-c(13, 15),])</pre>	<p>针对数据集 <b>women</b>，做做 <b>weight</b> 对 <b>height</b> 和 <b>height</b> 平方的回归， 删除观测点 13 和 15，并保存在 <b>newfit</b> 中</p>
---	--

### 8.3.2 改进的方法

#### 1. 正态性

<p><b>car</b> 包中的 <b>qqPlot()</b> 函数</p>	<p>与基础包中的 <b>plot()</b> 函数相比，_____包的_____函数提供了更为精确的正态假设检验方法，它画出了在 <math>n - p - 1</math> 个自由度的 <b>t</b> 分布下的学生化残差图形，其中 <b>n</b> 是样本大小，<b>p</b> 是回归参数的数目（包括截距项）</p>
<pre>qqPlot(fit, labels=row.names(states), id.method="identify", simulate=TRUE, main</pre>	<p>已知：</p>

<sup>102</sup> 这三类点在 8.4 节中还有更详细的介绍。

="Q-Q Plot")	<pre>states &lt;- as.data.frame(state.x77[,c("Murder", "Population", "illiteracy", "Income", "Frost")])  fit &lt;- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states)</pre> <p>使用上述函数为 fit 做正态性检验（点的标签使用 states 数据集的行名称；使用交互式绘图；用参数自助法生成 95%的置信区间；图名称为 Q-Q Plot）</p>
所有的点都离直线很近，并都落在置信区间内，表明正态性假设符合得很好	上图的理想状态为_____

fitted(fit) ["Nevada"]	获得上述回归中 Nevada 观测的拟合值
residuals(fit) ["Nevada"]	获得上述回归中 Nevada 观测的残差
rstudent(fit) ["Nevada"]	获得上述回归中 Nevada 观测的学生化残差

	自制学生化残差函数：
residplot <- function(fit, nbreaks=10) {	函数名为 residplot(), 参数为 fit 和 nbreaks（默认值为 10）
z <- rstudent(fit)	令 z 为 fit 的学生化残差
hist(z, breaks=nbreaks, freq=FALSE)	绘制 z 的直方图，共分成 nbreaks 组，根据概率密度而不是频数绘制图形
rug(jitter(z), col="brown")	绘制 z 的轴须图，把 z 打散，颜色为棕色
curve(dnorm(x, mean=mean(z), sd=sd(z)), add=TRUE, col="blue", lwd=2) <sup>103</sup>	添加均值为 z 的均值，标准差为 z 的标准差的正态密度曲线，颜色为蓝色，2 倍标准宽度
lines(density(z)\$x, density(z)\$y, col="red", lwd=2, lty=2)	添加 z 的核密度图，颜色为红色，2 倍标准宽度，线型为 2
legend("topright", legend = c("NC", "KDC"), lty=1:2, col=c("blue", "red"), cex=.7) }	在图形右上角添加图例，标签分别为 NC 和 KDC，线型分别为 1 号和 2 号线型，颜色分别为蓝色和红色，字号为标准字号的 0.7 倍
residplot(fit)	对 fit 使用上述函数

## 2. 误差的独立性

<sup>103</sup> curve()函数用于绘制函数对应的曲线，确定函数的表达式、起始和终止的坐标就能绘制该区间的函数图形。例如，curve(sin,-2\*pi,2\*pi)绘制了正弦函数在-2π到 2π间的图形。又例如 curve(expr=2\*x+1, from=2, to=6)绘制了函数 y=2x+1 在 x=2 到 x=6 区间上的图形。

car 包的 <code>durbinWatsonTest()</code> 函数	_____包的_____函数可做 Durbin-Watson 检验，能够检测误差的序列相关性
<code>durbinWatsonTest(fit)</code>	针对多元回归 fit，使用上述函数做 Durbin-Watson 检验

### 3. 线性

car 包中的 <code>crPlots()</code> 函数	_____包的_____函数可以检验因变量与自变量之间是否存在非线性关系
<code>library(car)</code> <code>crPlots(fit)</code>	对针对多元回归 fit 做上述检验
若图形存在非线性，则说明可能对预测变量的函数形式建模不够充分，那么就需要添加一些曲线成分，比如多项式项，或对一个或多个变量进行变换（如用 $\log(X)$ 代替 $X$ ），或用其他回归变体形式而不是线性回归。	如何解释检验结果？

### 4. 同方差性

<code>ncvTest()</code> <sup>104</sup> 存在异方差性（误差方差不恒定）	car 包的_____函数生成一个计分检验，零假设为误差方差不变，备择假设为误差方差随着拟合值水平的变化而变化。若检验显著，则说明_____
<code>library(car)</code> <code>ncvTest(fit)</code>  计分检验不显著（ $p=0.19$ ），说明满足同方差假设	对 fit 做上述检验； 若得到 $p=0.19$ 说明_____

<code>spreadLevelPlot()</code>	car 包的_____函数创建一个添加了最佳拟合曲线的散点图，展示标准化残差绝对值与拟合值的关系
在最佳拟合曲线周围水平随机分布； 将会出现一个非水平的曲线	在满足同方差性时，该图的点_____，若不满足该假设_____

### 8.3.3 线性模型假设的综合验证

gvlma 包中的 <code>gvlma()</code> 函数	_____包中的_____函数能对线性模型假设进行综合验证，同时还能做偏斜度、峰度和异方差性的评价
-----------------------------------	---

### 8.3.4 多重共线性

置信区间	多重共线性会导致模型参数的_____过大，使单个系数解释起来很困难
car 包中的 <code>vif()</code> 函数	_____包中的_____函数能提供检验多重共线性的方差膨胀因

<sup>104</sup> `ncvTest()` 实质上做的是布伦斯-帕甘(Breusch - Pagan)检验。另外一种常见的异方差检验怀特 (White) 检验可以参考“计量经济学上机实验——异方差检验”（山东理工大学经济学院，王金田）：<https://wenku.baidu.com/view/f7158e620b1c59eef8c7b43f.html>。

$\sqrt{vif} > 2$	子 VIF (VIF 的平方根表示变量回归参数的置信区间能膨胀为与模型无关的预测变量的程度 <sup>105</sup> ；_____表明存在多重共线性问题)
------------------	---

## 8.4 异常观测值

### 8.4.1 离群点

那些模型预测效果不佳的观测点。 它们通常有很大的、或正或负的残差 ( $Y_i - \hat{Y}_i$ )。 正的残差说明模型低估了响应值, 负的残差则说明高估了响应值。	离群点是指_____
car 包的 outlierTest() 函数	_____包的 _____函数可以求得最大标准化残差绝对值 Bonferroni 调整后的 p 值
该函数根据单个最大 (或正或负) 残差值的显著性来判断是否有离群点 若不显著, 则说明 <u>数据集中没有离群点</u> ; 若显著, 则 <u>必须删除该离群点</u> , 然后再检验是否还有其他离群点存在	上述函数的评判标准是_____; 若不显著, 则说明_____; 若显著, 则_____
library(car) outlierTest(fit) <sup>106</sup>	对 fit 做离群点检验

### 8.4.2 高杠杆值点

高杠杆值观测点, 即与其他自变量有关的离群点。换句话说, 它们是由许多异常的预测变量值组合起来的, 与响应变量值没有关系。(高杠杆值点是自变量因子空间中的离群点, 由于其附近没有其他观测点, 拟合直线会向这些离群点偏移)	高杠杆值点是指_____
高杠杆值点可能是也可能不是强影响点, 这要看高杠杆值是否是离群点。	高杠杆值点是强影响点吗?
看该观测的帽子统计量有没有大于均值的 2 至 3 倍	如何判断高杠杆值点?
	原书第一版第 182 页, 第二版第 183 页的自制作图函数 hat.plot() 可以绘制高杠杆点

### 8.4.3 强影响点

强影响点是对模型参数估计值影响有些比例失衡的点, 若删除这些点, 回归模型的截距项和斜率会发生显著变化。	强影响点是指_____
--	-------------

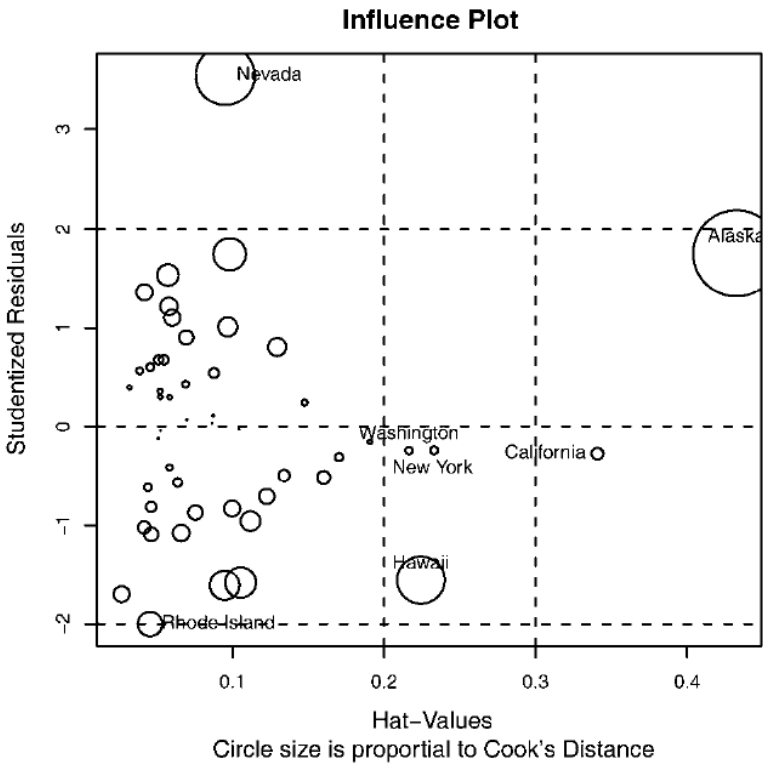
<sup>105</sup> 一个对  $\sqrt{vif}$  清晰的解释是: The square root of the VIF tells you how much larger the standar error (标准误) is, compared with what is would be if that variable were uncorelated with other independent variables in the equition.

<sup>106</sup> outlierTest()报出的结果中有两个 p 值, 一个是 p-value, 另一个是 Bonferonni P, 判断离群点基于后者是否小于 0.05, 若 Bonferonni P 小于 0.05 说明是离群点。例如本例中 Nevada 的 Bonferonni P 是 0.048, 我们判断其是离群点。采用命令 fit1 <- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states[-c(28), ])做删除 Nevada 样本后的回归, 再做 outlierTest(fit1)会返回结果“No Studentized residuals with Bonferonni P<0.05”, 说明不再有离群点。

在删除观测时需要知道观测的顺序号, 当观测样本很多时, 如何确定某个观测的顺序号呢? 例如如何获知本例中 Nevada 是第 28 个观测呢? 我的方法如下: (1) length(rownames(states))返回 50, 可知一共有 50 个观测。(2) b <- seq(1,50,1), 生成各观测的顺序号。(3) states1 <- cbind(states, b), 把各观测的顺序号加在原数据集的右侧。(4) view(states1), 调出修改数据集的视图框。(5) 在视图框的查询栏 (右上角) 中输入 “Nevada”, 查看 b 列为 28。

Cook 距离（或称 D 统计量）和变量添加图	检测强影响点的两种方法是_____和_____
Cook's D 值大于 $4/(n-k-1)$ ，则表明它是强影响点，其中 n 为样本量大小，k 是预测变量数目	用 Cook's D 值判断强影响点的标准是_____
	原书第一版第 183 页，第二版第 184 页的图形命令可绘制 Cook's D 值

car 包中的 avPlots() 函数	_____包的_____函数可提供变量添加图
library(car) avPlots(fit, ask=FALSE, id.method="identify")	用上述函数做 fit 的变量添加图（图形一次生成一个，用户可以通过单击点来判断强影响点）

car 包中的 influencePlot() 函数	_____包的_____函数可将离群点、高杠杆值点、强影响点的信息都整合在一张图中
library(car) influencePlot(fit, id.method="identify")	用上述函数将 fit 的离群点、高杠杆值点、强影响点的信息都整合在一张图中（使用交互式作图）
纵坐标大于+2 或者小于-2 的点	 <p>上图中_____被认为是离群点</p>
横坐标超过 0.2 或 0.3 的点	上图中_____被认为是高杠杆点
圆圈很大的点	上图中_____被认为是强影响点

## 8.5 改进措施

### 8.5.1 删除观测点

	<p>删除离群点通常可以提高数据集对于正态假设的拟合度，而强影响点会干扰结果，通常也会被删除。</p> <p>删除最大的离群点或者强影响点后，模型需要重新拟合。</p> <p>若离群点或强影响点仍然存在，重复以上过程直至获得比较满意的拟合。</p>
--	--

### 8.5.2 变量变换

一个或多个变量的变换	当模型不符合正态性、线性或者同方差性假设时，_____通常可以改善或调整模型效果
$Y^\lambda$	变换多用_____替代 $Y$
logit 变换 $[\ln(Y/1-Y)]$	若 $Y$ 是比例数，通常使用_____

因变量	当模型违反正态假设时，通常可以尝试对_____进行某种变换
car 包中的 powerTransform() 函数	_____包中的_____函数通过 $\lambda$ 的最大似然估计来正态化变量 $X^\lambda$
<pre>library(car)</pre> <pre>summary(powerTransform(states\$Murder))</pre> <p>Est. Power 一栏的 0.6 用来正态化变量 Murder 的 <math>\lambda</math>，即 <math>\text{Murder}^\lambda</math></p> <p>同时也应该关注 LR test, lambda=(1) 这一栏，如果 <math>\lambda=1</math> 的假设无法拒绝（pval 值&gt;0.1）则没有有力证据表明需要进行变量变换</p>	<p>已知：</p> <pre>states &lt;- as.data.frame(state.x77[,c("Murder", "Population", "Illiteracy", "Income", "Frost")])</pre> <p>对 states 数据集的 Murder 变量做 Box-Cox 正态变换，并对结果进行解释</p> <pre>bcPower Transformation to Normality       Est.Power Std.Err. Wald Lower Bound Wald Upper Bound states\$Murder    0.6    0.26      0.088      1.1  Likelihood ratio tests about transformation parameters               LRT df pval LR test, lambda=(0) 5.7 1 0.017 LR test, lambda=(1) 2.1 1 0.145</pre>

自变量	当模型违反线性假设时，对_____进行变换常常会比较有用
car 包中的 boxTidwell() 函数	_____包中的_____函数通过获得预测变量幂数的最大似然估计来改善线性关系
<pre>library(car)</pre> <pre>boxTidwell(Murder~Population+Illiteracy, data=states)</pre>	<p>已知回归模型为 <math>\text{Murder} \sim \text{Population} + \text{Illiteracy}</math>，数据来自 states 数据集，采用上述函数对上述模型进行 Box-Tidwell 变换，并对结果</p>

lambda 一栏提供了用于变换自变量的 $\lambda$ ，但同时需要关注 p-value，当 p 值较大时表明没有理由进行变换	进行解释				
		Score	Statistic	p-value	MLE of lambda
	Population	-0.32	0.75	0.87	
	Illiteracy	0.62	0.54	1.36	

因变量	_____变换还能改善异方差性（误差方差非恒定）
<p><b>car</b> 包中 <b>spreadLevelPlot()</b> 函数</p> <p>代码结果建议幂次变换（<b>suggested power transformation</b>）的含义是，经过 <b>p</b> 次幂（<math>Y^p</math>）变换，非恒定的误差方差将会平稳。对于当前例子（<b>fit</b>），异方差性很不明显，因此建议幂次接近 <b>1</b>（<b>1.2</b>，不需要进行变换）。</p>	<p>_____包中的_____函数提供了幂次变换的参考值</p> <p>解释其结果：</p> <pre>&gt; spreadLevelPlot(fit)</pre> <p>Suggested power transformation: 1.2</p>

## 8.6 选择“最佳”的回归模型

### 8.6.1 模型比较

<b>anova()</b> 函数	用基础安装中_____的可以比较两个嵌套模型的拟合优度 <sup>107</sup> ，所谓嵌套模型，即它的一些项完全包含在另一个模型中																					
<b>anova(fit2, fit1)</b>  若 <b>Pr(&gt;F)</b> 的值较大（ <b>p</b> 不显著，如本例中 <b>p=0.994</b> ），则不需要将两个变量添加到线性模型中	已知：  fit1 <- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states)  fit2 <- lm(Murder ~ Population + Illiteracy, data=states)  用上述方法比较 fit1 和 fit2，并解释结果  Analysis of Variance Table  Model 1: Murder ~ Population + Illiteracy Model 2: Murder ~ Population + Illiteracy + Income + Frost <table><thead><tr><th></th><th>Res.Df</th><th>RSS</th><th>Df</th><th>Sum of Sq</th><th>F</th><th>Pr(&gt;F)</th></tr></thead><tbody><tr><td>1</td><td>47</td><td>289.246</td><td></td><td></td><td></td><td></td></tr><tr><td>2</td><td>45</td><td>289.167</td><td>2</td><td>0.079</td><td>0.0061</td><td>0.994</td></tr></tbody></table>		Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)	1	47	289.246					2	45	289.167	2	0.079	0.0061	0.994
	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)																
1	47	289.246																				
2	45	289.167	2	0.079	0.0061	0.994																

<p>模型的统计拟合度以及用来拟合的参数数目小</p> <p>模型用较少的参数获得了足够的拟合度</p> <p><b>AIC()</b></p>	<p><b>AIC</b>（Akaike Information Criterion，赤池信息准则）也可以用来比较模型，它考虑了_____</p> <p><b>AIC</b> 值较_____的模型要优先选择，它说明_____</p> <p>该准则可用_____函数实现</p>
<b>AIC(fit1, fit2)</b>	已知：

<sup>107</sup> 一个比较低的拟合优度并不意味着所使用的 OLS 回归方程没有用。在很多公司金融实证研究中，重点关注 **x** 对 **y** 的影响的方向，大小和显著度，对  $R^2$  并不在意。注意不能以  $R^2$  为标准评判是否应该增加一个变量， $R^2$  总会随着变量数目的增加而增加。

此处 <code>fit1</code> 的变量个数多于 <code>fit2</code> ，而前者的 AIC 大于后者，因此 <code>fit2</code> 更佳 <sup>108</sup>	<pre>fit1 &lt;- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states)  fit2 &lt;- lm(Murder ~ Population + Illiteracy, data=states)</pre> <p>用 AIC 来比较 <code>fit1</code> 和 <code>fit2</code>，并解释结果</p> <table><thead><tr><th></th><th>df</th><th>AIC</th></tr></thead><tbody><tr><td><code>fit1</code></td><td>6</td><td>241.6429</td></tr><tr><td><code>fit2</code></td><td>4</td><td>237.6565</td></tr></tbody></table>		df	AIC	<code>fit1</code>	6	241.6429	<code>fit2</code>	4	237.6565
	df	AIC								
<code>fit1</code>	6	241.6429								
<code>fit2</code>	4	237.6565								
嵌套模型	ANOVA 需要_____，而 AIC 方法不需要。									

## 8.6.2 变量选择

逐步回归法和全子集回归	从大量候选变量中选择最终的预测变量有两种流行的方法：_____和_____
-------------	---------------------------------------

### 1. 逐步回归

每次添加一个预测变量到模型中，直到添加变量不会使模型有所改进为止	向前逐步回归是指_____
从模型包含所有预测变量开始，一次删除一个变量直到会降低模型质量为止	向后逐步回归是指_____
结合了向前逐步回归和向后逐步回归的方法，变量每次进入一个，但是每一步中，变量都会被重新评价，对模型没有贡献的变量将会被删除，预测变量可能会被添加、删除好几次，直到获得最优模型为止	向前向后逐步回归是指_____
MASS 包中的 <code>stepAIC()</code> 函数	_____包中的_____函数可以实现 <u>逐步回归模型</u> （向前、向后和向前向后） <sup>109</sup> ，依据的是精确 AIC 准则。
<pre>library(MASS) stepAIC(fit, direction="backward")</pre>	<p>已知：</p> <pre>states &lt;- as.data.frame(state.x77[,c("Murder", "Population", "Illiteracy", "Income", "Frost")])  fit &lt;- lm(Murder ~ Population + Illiteracy + Income + Frost, data=states)</pre> <p>使用上述函数对 <code>fit</code> 做向后逐步回归</p>
不是每一个可能的模型都被评价了	逐步回归法虽然它可能会找到一个好的模型，但是不能保证模型就是最佳模型，因为_____。为克服这个限制，便有了全子集回归法。

<sup>108</sup> 赤池信息准则要求仅当所增加的解释变量能够减少 AIC 值时才在原模型中增加该解释变量。

<sup>109</sup> 所对应的 `direction=` “ ” 参数分别为 `forward`（向前）、`backward`（向后）和 `both`（向前向后）。



## 2. 全子集回归

所有可能的模型都会被检验	全子集回归是指_____
<code>leaps</code> 包中的 <code>regsubsets()</code> 函数	全子集回归可用_____包中的_____函数实现, 该函数能通过 $R$ 平方、调整 $R$ 平方或 Mallows $C_p$ 统计量等准则来选择“最佳”模型
<code>leaps</code> 包中的 <code>plot()</code> 函数 <code>car</code> 包中的 <code>subsets()</code> 函数	上述函数的结果可用_____包中的_____函数绘制, 或者用_____包中的_____函数绘制
<code>library(leaps)</code> <code>leaps &lt;- regsubsets(Murder ~ Population + Illiteracy + Income + Frost, data=states, nbest=4)</code>	已知:  <code>states &lt;- as.data.frame(state.x77[,c("Murder", "Population", "Illiteracy", "Income", "Frost")])</code>  针对 <code>states</code> 数据集, 使用上述函数做 <code>Murder</code> 对 <code>Population</code> , <code>Illiteracy</code> , <code>Income</code> , <code>Frost</code> 回归的全子集回归
方法一: <code>plot(leaps, scale="adj r2")</code> 方法二: <code>library(car)</code> <code>subsets(leaps, statistic="cp")</code> <code>abline(1, 1)</code>	用两种方法绘制全子集回归结果 <code>leaps</code> 的图形, 第一种方法要求基于调整 $R$ 平方, 第二种方法要求基于 Mallows $C_p$ 统计量

## 8.7 深层次分析

### 8.7.1 交叉验证

将一定比例的数据挑选出来作为训练样本, 另外的样本作保留样本, 先在训练样本上获取回归方程, 然后在保留样本上做预测。	所谓交叉验证, 即_____
	在 $k$ 重交叉验证中, 样本被分为 $k$ 个子样本, 轮流将 $k - 1$ 个子样本组合作为训练集, 另外 $1$ 个子样本作为保留集。这样会获得 $k$ 个预测方程, 记录 $k$ 个保留样本的预测表现结果, 然后求其平均值
	第一版第 193 页, 第二版第 194 页的自编函数 <code>shrinkage()</code> 可做 $R$ 平方统计量的 $k$ 重交叉验证
减少得越少	泛化能力越好的模型, 使用 <code>shrinkage()</code> 对其进行交叉验证后, 其 $R$ 平方统计量_____

### 8.7.2 相对重要性

<p><code>scale()</code><sup>110</sup></p>	<p>比较自变量相对重要性的方法之一是比较标准化的回归系数，在进行回归分析前可用_____函数将数据标准化为均值为 0，标准差为 1 的数据集</p>
<pre>a &lt;- scale(states) zstates &lt;- as.data.frame(a) zfit &lt;- lm(Murder~Population + Income + Illiteracy + Frost, data=zstates) coef(zfit)</pre> <p>此处可以看到，当其他因素不变时，文盲率一个标准差的变化将增加 <b>0.68</b> 个标准差的谋杀率。</p> <p>根据标准化的回归系数，我们可认为 <b>Illiteracy</b> 是最重要的预测变量，而 <b>Frost</b> 是最不重要的。</p>	<p>已知：</p> <pre>states &lt;- as.data.frame(state.x77[,c("Murder", "Population", "Illiteracy", "Income", "Frost")])</pre> <p>针对 <code>states</code> 数据集，获得 <code>Murder</code> 对 <code>Population</code>、<code>Income</code>、<code>Illiteracy</code>、<code>Frost</code> 回归的标准化回归系数</p> <p>将 <code>states</code> 标准化，保存在 <code>a</code> 中；</p> <p>将 <code>a</code> 转化为数据框，保存在 <code>zstates</code> 中；</p> <p>做 <code>Murder</code> 对 <code>Population</code>、<code>Income</code>、<code>Illiteracy</code>、<code>Frost</code> 的回归，保存在 <code>zfit</code> 中；</p> <p>展示 <code>zfit</code> 的回归系数，并解释所得到的结果</p> <pre>(Intercept)    Population      Income    illiteracy      Frost -2.054026e-16  2.705095e-01  1.072372e-02  6.840496e-01  8.185407e-03</pre>
	<p>第一版第 195 页，第二版第 196 页的自编函数 <code>relweights()</code> 函数可以得到每个因变量对 <code>R</code> 平方的贡献</p>

<sup>110</sup> 注意 `scale()` 函数返回的是一个矩阵，而 `lm()` 针对的数据格式是数据框，因而在计算标准化的回归系数前需要把数据类型转化为数据框。

第 9 章 方差分析

9.1 术语速成

<p>单因素方差分析（单因素组间方差分析）</p> <p>治疗方案</p> <p>STAI 是因变量</p> <p>两种治疗方法（CBT、EMDR）是自变量</p> <p>由于在每种治疗方案下观测数相等，因此这种设计也称为均衡设计；若观测数不同，则称作非均衡设计。</p>	<p>已知：</p> <p>以焦虑症治疗为例，现有两种治疗方案：认知行为疗法（CBT）和眼动脱敏再加工法（EMDR）。招募 10 位焦虑症患者作为志愿者，随机分配一半的人接受为期五周的 CBT，另外一半接受为期五周的 EMDR，在治疗结束时，要求每位患者都填写状态特质焦虑问卷（STAI）</p> <table><tr><th colspan="2">疗法</th></tr><tr><th>CBT</th><th>EMDR</th></tr><tr><td>s1</td><td>s6</td></tr><tr><td>s2</td><td>s7</td></tr><tr><td>s3</td><td>s8</td></tr><tr><td>s4</td><td>s9</td></tr><tr><td>s5</td><td>s10</td></tr></table> <p>该方差分析的类型是_____</p> <p>_____是两个水平（CBT、EMDR）的组间因子</p> <p>_____是因变量， _____是自变量</p> <p>什么是均衡设计和非均衡设计？</p>	疗法		CBT	EMDR	s1	s6	s2	s7	s3	s8	s4	s9	s5	s10																					
疗法																																				
CBT	EMDR																																			
s1	s6																																			
s2	s7																																			
s3	s8																																			
s4	s9																																			
s5	s10																																			
<p>单因素组内方差分析</p> <p>作重复测量方差分析</p> <p>时间</p>	<p>已知：</p> <p>只对 CBT 的效果感兴趣，则需将 10 个患者都放在 CBT 组中，然后在治疗五周和六个月后分别评价疗效</p> <table><tr><th rowspan="2">患者</th><th colspan="2">时间</th></tr><tr><th>5 周</th><th>6 个月</th></tr><tr><td>s1</td><td></td><td></td></tr><tr><td>s2</td><td></td><td></td></tr><tr><td>s3</td><td></td><td></td></tr><tr><td>s4</td><td></td><td></td></tr><tr><td>s5</td><td></td><td></td></tr><tr><td>s6</td><td></td><td></td></tr><tr><td>s7</td><td></td><td></td></tr><tr><td>s8</td><td></td><td></td></tr><tr><td>s9</td><td></td><td></td></tr><tr><td>s10</td><td></td><td></td></tr></table>	患者	时间		5 周	6 个月	s1			s2			s3			s4			s5			s6			s7			s8			s9			s10		
患者	时间																																			
	5 周	6 个月																																		
s1																																				
s2																																				
s3																																				
s4																																				
s5																																				
s6																																				
s7																																				
s8																																				
s9																																				
s10																																				

	<p>该方差分析的类型是_____，由于每个受试者都不止一次被测量，也称作_____</p> <p>_____是两水平（五周、六个月）的组内因子</p>																																										
<p>含组间和组内因子的双因素方差分析</p> <p>疗法和时间都作为因子时，我们既可分析疗法的影响（时间跨度上的平均）和时间的影响（疗法类型跨度上的平均），又可分析疗法和时间的交互影响。前两个称作主效应，交互部分称作交互效应。</p> <p>当设计包含两个甚至更多的因子时，便是因素方差分析设计，比如两因子时称作双因素方差分析，三因子时称作三因素方差分析，以此类推。若因子设计包括组内和组间因子，又称作混合模型方差分析，当前的例子就是典型的双因素混合模型方差分析。</p>	<p>已知：</p> <p>以焦虑症治疗为例，现有两种治疗方案：认知行为疗法（CBT）和眼动脱敏再加工法（EMDR）。招募 10 位焦虑症患者作为志愿者，随机分配五位患者到 CBT，另外五位到 EMDR，在五周和六个月后分别评价他们的 STAI 结果。</p> <table><tr><td colspan="2"></td><td rowspan="2">患者</td><td colspan="2">时间</td></tr><tr><td colspan="2"></td><td>5 周</td><td>6 个月</td></tr><tr><td rowspan="10">疗法</td><td rowspan="5">CBT</td><td>s1</td><td></td><td></td></tr><tr><td>s2</td><td></td><td></td></tr><tr><td>s3</td><td></td><td></td></tr><tr><td>s4</td><td></td><td></td></tr><tr><td>s5</td><td></td><td></td></tr><tr><td rowspan="5">EMDR</td><td>s6</td><td></td><td></td></tr><tr><td>s7</td><td></td><td></td></tr><tr><td>s8</td><td></td><td></td></tr><tr><td>s9</td><td></td><td></td></tr><tr><td>s10</td><td></td><td></td></tr></table> <p>该方差分析的类型是_____</p> <p>该实验中_____称作主效应，_____称作交互效应。</p> <p>什么是双因素方差分析、三因素方差分析、混合模型方差分析？</p>			患者	时间				5 周	6 个月	疗法	CBT	s1			s2			s3			s4			s5			EMDR	s6			s7			s8			s9			s10		
		患者	时间																																								
			5 周	6 个月																																							
疗法	CBT	s1																																									
		s2																																									
		s3																																									
		s4																																									
		s5																																									
	EMDR	s6																																									
		s7																																									
		s8																																									
		s9																																									
		s10																																									
<p>混淆因素，干扰变数</p> <p>协变量，协方差分析（ANCOVA）</p>	<p>抑郁症对病症治疗有影响，而且抑郁症和焦虑症常常同时出现。</p> <p>抑郁症也可以解释因变量的组间差异，因此它常称为_____；由于你对抑郁症不感兴趣，它也被称作_____</p> <p>假设用白氏抑郁症量表（BDI）评测患者的抑郁症，那么可以在评测疗法类型的影响前，对任何抑郁水平的组间差异进行统计性调整。本案例中，BDI 为_____，该设计为_____</p>																																										
<p>多元方差分析（MANOVA）</p> <p>多元协方差分析（MANCOVA）</p>	<p>当因变量不止一个时，设计被称作_____</p> <p>若协变量也存在，那么就叫_____</p>																																										

## 9.2 ANOVA 模型拟合

### 9.2.1 aov()函数

aov() aov(formula, data=dataframe)	ANOVA 分析的函数为_____, 其语法格式为_____
---------------------------------------	--------------------------------

	aov()表达式中的特殊符号:
~ 为 $y \sim A + B + C$	_____为分隔符号, 左边为响应变量, 右边为解释变量 用 A、B 和 C 预测 y, 代码为_____
: $y \sim A + B + A:B$	_____表示变量的交互项 用 A、B 和 A 与 B 的交互项来预测 y, 代码为_____
* $y \sim A + B + C + A:B + A:C + B:C + A:B:C$	_____表示所有可能交互项 代码 $y \sim A * B * C$ 可展开为_____
^ $y \sim A + B + C + A:B + A:C + B:C$	_____表示交互项达到某个次数 代码 $y \sim (A + B + C)^2$ 可展开为_____
. $y \sim A + B + C$	_____表示包含除因变量外的所有变量 若一个数据框包含变量 y、A、B 和 C, 代码 $y \sim .$ 可展开为_____

	常见研究设计的表达式: (用小写字母 (y, x, x1, x2) 表示定量变量, 大写字母 (A 或 B) 表示组别因子, subject 是对被试者独有的标识变量)
$y \sim A$	单因素 ANOVA
$y \sim x + A$	含单个协变量的单因素 ANCOVA
$y \sim A * B$	双因素 ANOVA
$y \sim x1 + x2 + A*B$	含两个协变量的双因素 ANCOVA
$y \sim B + A$ (B 是区组因子)	随机化区组
$y \sim A + \text{Error}(\text{Subject}/A)$	单因素组内 ANOVA <sup>111</sup>
$y \sim B * W + \text{Error}(\text{Subject}/W)$	含单个组内因子 (W) 和单个组间因子 (B) 的重复测量 ANOVA

### 9.2.2 表达式中各项的顺序

因子不止一个, 并且是非平衡设计 存在协变量	表达式中效应的顺序在两种情况下会造成影响: (a) _____ (b) _____
---------------------------	---

<sup>111</sup> Subject 是重复受试对象。例子参见第 9.6 节。

观测数	对于双因素方差分析，若不同处理方式中的_____不同，那么模型 $y \sim A*B$ 与模型 $y \sim B*A$ 的结果不同
A 对 y 的影响； 控制 A 时，B 对 y 的影响 控制 A 和 B 的主效应时，A 与 B 的交互效应	已知： $y \sim A + B + A:B$  R 中的 ANOVA 表的结果将评价_____
样本大小越不平衡	_____，效应项的顺序对结果的影响越大
放在表达式前面 协变量 主效应 双因素的交互项 三因素的交互项	一般来说，越基础性的效应越需要_____ 具体来讲，首先是_____，然后是_____，接着是_____，再接着是_____，以此类推
性别	对于主效应，越基础性的变量越应放在表达式前面，因此_____要放在处理方式之前

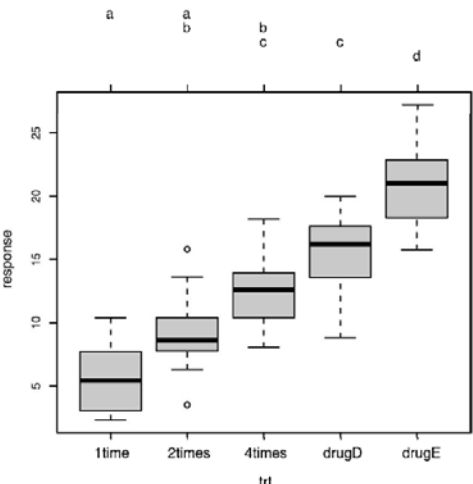
### 9.3 单因素方差分析

<pre>library(multcomp) attach(cholesterol) table(trt) aggregate(response, by=list(trt), FUN=mean) aggregate(response, by=list(trt), FUN=sd) fit &lt;- aov(response ~ trt) summary(fit)</pre> <p>五种治疗方法的效果显著不同</p>	<p>已知：</p> <p>multcomp 包中的 cholesterol 数据集包含了两个变量：trt（5 种治疗方法，5 种治疗方法分别对应 5 组实验对象）和 response（5 种治疗方法的实验对象的对应的治疗效果）</p> <p>对此进行单因素（组间）方差分析：</p> <p>绑定数据集 cholesterol；</p> <p>求各组样本的大小；</p> <p>求各组均值；</p> <p>求各组标准差；</p> <p>检验组间差异，保存在 fit 中；</p> <p>展示 fit 的结果；</p> <pre>       Df Sum Sq Mean Sq F value    Pr(&gt;F) trt      4 1351.4    337.8   32.43 9.82e-13 *** Residuals 45  468.8     10.4 --- Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1</pre> <p>若 <math>Pr(&gt;F)</math> 显著说明_____</p>
---	---

<code>gplots</code> 包中的 <code>plotmeans()</code>	_____包中的_____函数可以用来绘制带有置信区间的组均值图形
<code>library(gplots)</code> <code>plotmeans(response~trt)</code>	使用上述函数绘制 95%置信区间（默认）的各组疗法（trt）的疗效（response）均值

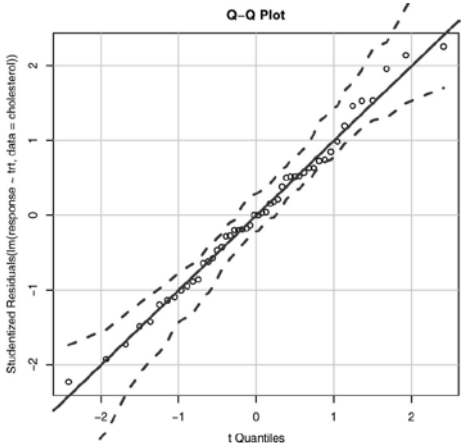
### 9.3.1 多重比较

<code>TukeyHSD()</code> 函数	_____函数提供了对各组均值差异的成对检验
<code>TukeyHSD(fit)</code>	使用上述函数对 <code>fit</code> 做成对检验
<code>p adj</code> 值超过 0.1	若上述检验中_____, 可认为该项的均值差异不显著
<code>par(las=2)</code> <code>par(mar=c(5, 8, 4, 2))</code> <code>plot(TukeyHSD(fit))</code>	绘制上述检验的图形: 旋转轴标签; 增加左边界面积至 8; 作图

<code>multcomp</code> 包中的 <code>glht()</code> 函数	_____包中的_____函数提供了多重均值比较更为全面的方法, 既适用于线性模型, 也适用于广义线性模型
<code>library(multcomp)</code> <code>par(mar=c(5, 4, 6, 2))</code> <code>tuk &lt;- glht(fit, linfct=mcp(trt="Tukey"))</code> <code>plot(cld(tuk, level=.05), col="lightgrey")</code>  有相同字母的组, 其均值差异不显著	用上述函数做 <code>fit</code> 的 Tukey HSD 成对检验, 并作图展示: 载入包; 增大上方边界至 6; 做 Tukey HSD 成对检验, 结果保存在 <code>tuk</code> 中; 作图, 显著性水平为 0.05 (即 95%的置信区间) 颜色为 <code>lightgrey</code> ;   解释上图 <sup>112</sup>

<sup>112</sup> 第一版第 206 页, 第二版第 206 页, 对该图进行解释时写到“也优于候选药物 drugD”, 这里应该是一处翻译错误, 原文为: “The competitor DrugD wasn’t superior th this four-time-per-day regimen”。

### 9.3.2 评估检验的假设条件

正态分布，各组方差相等	单因素方差分析中，我们假设因变量服从_____
Q-Q 图	可以使用_____来检验正态性假设
<pre>library(car) qqPlot(lm(response ~ trt, data=cholesterol), simulate=TRUE, las=1)</pre>	针对 cholesterol 数据集，使用 Q-Q 图对每组疗法（trt）的疗效（response）做正态性检验，使用 95%参数自助法，不添加点的标签，y 轴刻度标签竖直摆放（与 y 轴垂直） <sup>113</sup>
数据落在 95%的置信区间范围内，说明满足正态性假设。	 <p>如何解读上图？</p>

bartlett.test()	R 提供了一些可用来做方差齐性检验的函数，如做 Bartlett 检验的_____函数
bartlett.test(response ~ trt, data=cholesterol)	针对 cholesterol 数据集，使用上述函数检验每组疗法（trt）的疗效（response）的方差是否不同

<sup>113</sup> 原文的代码遗漏了 las=1，但原图显示出 y 轴刻度标签竖直摆放（与 y 轴垂直）。在这里：若 las=0，横轴和纵轴标签都平行于各自的标签轴，图 1。若 las=1，横轴标签平行于 x 轴，而纵轴标签垂直于 y 轴，图 2。

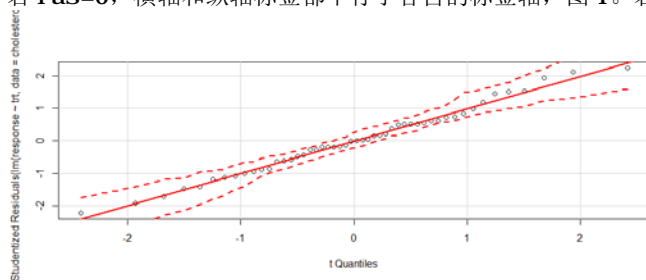


图 1

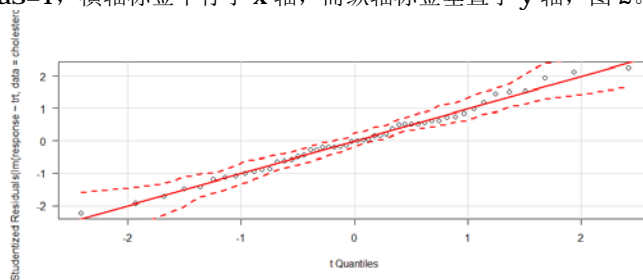


图 2

若 las=2，横轴和纵轴标签都垂直于各自的标签轴，图 3。若 las=3，横轴标签垂直于 x 轴，而纵轴标签平行于 y 轴，图 4。

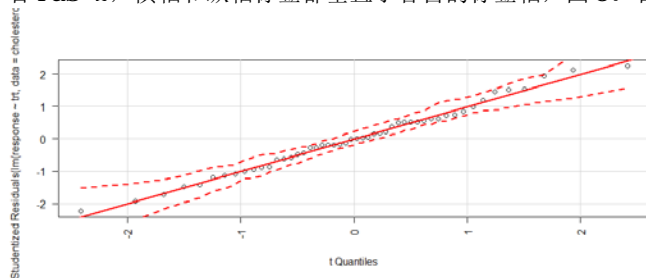


图 3

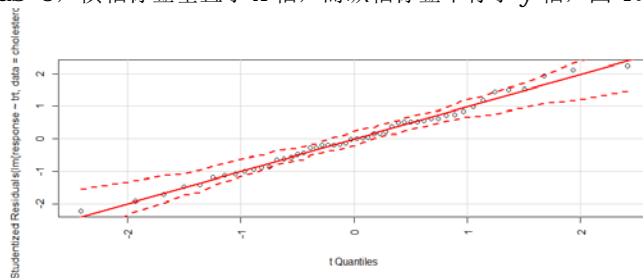


图 4



五组的方差并没有显著不同	<p><b>Bartlett test of homogeneity of variances</b></p> <p>data: response by trt Bartlett's K-squared = 0.57975, df = 4, p-value = 0.9653</p> <p>若上述检验的结果中 p-value 项不显著说明_____</p>
--------------	--

car 包中的 outlierTest() 函数	_____包中的_____函数可以用来检验离群点
library(car) outlierTest(fit)	使用上述函数对方差分析 fit 做离群点检验
没有离群点	<p>No Studentized residuals with Bonferonni p &lt; 0.05 Largest  rstudent : rstudent unadjusted p-value Bonferonni p 19 2.251149 0.029422 NA</p> <p>上述检验的结果中没有学生化残差的 Bonferonni P 小于 0.05, 说明 _____</p>

## 9.4 单因素协方差分析

	<p>已知:</p> <p>multcomp 包中的 litter 数据集含有四个变量: dose (4 组分别对应 4 个水平的药物剂量, 0、5、50 或 500); gesttime (妊娠时间, 协变量); number (一胎所生的幼鼠的数量); weight (一胎所生的幼鼠的平均体重)</p>
data(litter, package="multcomp")	载入 multcomp 包的 litter 数据集
attach(litter)	绑定数据集
table(dose)	计算每组 (dose) 样本的数量 <sup>114</sup>
aggregate(weight, by=list(dose), FUN=mean)	计算每组 (dose) 的幼鼠平均体重 (weight) 的均值
fit <- aov(weight ~ gesttime + dose)	以 weight 为因变量做单因素协方差分析, gesttime 为协变量, dose 为主效应, 将结果保存在 fit 中
summary(fit)	展示结果

effects 包中的 effects() 函数	可以使用_____包中的_____函数来计算调整的均值 (即去除协变量效应后的组均值)
library(effects) effect("dose", fit)	计算 fit 中不同组 (dose) 的去除协变量效应后 weight 的均值

<sup>114</sup> 原文 (包括英文版原文) 把 table(litter) 解释为 “每种剂量下所产的幼崽数”, 这是不正确的。这里应该是计算每组对应着多少个样本 (此处的一个样本是: 一只母鼠和她的一胎宝宝们)。若比较 “每种剂量下所产的幼崽数” 应该使用 aggregate(number, by = list(dose), FUN=mean)。

哪种处理方式与其他方式不同	和上一节的单因素 ANOVA 例子一样，剂量的 F 检验虽然表明了不同的处理方式幼崽的体重均值不同，但无法告知我们_____
<code>multcomp</code> 包	可以使用_____包来对所有均值进行成对比较，该包还可以用来检验用户自定义的均值假设。
<pre>library(multcomp) contrast &lt;- rbind("no drug vs. drug" = c (3, -1, -1, -1)) <sup>115</sup> summary(glht(fit, linfct=mcp(dose=contrast))) <sup>116</sup></pre>	使用上述包对 <code>fit</code> 做多重比较（设定第一组和其他三组的均值进行比较，命名为“no drug vs. drug”）

#### 9.4.1 评估检验的假设条件

<p>正态性和同方差性</p> <p>回归斜率相同</p> <p>怀孕时间（协变量）×剂量（主效应）的交互项不显著</p>	<p>ANCOVA 与 ANOVA 相同，都需要_____和_____假设，另外，ANCOVA 还假定_____；</p> <p>ANCOVA 模型包含_____时，可对回归斜率的同质性进行检验；交互效应若_____，则说明怀孕时间和幼崽出生体重间的关系依赖于药物剂量的水平</p>
<pre>library(multcomp) fit2 &lt;- aov(weight ~ gesttime*dose, data=litter) summary(fit2)</pre>	针对 <code>multcomp</code> 包的 <code>litter</code> 数据集，以检验回归斜率是否相同为目的做 <code>weight</code> 的协方差分析，因子项为 <code>gesttime</code> 和 <code>dose</code> ，结果保存在 <code>fit2</code> 中
<code>sm</code> 包中的 <code>sm.ancova()</code> 函数	不需要假设回归斜率同质性的非参数 ANCOVA 方法可用_____包中的_____函数实现

#### 9.4.2 结果可视化

<code>HH</code> 包中的 <code>ancova()</code> 函数	_____包中的_____函数可以绘制因变量、协变量和因子之间的关系图 <sup>117</sup>
<pre>library(HH) ancova(weight ~ gesttime + dose, data=litter)</pre>	绘制以 <code>weight</code> 为因变量做单因素协方差分析（ <code>gesttime</code> 为协变量， <code>dose</code> 为主效应）的图形
<code>ancova(weight ~ gesttime*dose)</code>	若用在上例中使用_____，生成的图形将允许斜率和截距项依据组别而发生变化，这对可视化那些违背回归斜率同质性的实例非常有用

<sup>115</sup> `rbind(a, b, c...)` 用于纵向合并矩阵或向量。经此操作后，`contrast` 是一个矩阵。

<sup>116</sup> 此处参考用户名为“dazzlingpuck”的同学在人大经济论坛的解答（<http://bbs.pinggu.org/thread-4167609-1-1.html>）。`dose=contrast` 表示以剂量分组进行多重比较，而多重比较函数 `mcp` 的参数 `contrast` 由之前的 `rbind` 给出。在 `rbind` 中，`3` 表示该组与其他 3 组对比，`-1` 表示该组为对照组，`0` 表示该组不参加比较。例如，如果只想比较第一组和第三组，可在 `rbind` 中用 `c(1, 0, -1, 0)` 表示。

<sup>117</sup> 同时也会给出方差分析的结果总结。

## 9.5 双因素方差分析

<pre>attach(ToothGrowth) table(supp, dose) aggregate(len, by=list(supp, dose), FUN=mean) aggregate(len, by=list(supp, dose), FUN=sd) dose &lt;- factor(dose) fit &lt;- aov(len ~ supp*dose) summary(fit)</pre>	<p>已知：</p> <p>在基础安装中的 <b>ToothGrowth</b> 数据集中有三个变量：豚鼠的牙齿长度（<b>len</b>），两种喂食方法（<b>supp</b>）（橙汁（<b>OJ</b>）或维生素 C（<b>VC</b>）），各喂食方法中抗坏血酸含量的水平（<b>dose</b>）（0.5mg、1mg 或 2mg）。共随机分配 60 只豚鼠，每种处理方式组合都被分配 10 只豚鼠。</p> <p>以牙齿长度（<b>len</b>）为因变量做双因素方差分析：</p> <p>绑定 <b>ToothGrowth</b> 数据集；</p> <p>展现双因子的交叉频数列联表；</p> <p>同时以 <b>supp</b> 和 <b>dose</b> 分组计算 <b>len</b> 的平均值；</p> <p>同时以 <b>supp</b> 和 <b>dose</b> 分组计算 <b>len</b> 的标准差；</p> <p>以牙齿长度（<b>len</b>）为因变量做双因素方差分析，主效应为 <b>supp</b> 和 <b>dose</b>，结果保存在 <b>fit</b> 中；</p> <p>展示结果</p>
--	--

<b>interaction.plot()</b> 函数	_____函数可以展示双因素方差分析的交互效应
<b>interaction.plot(dose, supp, len, type="b", col=c("red", "blue"), pch=c(16, 18))</b>	用上述函数绘制双因素方差分析 <b>fit</b> 的图形（图形类型为 <b>b</b> ，颜色为 <b>red</b> 和 <b>blue</b> ，点符号为 <b>16</b> 和 <b>18</b> ）

<b>gplots</b> 包中的 <b>plotmeans()</b> 函数	可用_____包中的_____函数来展示交互效应的图形，该图形包含了均值，误差棒（95%的置信区间）和样本大小
<pre>library(gplots) plotmeans(len ~ interaction(supp, dose, sep=" "), connect=list(c(1, 3, 5), c(2, 4, 6)), col=c("red", "darkgreen"))</pre>	用上述函数绘制以牙齿长度（ <b>len</b> ）为因变量做双因素方差分析的图形，主效应为 <b>supp</b> 和 <b>dose</b> ，图形从左至右第 1、第 3、第 5 个点连成一条线，第 2、第 4、第 6 连成一条线 <sup>118</sup> ，两条线的颜色分别为 <b>red</b> 和 <b>darkgreen</b>

<b>HH</b> 包中的 <b>interaction2wt()</b> 函数	用_____包中的_____函数来可视化方差分析的结果，图形对任意顺序的因子设计的主效应和交互效应都会进行展示
<pre>library(HH) interaction2wt(len~supp*dose)</pre>	用上述函数绘制以牙齿长度（ <b>len</b> ）为因变量做双因素方差分析的

<sup>118</sup> 即两种喂食方法（**supp**）中橙汁（**OJ**）的连成一条线，维生素 C（**VC**）的连成另外一条线。

图形，主效应为 **supp** 和 **dose**

## 9.6 重复测量方差分析

<pre> C02\$conc &lt;- factor(C02\$conc)  w1b1 &lt;- subset(C02, Treatment=='chilled')  fit &lt;- aov(uptake ~ conc*Type + Error(Plant/(conc)), w1b1)  summary(fit) </pre>	<p>已知：</p> <p>C02 数据集<sup>119</sup>含有以下四个变量：</p> <p><b>plant</b>（12 种植物，Qn1、Qn2、Qn3、Qc1、Qc1、Qc1、Mn1、Mn2、Mn3、Mc1、Mc2、Mc3）</p> <p><b>Type</b>（植物类型，魁北克、密西西比）</p> <p><b>Treatment</b>（nonchilled、chilled）</p> <p><b>conc</b>（7 种水平二氧化碳浓度，每种植物都要接受这 7 种水平二氧化碳的测试，即重复测量）</p> <p><b>uptake</b>（二氧化碳吸收量）</p> <p>将 <b>conc</b> 变量因子化</p> <p>提取 C02 数据集中 <b>Treatment</b> 为 <b>chilled</b> 的观测作为子集，保存在 <b>w1b1</b> 中；</p> <p>做重复测量方差分析，<b>uptake</b> 为因变量，<b>Type</b> 为组间因子，<b>conc</b> 为组内因子，<b>plant</b> 为重复受试对象，分析结果保存在 <b>fit</b> 中；</p> <p>展示结果：</p>
<pre> par(las=2)  par(mar=c(10, 4, 4, 2))  with(w1b1, interaction.plot(conc, Type, uptake, type="b", col=c("red", "blue"), pch=c(16, 18))) </pre>	<p>用 <b>interaction.plot()</b> 函数展示上述方差分析中的交互效用：</p> <p>标签垂直于坐标轴；</p> <p>图像边界为（10,4,4,2）；</p> <p>绘制交互影响图，用 <b>with</b> 绑定数据集，类型为 <b>b</b>，颜色为 <b>red</b> 和 <b>blue</b>，点的形状为 16 和 18</p>
<pre> boxplot(uptake ~ Type*conc, data=w1b1, col=c("gold", "green")) </pre>	<p>用 <b>boxplot()</b> 对上述方差分析画图，颜色为 <b>gold</b> 和 <b>green</b></p>

## 9.7 多元方差分析

多元方差分析（MANOVA）	当因变量（结果变量）不止一个时，可用_____对它们同时进行分析
<pre> library(MASS) attach(UScereal) </pre>	已知：

<sup>119</sup> 注意大写的 **C02** 和小写的 **co2** 是两个不同的数据集。

<pre>shelf &lt;- factor(shelf) y &lt;- cbind(calories, fat, sugars) aggregate(y, by=list(shelf), FUN=mean) cov(y) fit &lt;- manova(y ~ shelf) summary(fit)</pre>	<p>MASS 包中的 UScereal 数据集中包含但不限于以下变量：</p> <p>calories（谷物中的卡路里）</p> <p>fat（谷物中的脂肪）</p> <p>sugars（谷物中的糖）</p> <p>shelf（货架的三个水平，1、2、3 分别是底层、中层、顶层）</p> <p>为了研究谷物中的卡路里、脂肪和糖含量是否会因为储存架位置的不同而发生变化，以 calories、fat、shelf 为因素做单因素多元方差分析：</p> <p>载入 MASS 包：</p> <p>绑定 UScereal 数据集：</p> <p>横向合并 calories、fat、sugars 三个变量，组成数据集 y；</p> <p>按 shelf 分组计算 y 中三个变量的均值；</p> <p>计算 y 中各变量的方差协方差矩阵；</p> <p>计算上述单因素多元方差分析，保存在 fit 中；</p> <p>查看分析结果</p>
三个组的营养成分测量值不同	若上述分析的结果中 F 值显著，说明_____
summary.aov()	由于多元检验是显著的，可以使用_____函数对每一个变量做单因素方差分析
summary.aov(fit)	使用上述方法输出单变量分析结果

### 9.7.1 评估假设检验

多元正态性 方差协方差矩阵同质性	单因素多元方差分析有两个前提假设，一个是_____，一个是_____。
因变量组合成的向量服从一个多元正态分布 Q-Q 图	第一个假设即指_____，可以用_____来检验该假设条件

<pre>center &lt;- colMeans(y) n &lt;- nrow(y) p &lt;- ncol(y) cov &lt;- cov(y) d &lt;- mahal anobis(y, center, cov) coord &lt;- qqplot(qchisq(ppoints(n), df=p), d) abline(a=0, b=1)</pre>	<p>用 Q-Q 图检验多元正态性：</p> <p>获取 y 的列均值组成的向量，保存在 center 中；</p> <p>获取 y 的行数，保存在 n 中；</p> <p>获取 y 的列数，保存在 p 中；</p> <p>获取 y 的方差协方差矩阵，保存在 cov 中；</p>
--	--

<code>identify(coord\$x, coord\$y, label s=row.names(UScereal))</code> <sup>120</sup>	<p>返回 <code>y</code>（向量的组合）与 <code>center</code>（均值）的马氏距离的平方；</p> <p>绘制 Q-Q 图，保存在 <code>coord</code> 中；</p> <p>添加斜率为 1，截距为 0 的直线；</p> <p>用鼠标为离群点添加标签，标签取自 <code>UScereal</code> 数据集的行名称</p>
---	---

<code>mvoutlier</code> 包的 <code>ap.plot()</code> 函数	可用_____包中的_____函数来检验多元离群点
<pre>library(mvoutlier) outliers &lt;- aq.plot(y) outliers</pre>	<p>用上述方法检验多元离群点（多元因变量为 <code>y</code>），检验结果保存在 <code>outliers</code> 中</p>

## 9.7.2 稳健多元方差分析

<code>rrcov</code> 包中的 <code>Wilks.test()</code> 函数	稳健单因素 MANOVA 可通过_____包中的_____函数实现
<pre>library(rrcov) Wilks.test(y, shelf, method="mcd")</pre>	用上述方法做稳健单因素 MANOVA（多元因变量为 <code>y</code> ，因素为 <code>shelf</code> ）

## 9.8 用回归来做 ANOVA

<p>用一系列与因子水平相对应的数值型对照变量来代替因子</p> <p><code>k-1</code></p>	<p>因为线性模型要求预测变量是数值型，当 <code>lm()</code> 函数碰到因子时，它会_____；</p> <p>如果因子有 <code>k</code> 个水平，将会创建_____个对照变量。</p>
--	--

<sup>120</sup> 在原图中点旁边的标签处于点的左侧，但在命令中却没有相应的语句。我尝试在命令中加入 `pos=2`，但似乎并不管用。如果想在点左侧展示所有标签可用 `text()` 函数。

## 第 10 章 功效分析

检测到给定效应值时所需的样本量	功效分析可以帮助在给定置信度的情况下，判断_____
在某样本量内能检测到给定效应值的概率	功效分析可以也可以在给定置信度水平情况下，计算_____

### 10.1 假设检验速览

每种条件/组中观测的数目	样本大小指的是实验设计中_____
I 型错误的概率 发现效应不发生的概率	显著性水平（也称为 $\alpha$ ）由_____来定义 可以把它看作_____
1 减去 II 型错误的概率 真实效应发生的概率	功效通过_____来定义 可以把它看作_____
在备择或研究假设下效应的量 假设检验中使用的统计方法	效应值指的是_____ 效应值的表达式依赖于_____
便可推算第四个量	四个量（样本大小、显著性水平、功效和效应值）紧密相关，给定其中任意三个量，便可_____

### 10.2 用 pwr 包做功效分析

pwr 包	使用_____包做功效分析
-------	---------------

#### 10.2.1 t 检验

<p><code>pwr.t.test()</code></p> <p><code>pwr.t.test(n=, d=, sig.level=, power=, type=, alternative=)</code></p> <p><b>d</b> 为效应值，即标准化的均值之差  <math>d = \frac{\mu_1 - \mu_2}{\sigma}</math>, <math>\mu_1</math> 为组 1 均值, <math>\mu_2</math> 为组 2 均值, <math>\sigma^2</math> 为误差方差</p>	<p>使用_____函数做 t 检验的功效分析</p> <p>其语法格式为_____, 其中效应值是_____</p>
---	---

<p><code>d=1/1.25=0.8</code></p> <p><code>library(pwr)</code></p> <p><code>pwr.t.test(d=.8, sig.level=.05, power=.9, type="two.sample", alternative="two.sided")</code></p>	<p>已知：</p> <p>使用双尾独立样本 t 检验来比较使用手机和不使用手机两种情况下驾驶反应时间的均值。根据过去的经验知道反应时间有 1.25s 的标准偏差，并认定反应时间 1s 的差值是巨大的差异；如果差异存在，你希望有 90%的把握检测到它，也希望有 95%的把握不会误报差异显著</p>
---	---

	<p>此例中效应值 <math>d</math> 是多少？</p> <p>该研究需要多少受试者？</p>
<p><code>pwr.t.test(n=20, d=.5, sig.level=.01, type="two.sample", alternative="two.sided")</code></p>	<p>已知：</p> <p>使用双尾独立样本 <math>t</math> 检验来比较使用手机和不使用手机两种情况下驾驶反应时间的均值。想检测到总体均值 0.5 个标准偏差的差异，并且将误报差异的几率限制在 1% 内，能获得的受试者只有 40 人</p> <p>在该研究中，能检测到这么大总体均值差异的概率是多少？</p>

<p><code>pwr.t2n.test(n1=, n2=, d=, sig.level=, power=, alternative=)</code></p>	<p>如果两组中样本的数量不同，可用函数_____来做功效分析</p>
--	-------------------------------------

### 10.2.2 方差分析

<p><code>pwr.anova.test()</code></p> <p><code>pwr.anova.test(k=, n=, f=, sig.level=, power=)</code></p> $f = \sqrt{\frac{\sum_{i=1}^k p_i (\mu_i - \mu)^2}{\sigma^2}}$ <p>其中，<math>p_i = \frac{n_i}{N}</math></p> <p><math>n_i</math> = 组 <math>i</math> 的观测数目</p> <p><math>N</math> = 总观测数目</p> <p><math>\mu_i</math> = 组 <math>i</math> 均值</p> <p><math>\mu</math> = 总体均值</p> <p><math>\sigma^2</math> = 组内误差方差</p>	<p>_____函数可以对平衡单因素方差分析进行功效分析</p> <p>其语法格式为_____，其中效应值为_____</p>
<p><code>pwr.anova.test(k=5, f=.25, sig.level=.05, power=.8)</code></p>	<p>对五个组做单因素方差分析，要达到 0.8 的功效，效应值为 0.25，并选择 0.05 的显著性水平，计算各组需要的样本大小</p>

### 10.2.3 相关性

<p><code>pwr.r.test()</code></p> <p><code>pwr.r.test(n=, r=, sig.level=, power=, alternative=)</code></p> <p><math>r</math> 是效应值（通过线性相关系数衡量）</p>	<p>_____函数可以对相关性分析进行功效分析</p> <p>其语法格式为_____，其中效应值为_____</p>
<p><code>pwr.r.test(r=.25, sig.level=.05, power=.90, alternative="greater")</code></p>	<p>已知：</p> <p>正在研究抑郁与孤独的关系；<math>H_0: \rho \leq 0.25</math> 和 <math>H_1: \rho &gt; 0.25</math>，<math>\rho</math> 是两个心理变量的总体相关性大小；显著性水平为 0.05，而且如果 <math>H_0</math> 是错误的，有 90% 的信心拒绝 <math>H_0</math></p>



	问研究需要多少观测？
<b>10.2.4 线性模型</b>	
<p><code>pwr.f2.test()</code></p> <p><code>pwr.f2.test(u=, v=, f2=, sig.level=, power=)</code></p> <p><b>u</b> 和 <b>v</b> 分别是分子自由度和分母自由度，<b>f2</b> 是效应值</p>	<p>_____函数可以完成线性模型（比如多元回归）的功效分析</p> <p>其语法格式为_____，其中效应值为_____，<b>u</b> 和 <b>v</b> 分别指_____</p>
$f^2 = \frac{R^2}{1 - R^2}$ <p><math>R^2</math> = 多重相关性的总体平方值</p>	<p>当要评价一组预测变量对结果的影响程度时，适宜用_____公式来计算 <b>f2</b></p>
$f^2 = \frac{R_{AB}^2 - R_A^2}{1 - R_{AB}^2}$ <p><math>R_A^2</math> = 集合 <b>A</b> 中变量对总体方差的解释率</p> <p><math>R_{AB}^2</math> = 集合 <b>A</b> 和 <b>B</b> 中变量对总体方差的解释率</p>	<p>当要评价一组预测变量对结果的影响超过第二组变量（协变量）多少时，适宜用_____公式来计算 <b>f2</b></p>
总自变量数 <b>k</b>	【补】当要评价一组预测变量对结果的影响程度时， <b>u</b> 为_____
总自变量 <b>k</b> 减去集合 <b>B</b> 中的预测变量数	【补】当要评价一组预测变量对结果的影响超过第二组变量（协变量）多少时， <b>u</b> 为_____
<p><b>sig.level=0.05,</b></p> <p><b>power=0.90,</b></p> <p><b>u=3</b>（总预测变量数减去集合 <b>B</b> 中的预测变量数），</p> <p><b>f2=(0.35-0.30)/(1-0.35) = 0.0769</b></p> <p><b>pwr.f2.test(u=3, f2=0.0769, sig.level=0.05, power=0.90)</b></p>	<p>现假设想研究老板的<u>领导风格</u>（集合 <b>B</b>）对员工满意度的影响，是否超过<u>薪水和工作小费</u>（集合 <b>A</b>）对员工满意度的影响。领导风格可用四个变量来评估，薪水和与小费与三个变量有关。过去的经验表明，薪水和与小费能够解释约 30%的员工满意度的方差。而从现实出发，领导风格以及薪水和与小费（即集合 <b>A</b> 和集合 <b>B</b>）至少能解释 35%的方差。<sup>121</sup></p> <p>假定显著性水平为 0.05，那么在 90%的置信度情况下，你需要多少受试者才能得到这样的方差贡献率呢？</p> <p>本例中 sig.level=_____</p> <p>power=_____</p> <p>u=_____</p> <p>f2=_____</p> <p>计算样本量</p>

<sup>121</sup> 此处原文基础上进行了修改，原因是翻译不到位。这里的英文原文是：“Past experience suggests that salary and perks account for roughly 30 percent of the variance in worker satisfaction. From a practical standpoint, it would be interesting if leadership style accounted for at least 5 percent above this figure.” 理解成“在薪水和与小费的基础上，如果把领导风格的相应变量加进来，能够多解释 5%的总体方差”。

### 10.2.5 比例检验

<p><code>pwr.2p.test()</code>  <code>pwr.2p.test(h=, n=, sig.level=, power=)</code>  <b>h</b> 是效应值  <b>ES.h(p1, p2)</b> 函数</p>	<p>当比较两个比例时, 可使用_____函数进行功效分析          其语法格式为_____, 其中效应值为_____, 效应值可用_____计算</p>
<p><code>pwr.2p2n.test(h=, n1=, n2=, sig.level=, power=)</code></p>	<p>当各组中 <b>n</b> 不相同时, 则使用函数_____</p>

<p><code>pwr.2p.test(h=ES.h(.65, .6), sig.level=.05, power=.9, alternative="greater")</code></p>	<p>假定现在对某流行药物能缓解 60%使用者的症状感到怀疑。而一种更贵的新药如果能缓解 65%使用者的症状, 就会被投放到市场中。假设有 90%的把握得出新药更有效的结论, 并且希望有 95%的把握不会误得结论。由于只对评价新药是否比标准药物更好感兴趣, 因此只需用单边检验。</p> <p>在本例中, 需要多少受试者才能够检测到两种药物存在这一特定的差异?</p>
--	--

### 10.2.6 卡方检验

<p>两个类别型变量的关系          零假设是变量之间独立, 备择假设是不独立  <code>pwr.chisq.test()</code>  <code>pwr.chisq.test(w=, N=, df=, sig.level=, power=)</code>  <b>w</b> 是效应值</p> $w = \sqrt{\sum_{i=1}^m \frac{(p_{0i} - p_{1i})^2}{p_{0i}}}$ <p><math>p_{0i}</math> = <b>H0</b> 时第 <b>i</b> 单元格中的概率  <math>p_{1i}</math> = <b>H1</b> 时第 <b>i</b> 单元格中的概率  <b>m</b> 指列联单元格的数目</p>	<p>卡方检验常常用来评价_____;</p> <p>典型的零假设是变量之间_____, 备择假设是_____;</p> <p>_____函数可以评估卡方检验的功效、效应值和所需的样本大小;</p> <p>其语法格式为_____, 其中效应值为_____, 效应值可用_____计算</p>
--	---

<p><b>ES.w2(P)</b><sup>122</sup>          假设的双因素概率表</p>	<p>函数_____可以计算双因素列联表中备择假设的效应值, <b>P</b> 是一个_____</p>
<p><math>(r-1)(c-1)</math>, <b>r</b> 是行数, <b>c</b> 是列数</p>	<p>假设你想研究人种与工作晋升的关系。你预期样本中 70%是白种</p>

<sup>122</sup> `ES.w2()` 的函数为

```
function(p)
{
  pi <- apply(P, 1, sum)
  pj <- apply(P, 2, sum)
  P0 <- pi %*% t(pj)
  sqrt(sum((P - P0)^2/P0))
}
```

由此可见 **p** 为 **H1** 是第 **i** 个单元格中的概率, **p0** 是经过计算后得到的 **H0** 时第 **i** 个单元格中的概率。

<pre>prob &lt;- matrix(c(.42, .28, .03, .07, .10, .10), byrow=TRUE, nrow=3)  ES.w2(prob)  pwr.chi.sq.test(w=.1853, df=2, sig.level=.05, power=.9)</pre>	<p>人，10%是非洲裔，20%是拉丁裔。而且，你认为 60%的白种人倾向于得到晋升，相比之下，30%的非洲裔和 20%的拉丁裔倾向于得到晋升。<sup>123</sup>取 0.05 的显著性水平和 0.9 的预期功效水平。</p> <table><tr><th>人种</th><th>晋升比例</th><th>未晋升比例</th></tr><tr><td>白种人</td><td>0.42</td><td>0.28</td></tr><tr><td>非洲裔</td><td>0.03</td><td>0.07</td></tr><tr><td>拉丁裔</td><td>0.10</td><td>0.10</td></tr></table> <p>双因素列联表的自由度为_____；</p> <p>求效应值；</p> <p>计算所需样本数量</p>	人种	晋升比例	未晋升比例	白种人	0.42	0.28	非洲裔	0.03	0.07	拉丁裔	0.10	0.10
人种	晋升比例	未晋升比例											
白种人	0.42	0.28											
非洲裔	0.03	0.07											
拉丁裔	0.10	0.10											

### 10.2.7 在新情况中选择合适的效应值

<pre>samsize &lt;- NULL for (i in 1:nes){   result &lt;- pwr.anova.test(k=5,     f=es[i], sig.level=.05, power=.9)   samsize[i] &lt;- ceiling(result\$n) }</pre> <sup>124</sup>	<p>已知：</p> <pre>es &lt;- seq(0.1, 0.5, 0.01) nes &lt;- length(es) k=5 sig.level=0.05 power=0.9</pre> <p>将 es 中的每一个元素当做效应值，用 for 循环生成每个元素对应的单因素 ANOVA 检验中检测显著效用所需要的样本大小，保存在 samsize 中</p>
---	--

## 10.3 绘制功效分析图形

	<p>已知：</p> <pre>r &lt;- seq(0.1, 0.5, 0.01) nr &lt;- length(r) p &lt;- seq(0.4, 0.9, 0.1) np &lt;- length(p)</pre>
<pre>samsize &lt;- array(numeric(nr*np), dim=c(nr, np))</pre>	

<sup>123</sup> 也许我这样翻译更直白一些。

<sup>124</sup> 此例中用 for 循环生成一个序列的数的方式有借鉴意义。另外，注意，尽管 result 不是一个数据集，这里仍然能使用 result\$n。ceiling(x) 的含义是不小于 x 的最小整数。

	创建一个 <code>nr*np</code> 的二维数组，数组中的元素均为 0，保存在 <code>samsize</code> 中
<pre>for (i in 1:np){   for (j in 1:nr){     result &lt;- pwr.r.test(n = NULL, r = r       [j],     sig.level = .05, power = p[i],     alternative = "two.sided")     samsize[j,i] &lt;- ceiling(result\$n)   } }</pre> <sup>125</sup>	将 <code>p</code> 中的每一个元素当做效应值，同样的，将 <code>r</code> 中的每一个元素当做效应值，显著性水平为 0.05（双尾检测），做相关性分析的 功效分析，得到每一个功效及其效应水平下的样本大小，并按顺序放入 <code>samsize</code> 中。

10.4 其他软件包

<sup>125</sup> 此例中用两个 `for` 循环生成一个二维数组的方式有借鉴意义。

## 第 11 章 中级绘图

### 11.1 散点图

<pre>attach(mtcars)  plot(wt, mpg, pch=19)  abline(lm(mpg~wt), col="red", lwd=2, lty=1)  lines(lowess(wt, mpg), col="blue", lwd=2, lty=2)</pre>	<p>绘制添加了最佳拟合曲线的散点图：</p> <p>绑定数据集 <code>mtcars</code>；</p> <p>绘制横轴为车重（<code>wt</code>），纵轴为每加仑英里数（<code>mpg</code>）的散点图，点的符号为 19；</p> <p>添加最佳拟合的线性直线，红色，线宽为 2，线型为 1；</p> <p>添加平滑拟合曲线，蓝色，线宽为 2，线型为 2</p>
---	---

car 包中的 <code>scatterplot()</code> 函数	_____包中的_____函数增强了散点图的许多功能，能生成各子集的散点图与其相应的拟合曲线
<pre>library(car) scatterplot(mpg ~ wt   cyl, data=mtcars, lwd=2, legend.plot=TRUE, id.method="identify", labels=row.names(mtcars), boxplots="xy")</pre>	<p>针对数据集 <code>mtcars</code>，用上述函数绘制有四个、六个、八个气缸（<code>cyl</code>）的汽车每加仑英里数（<code>mpg</code>）对车重（<code>wt</code>）的图形，线宽为 2，在左上边界添加图例，采用交互式的方式取点，点标签取自 <code>mtcars</code> 数据集的行名称，添加 <code>mpg</code> 和 <code>wt</code> 的边界箱线图</p>

#### 11.1.1 散点图矩阵

<code>pairs()</code>	用_____函数创建基础的散点图举证
<code>pairs(~mpg+disp+drat+wt, data=mtcars)</code>	针对数据集 <code>mtcars</code> ，用上述函数创建包含 <code>mpg</code> 、 <code>disp</code> 、 <code>drat</code> 、 <code>wt</code> 四个变量的散点图矩阵 <sup>126</sup>

car 包中的 <code>scatterplotMatrix()</code> 函数	<p>_____包中的_____函数也可以生成散点图矩阵，并有以下可选操作：</p> <p>以某个因子为条件绘制散点图矩阵；</p> <p>包含线性和平滑拟合曲线；</p> <p>在主对角线放置箱线图、密度图或者直方图；</p> <p>在各单元格的边界添加轴须图</p>
<pre>library(car) scatterplotMatrix(~ mpg + disp + drat + wt, data=mtcars, spread=FALSE, lty.smooth=2)</pre>	<p>针对数据集 <code>mtcars</code>，用上述函数创建包含 <code>mpg</code>、<code>disp</code>、<code>drat</code>、<code>wt</code> 四个变量的散点图矩阵，不添加展示分散度和对称信息的直线，拟</p>

<sup>126</sup> 尽管原文说“主对角线上方和下方的六幅散点图是相同的”，但实际上以主对角线为对称的两幅图坐标系不同。可以这样识别：每个变量所处的那一列的三幅图的横轴都是该变量，每个变量所处的那一行的三幅图的纵轴都是该变量。

	合曲线使用虚线（默认添加线性和平滑拟合曲线，在主对角线上默认添加核密度曲线和轴须图） <sup>127</sup>
<code>library(car)</code> <code>scatterplotMatrix(~ mpg + disp + drat + wt   cyl, data=mtcars, spread=FALSE, diagonal="histogram")</code>	针对数据集 <code>mtcars</code> ，依据气缸数（ <code>cyl</code> ）划分子群，创建包含 <code>mpg</code> 、 <code>disp</code> 、 <code>drat</code> 、 <code>wt</code> 四个变量的散点图矩阵，主对角线图形为直方图，不添加展示分散度和对称信息的直线
<code>by.group=TRUE</code>	默认地，回归直线拟合整个样本，如果要依据各子集分别生成回归拟合直线，而不是拟合整个样本可以在上述函数中添加_____参数

<code>gclus</code> 包中的 <code>cpairs()</code> 函数	_____包中的_____函数提供了一个有趣的散点图矩阵变种：它含有可以重排矩阵中变量位置的选项，可以让相关性更高的变量更靠近主对角线，还能对各单元格进行颜色编码来展示变量间的相关性大小
<code>library(gclus)</code> <code>mydata &lt;- mtcars[c(1, 3, 5, 6)]</code> <code>mydata.corr &lt;- abs(cor(mydata))</code> <code>mycolors &lt;- dmat.color(mydata.corr)</code> <code>myorder &lt;- order.single(mydata.corr)</code> <code>cpairs(mydata, myorder,</code> <code>panel.colors=mycolors,</code> <code>gap=0.5)</code>	使用上述函数，依据 <code>mpg</code> 、 <code>disp</code> 、 <code>drat</code> 、 <code>wt</code> 四个变量的相关性强弱安排其散点图矩阵的排序（相关性更高的变量更靠近主对角线）和颜色：  将 <code>mtcars</code> 数据集的第 1、3、5、6 变量组成新的数据集 <code>mydata</code> ；  令 <code>mydata.corr</code> 为 <code>mydata</code> 中各变量相关系数的绝对值；  令 <code>mycolor</code> 为参照相关性所得的颜色参数；  令 <code>myorder</code> 为参照相关性所得的排序参数；  绘制参照相关性所得的散点图，各单元格间的间距为 0.5

### 11.1.2 高密度散点图

<code>set.seed(1234)</code> <code>n &lt;- 10000</code> <code>c1 &lt;- matrix(rnorm(n, mean=0, sd=.5), ncol=2)</code> <code>c2 &lt;- matrix(rnorm(n, mean=3, sd=2), ncol=2)</code> <code>mydata &lt;- rbind(c1, c2)</code> <code>mydata &lt;- as.data.frame(mydata)</code> <code>names(mydata) &lt;- c("x", "y")</code>	按下列代码获得 10000 个分布在两个重叠的数据群中的观测点：  设定随机数种子为 1234；  令 <code>n</code> 为 10000；  生成 <code>n</code> 个均值为 0，标准差为 0.5 的正态随机数，放入名为 <code>c1</code> 的两列矩阵中；  生成 <code>n</code> 个均值为 3，标准差为 2 的正态随机数，放入名为 <code>c2</code> 的两列矩阵中；
--	---

<sup>127</sup> 主对角线上的图形样式可以由参数 `diagonal=c()` 界定，括号中可填"`density`"、"`boxplot`"、"`histogram`"、"`oned`"、"`qqplot`"、"`none`"。

	<p>按行合并 <b>c1</b> 和 <b>c2</b>，生成 <b>mydata</b>；</p> <p>把 <b>mydata</b> 转换为数据框；</p> <p>把 <b>mydata</b> 的两列分别命名为 <b>x</b> 和 <b>y</b></p>
<b>with(mydata, plot(x, y))</b>	针对 <b>mydata</b> 数据集，生成 <b>x</b> 和 <b>y</b> 的散点图（此时数据点的重叠非常严重）

<b>smoothScatter()</b> 函数	_____函数可利用核密度估计生成用颜色密度来表示点分布的散点图
<b>with(mydata, smoothScatter(x, y))</b> <sup>128</sup>	针对 <b>mydata</b> 数据集，用上述函数生成 <b>x</b> 和 <b>y</b> 的散点图

<b>hexbin</b> 包中的 <b>hexbin()</b> 函数	_____包中的_____函数将二元变量的封箱放到六边形单元格中
<pre>library(hexbin) with(mydata, {   bin &lt;- hexbin(x, y, xbins=50)   plot(bin) })</pre>	用上述函数生成 <b>x</b> 和 <b>y</b> 的散点图，横轴被 50 个间隔分开 <sup>129</sup>

<b>IDPmisc</b> 包中的 <b>iplot()</b> 函数	_____包中的_____函数可通过颜色来展示点的密度（在某特定点上数据点的数目）
<pre>library(IDPmisc) with(mydata, iplot(x, y))</pre>	用上述包绘制 <b>x</b> 和 <b>y</b> 的散点图

### 11.1.3 三维散点图

<b>scatterplot3d</b> 包的 <b>scatterplot3d()</b> 函数	_____包中的_____函数可以绘制三维散点图
<pre>scatterplot3d(x, y, z)</pre> <p><b>x</b> 被绘制在水平轴上，<b>y</b> 被绘制在透视轴上，<b>z</b> 被绘制在竖轴上 <sup>130</sup></p>	三维散点图函数的语法格式为_____， <b>x</b> 被绘制在_____上， <b>y</b> 被绘制在_____上， <b>z</b> 被绘制在_____上
<pre>library(scatterplot3d) attach(mtcars) scatterplot3d(wt, disp, mpg, pch=16,   highlight.3d=TRUE,   type="h")</pre> <sup>131</sup>	绑定数据集 <b>mtcars</b> 数据集，绘制 <b>wt</b> （ <b>x</b> ）、 <b>disp</b> （ <b>y</b> ）、 <b>mpg</b> （ <b>z</b> ）的三维散点图，点形状为 16，按所处透视轴（ <b>y</b> ）的位置不同区分点的颜色 <sup>132</sup> ，添加链接点到水平面的垂直线 <sup>133</sup>

<sup>128</sup> 用参数 **colramp=colRampPalette(c("white", "red"))**，即 **with(mydata, smoothScatter(x, y, colramp = colRampPalette(c("white", "red"))))** 可以重新定义该图的颜色，其中 **white** 为背景色，**red** 为点的颜色。

<sup>129</sup> 例如，如果 **xbins=n**，那么在横轴上一个挨着一个摆放小六边形的话，正好可以摆放 51 个小六边形就能把横轴摆满。此处定义了每个小六边形的宽度为横轴边框宽度的  $\frac{1}{n+1}$ 。

<sup>130</sup> 此处描述和原文不一致。从作图的实际效果来看，**y** 应该是透视轴（即带来纵深感的那条轴），**z** 应该是竖轴。

<sup>131</sup> 如果遇到 “**Error in plot.new(): figure margins too large**” 可以采用以下代码取消边界限制：

```
op <- par(mar=rep(0, 4))
plot.new() 【转下一页】
```

<code>fit &lt;- lm(mpg ~ wt+dis p)</code> <code>s3d\$plane3d(fit)</code>	在上图中添加 mpg 对 wt 和 disp 回归的回归面
---	-------------------------------

rgl 包中的 plot3d() 函数	_____包中的_____函数可以创建可交互的三维散点图
<code>library(rgl)</code> <code>attach(mtcars)</code> <code>plot3d(wt, disp, mpg, col="red", size=5)</code>	针对 mtcars 数据集，用上述包创建 wt (x)、disp (y)、mpg (z) 的交互三维散点图，点的颜色为红色，点大小为 5

Rcmdr 包的 scatter3d() 函数	_____包中的_____函数也可以创建可交互的三维散点图，可包含各种回归曲面，比如线性、二次、平滑和附加等类型
<code>library(Rcmdr)</code> <code>attach(mtcars)</code> <code>scatter3d(wt, disp, mpg)</code>	针对 mtcars 数据集，用上述包创建 wt (x)、disp (y)、mpg (z) 的交互三维散点图

### 11.1.4 气泡图

<code>symbols()</code>	_____函数可以创建气泡图，该函数可以在指定的(x,y)坐标上绘制圆圈图、方形图、星形图、温度计图和箱线图
<code>symbols(x, y, circle=z)</code>	上述函数以 z 作圆圈半径的语法为_____
<code>symbols(x, y, circle=sqrt(z/pi))</code>	上述函数以 z 作圆圈面积的语法为_____

<code>r &lt;- sqrt(disp/pi)</code> <code>symbols(wt, mpg, r, inches=0.3, fg="white", bg="lightblue")</code>	针对 mtcars 数据集，用上述函数绘制车重 wt (x) 和每加仑英里数 mpg (y) 的气泡图，气泡面积由发动机排量 disp (z) 决定，圆圈大小的比例因子为 0.3
<code>text(wt, mpg, rownames(mtcars), cex=0.6)</code>	用 mtcars 数据集的行名称命名上图中的气泡，字体大小为 0.6

## 11.2 折线图

<code>plot(x, y, type=)</code> 和 <code>lines(x, y, type=)</code> <code>plot()</code> 创建新图 <code>lines()</code> 在已存在的图形上添加信息	折线图可以用下面两个函数之一来创建：_____和_____ 以上两个函数的工作原理并不相同，_____是创建一幅新图，_____是在已存在的图形上添加信息
<code>type=n</code> <code>lines()</code>	如果对图形有要求，可以先通过 <code>plot()</code> 函数中的_____来创建坐标轴、标题和其他图形特征，然后再用_____添加各种需要的曲线

	<code>plot()</code> 和 <code>line()</code> 中的 type 的可选值有：
--	--

`par(op)`  
<sup>132</sup> 由 `highlight.3d=TRUE` 实现。  
<sup>133</sup> 由 `type="h"` 实现。



p	只有空心点
l	只有线
o	实心点和线（即线覆盖在点上）
b	线段和空心点
c	线段（不绘制点）
s	阶梯线（先水平再竖直）
S	阶梯线（先竖直再水平）
h	直方图式的垂直线
n	不生成任何点和线（通常用来为后面的命令创建坐标轴）

	<p>已知：</p> <p>orange 数据集中包含了三个变量 Tree（因子型，从 1 到 5 表示五种橘树），age（生长天数），circumference（树的胸围）<sup>134</sup></p> <p>作图展示这五种橘树随时间推移的生长状况：</p>
Orange\$Tree <- as.numeric(Orange\$Tree)	将变量 Tree 转换为数值型
ntrees <- max(Orange\$Tree)	令 ntrees 为 Tree 中的最大值
xrange <- range(Orange\$age)	令 xrange 为 age 的值域
yrange <- range(Orange\$circumference)	令 yrange 为 circumference 的值域
plot(xrange, yrange, type="n")	创建由 xrange 定义横轴，由 yrange 定义纵轴的空图形
colors <- rainbow(ntrees)	令 colors 为由 ntrees 定义的彩虹色
linetype <- c(1:ntrees)	令 linetype 为由 1 到 ntrees 的向量
plotchar <- seq(18, 18+ntrees, 1)	令 plotchar 为从 18 起到 18+ntrees 截止的步长为 1 的序列
<pre>for (i in 1:ntrees) {   tree &lt;- subset(Orange, Tree==i)   lines(tree\$age, tree\$circumference,         type="b",         lwd=2,         lty=linetype[i],         col=colors[i],         pch=plotchar[i]   ) }</pre>	采用 for 循环生成每一种树的折线图，类型为 b，线宽为 2，线型由 linetype 定义，颜色由 colors 定义，点的形状由 plotchar 定义

<sup>134</sup> 即树干距地面约 1.2 米处的周长。原文翻译成年轮是不恰当的。

<pre>legend(xrange[1], yrange[2],       1: ntrees,       cex=0.8,       col=col ors,       pch=pl otchar,       lty=li netype)</pre>	<p>为上图添加图例，以 <code>xrange</code> 的第一个数为横坐标，以 <code>yrange</code> 的第二个数为纵坐标安放图例，图例的元素为 5 种树的类型，对应的点形状，颜色，线型分别以 <code>plotchar</code>、<code>colors</code>、<code>linetype</code> 定义，字号为 0.8。</p>
--	---

## 11.3 相关图

<p><code>corrgram</code>包中的 <code>corrgram()</code> 函数</p> <pre>corrgram(x, order=, panel =, text. panel =, di ag. panel =)</pre> <p><code>x</code> 是一行一个观测的数据框。</p>	<p>_____包中的_____函数能以图形的方式展示相关系数矩阵其语法格式为_____</p>
<p>使用主成分分析法对变量重排序 使得二元变量的关系模式更为明显</p>	<p>当 <code>order=TRUE</code> 时，相关矩阵将_____，这将使得_____</p>
<pre>lower. panel upper. panel text. panel di ag. panel</pre>	<p>可以通过选项_____和_____来分别设置主对角线下方和上方的元素类型，而_____和_____选项控制着主对角线元素类型</p>

	非对角线位置的面板选项有： <sup>135</sup>
<code>panel. pie</code>	用饼图的填充比例来表示相关性大小
<code>panel. shade</code>	用阴影的深度来表示相关性大小
<code>panel. ellipse</code>	画一个置信椭圆和平滑曲线
<code>panel. pts</code>	画一个散点图
<code>panel. conf</code>	画出相关性及其置信区间
	主对角线位置的面板选项有：
<code>panel. txt</code>	输出变量名
<code>panel. mi nmax</code>	输出变量的最大最小值和变量名
<code>panel. ednsi ty</code>	输出核密度曲线和变量名

<code>col orRampPal ette()</code> 函数	可在 <code>col.corrgram()</code> 函数中用_____函数来指定颜色
	使用上述函数，针对 <code>mtcars</code> 数据集中的变量作相关系数图： <sup>136</sup>
<code>li brary(corrgram)</code>	载入包
<code>col s &lt;- col orRampPal ette(c("darkgol denrod4", "burlywood1", "darkkhaki", "darkgr</code>	为将要生成的相关系数图指定四种颜色： <code>darkgol denrod4</code> 、

<sup>135</sup> 该表参照中文第二版内容，中文第一版表 11-2 的排版有问题。

<sup>136</sup> 此处采用了第二版的代码，其比第一版的代码更简洁。

een"))	<code>burlywood1</code> 、 <code>darkkhaki</code> 、 <code>darkgreen</code>
<code>corrgram(mtcars, order=TRUE, col.regions=cols, lower.panel=panel.shade, upper.panel=panel.conf, text.panel=panel.txt)</code>	绘制相关系数图，使用主成分分析法对变量重新排序，主对角线下方颜色深度表示相关性大小，主对角线上方用饼图填充比例来表示相关性大小，主对角线上展示各变量名

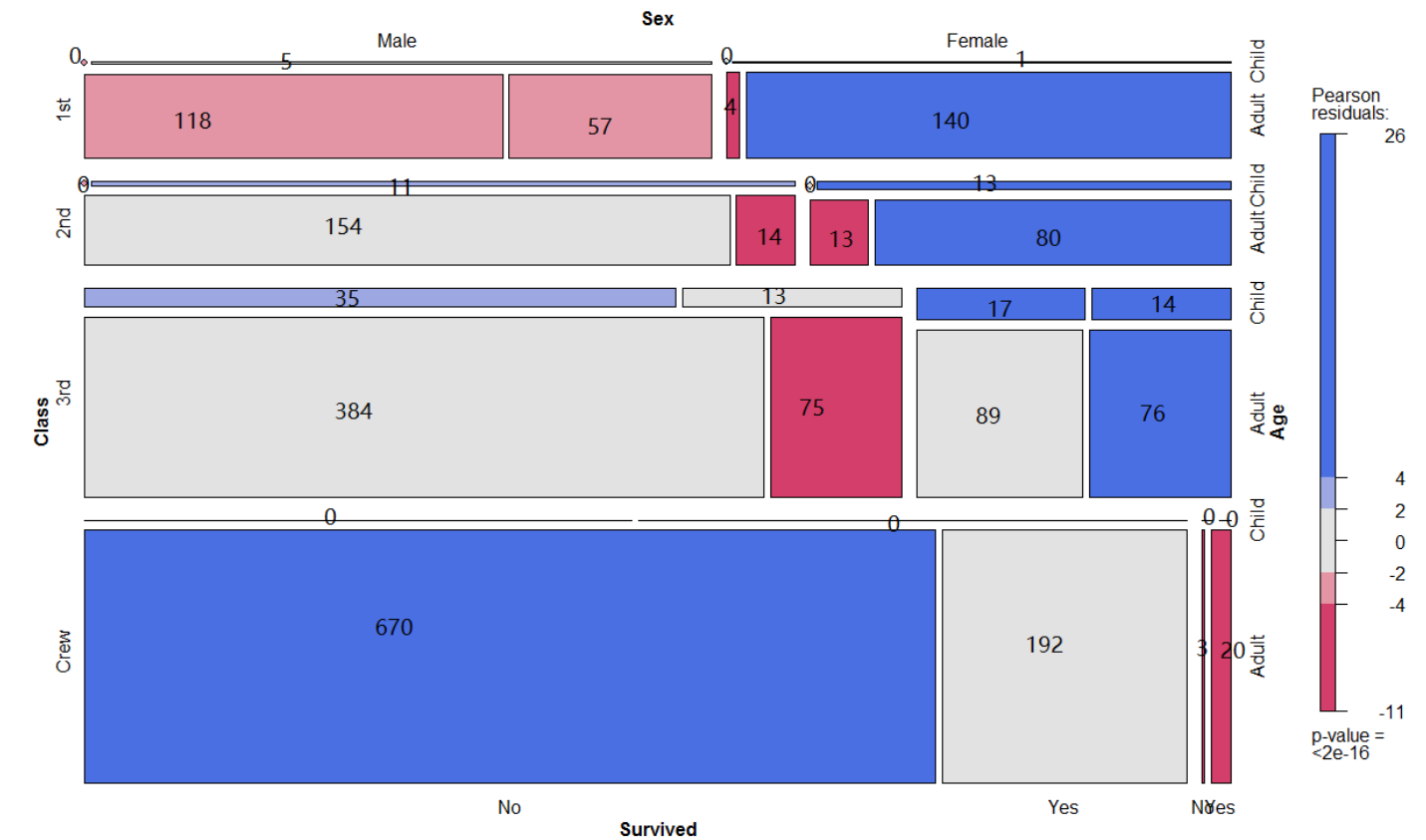
11.4 马赛克图

<code>vcd</code> 包中的 <code>mosaic()</code> 函数	_____包中的_____函数可以绘制马赛克图
<code>shade=TRUE</code>	在上述函数中添加选项_____将根据拟合模型的皮尔逊残差对图上色 <sup>137</sup>
<code>legend=TRUE</code>	添加选项_____将展示残差的图例

<code>library(vcd)</code> <code>mosaic(Titanic, shade=TRUE, legend=TRUE)</code>	针对 <code>Titanic</code> 数据集绘制马赛克图，根据拟合后的残差为图形上色并添加残差图例 <sup>138</sup>
颜色参照注释 136 中的图例，皮尔逊残差大于 0 的色块表示真实值大于拟合值，皮尔逊残差小于 0 的色块表示真实值小于拟合值	应该怎样解释马赛克图的配色？

<sup>137</sup> 马赛克图所用的拟合模型是对数线型模型，采用了迭代比例拟合的算法。可以查看 `mosaic`、`loglm`、`loglin` 的帮助文档获知更多。

<sup>138</sup> 第一版中第 261 页和彩图插图中的图 11-23 是有问题的，该图图例中 -4 至 -11 的部分以及图中的对应色块是应该是红色。另外，为了方便理解马赛克图的含义，我在这里放上一张标记了色块所对应 `ftable(Titanic)` 中的数字的马赛克图。



## 第 12 章 重抽样与自助法

### 12.1 置换检验

将统计量与理论分布进行比较 将其与置换观测数据后获得的经验分布进行比较	置换方法和参数方法都计算了相同的 $t$ 统计量。但置换方法并不是_____，而是_____，根据统计量值的极端性判断是否有足够理由拒绝零假设
数据所有可能的排列组合 从所有可能的排列中进行抽样	经验分布依据的是_____，此时的置换检验称作“精确”检验。随着样本量的增加，获取所有可能排列的时间开销会非常大，这种情况下，你可以使用蒙特卡洛模拟，_____，获得一个近似的检验

<code>set.seed(1 2 3 4)</code>	设定随机种子 1 2 3 4 （即_____）可以重现本章的所有结果
--------------------------------	------------------------------------

### 12.2 用 coin 包做置换检验

coin 包	_____包提供了一个进行置换检验的一般性框架，其可选用的 coin 函数有：
两样本和 K 样本置换检验	<code>oneway_test(y ~ A)</code>
含一个分层（区组）因子的两样本和 K 样本置换检验	<code>oneway_test(y ~ A   C)</code>
Wilcoxon-Mann-Whitney 秩和检验	<code>wilcox_test(y ~ A)</code>
Kruskal-Wallis 检验	<code>kruskal_test(y ~ A)</code>
Person 卡方检验	<code>chisq_test(A ~ B)</code>
Cochran-Mantel-Haenszel 检验	<code>cmh_test(A ~ B   C)</code>
线性关联检验	<code>lbl_test(D ~ E)</code>
Spearman 检验	<code>spearman_test(y ~ x)</code>
Friedman 检验	<code>friedman_test(y ~ A   C)</code>
Wilcoxon 符号秩检验	<code>wilcoxsign_test(y1 ~ y2)</code>

y 和 x 是数值变量 A 和 B 是分类因子 C 是类别型区组变量 D 和 E 是有序因子 y1 和 y2 是相匹配的数值变量	在 coin 函数中，y 和 x 是_____，A 和 B 是_____，C 是_____，D 和 E 是_____，y1 和 y2 是_____
<code>function_name( formula, data, distribution= )</code>	以上函数的形式都是_____

<b>exact</b> : 在零假设条件下, 分布的计算是精确的 (即依据所有可能的排列组合), 仅可用于两样本问题; <b>asymptotic</b> 渐进分布; <b>approximate(B=#)</b> : 蒙特卡洛重抽样, 其中#指所需重复的次数。	<b>distribution=</b> 的可选值是_____
--	---------------------------------

因子和有序因子 数据框	在 <b>coin</b> 包中, 类别型变量和序数变量必须分别转换为_____和_____ 数据要以_____形式储存
----------------	---

### 12.2.1 独立两样本和 K 样本检验

	已知: <pre>score &lt;- c(40, 57, 45, 55, 58, 57, 64, 55, 62, 65) treatment &lt;- factor(c(rep("A",5), rep("B",5))) mydata &lt;- data.frame(treatment, score)</pre> 分别对 <b>mydata</b> 做 <b>t</b> 检验和单因素置换检验:
<pre>t.test(score~treatment, data=mydata, var.equal=TRUE)</pre>	t 检验和
<pre>library(coin) oneway_test(score~treatment, data=mydata, distribution="exact")</pre>	单因素置换检验

<pre>wilcox_test(Prob ~ So, data=UScrime, distribution="exact")</pre>	已知: <pre>library(MASS)</pre> <pre>UScrime &lt;- transform(UScrime, So = factor(So))</pre> (即美国南部监禁概率与非南部监禁概率, 监禁概率由变量 <b>Prob</b> 表示) 对 <b>UScrime</b> 中的 <b>Prob</b> 和 <b>So</b> 做 <b>Wilcoxon</b> 秩和检验 (置换检验, 精确检验)
---	---

<pre>library(multcomp) set.seed(1234) oneway_test(response~trt, data=cholesterol, distribution=approximate(B=9999))</pre>	已知: <b>multcomp</b> 包的 <b>cholesterol</b> 数据集中, <b>response</b> 表示 50 名患者各自的降低胆固醇的疗效, <b>trt</b> 表示 5 种药物疗法, 对此做 <b>k</b> 样本置换检验 (设定随机数种子为 1234, 采用蒙特卡洛重抽样, 重复 9999 次)
---	---

### 12.2.2 列联表中的独立性

<code>chisq_test()</code> <code>cmh_test()</code>	通过_____或_____函数，我们可用置换检验判断两类别型变量的独立性
数据可根据第三个类别型变量进行分层	当_____时，需要使用后一个函数
<code>lbl_test()</code>	若变量都是有序型，可使 <code>lbl_test()</code> 函数来检验是否存在线性趋势。

	<p>已知：</p> <p><code>vcd</code> 包中的 <code>Arthritis</code> 数据集包含了关节炎治疗（<code>Treatment</code>）与效果（<code>Improvement</code>）两个变量。<code>Treatment</code> 有两个水平（安慰剂 <code>Placebo</code>、治疗 <code>Treated</code>），<code>Improvement</code> 有三个水平（无 <code>None</code>、部分 <code>Some</code>、显著 <code>Marked</code>），变量 <code>Improved</code> 以有序因子形式编码</p> <p>对此实施卡方检验的置换检验：</p>
<code>library(coin)</code> <code>library(vcd)</code>	载入 <code>coin</code> 包和 <code>vcd</code> 包；
<code>Arthritis &lt;- transform(Arthritis, Improved=as.factor(as.numeric(Improved)))</code>	<p>将 <code>Arthritis</code> 中的 <code>Improvement</code> 变成分类因子；</p> <p>如果不变成分类因子的话会怎样？</p>
不变成分类因子将会生成一个线性与线性趋势检验，而非卡方检验	
<code>set.seed(1234)</code>	设定随机数种子（1234）
<code>chisq_test(Treatment~Improved, data=Arthritis, distribution=approximate(B=9999))</code>	做卡方检验对应的置换检验（利用蒙特卡洛重抽样，重复 9999 次）

### 12.2.3 数值变量间的独立性

<code>spearman_test()</code>	_____函数提供了两数值变量的独立性置换检验
<code>spearman_test(Illiteracy~Murder, data=states, distribution=approximate(B=9999))</code>	<p>已知：</p> <p><code>set.seed(1234)</code></p> <p><code>states &lt;- as.data.frame(state.x77)</code></p> <p>其中包括了美国文盲率（<code>Illiteracy</code>）和谋杀率（<code>Murder</code>）两个变量，请对两者做独立性置换检验，利用蒙特卡洛法，重复 9999 次</p>

### 12.2.4 两样本和 K 样本相关性检验

处于不同组的观测已经被分配得当 使用了重复测量时	当_____, 或者_____, 样本相关检验便可派上用场
<code>wilcoxsign_test()</code> <code>friedman_test()</code>	对于两配对组的置换检验, 可使用_____函数; 多于两组时, 使用_____函数

<code>wilcoxsign_test(U1~U2, data=UScrime, distribution="exact")</code>	已知:  UScrime 数据集中包含了 U1 (城市男性 14-24 年龄段失业率) 和 U2 (城市男性 35-39 年龄段失业率), 两个变量对于美国 50 个州都有记录  用精确 Wilcoxon 符号秩检验来判断两个年龄段的失业率是否相等
---	--

### 12.2.5 深入探究

## 12.3 lmPerm 包的置换检验

lmPerm 包	_____可做线性模型的置换检验
<code>lmp()</code> 和 <code>aovp()</code>	_____和_____函数的参数与 <code>lm()</code> 和 <code>aov()</code> 函数类似, 只额外添加了 <code>perm =</code> 参数。
<b>Exact</b> 根据所有可能的排列组合生成精确检验。 <b>Prob</b> 从所有可能的排列中不断抽样, 直至估计的标准差在估计的 <b>p 值 0.1</b> 之下, 判停准则由可选的 <b>Ca</b> 参数控制。 <b>SPR</b> 使用贯序概率比检验来判断何时停止抽样。	<code>perm =</code> 选项的可选值有"Exact"、"Prob"或"SPR"。  Exact、Prob、 SPR 的区别是_____
观测数大于 10 小样本问题	若_____, <code>perm = "Exact"</code> 将自动默认转为 <code>perm = "Prob"</code> , 因为精确检验只适用于_____

### 12.3.1 简单回归和多项式回归

<code>library(lmPerm)</code> <code>fit &lt;- lmp(weight~height, data=women, perm="Prob")</code>	已知:  women 数据集包含了 15 名女性的身高 (height) 和体重 (weight) 间的关系  做 weight 对 height 的简单回归的置换检验, perm 参数为 "Prob", 结果保存在 fit 中
<code>library(lmPerm)</code> <code>fit &lt;- lmp(weight~height + I(height^2), data=women, perm="Prob")</code>	已知:  women 数据集包含了 15 名女性的身高 (height) 和体重 (weight) 间的关系

	做 weight 对 height 和 height 平方的简单回归的置换检验, perm 参数为 “Prob”, 结果保存在 fit 中
要达到判停准则所需的迭代次数	在 summary(fit)中, Coefficients 中的 Iter 列表示_____

### 12.3.2 多元回归

<pre>library(lmPerm) fit &lt;- lmp(Murder~Population + Illiteracy+Income+Frost, data=states, perm="Prob")</pre>	<p>已知:</p> <pre>states &lt;- as.data.frame(state.x77)</pre> <p>states 包含了美国 50 个州的人口数 (Population)、文盲率 (Illiteracy)、收入水平 (Income)、结霜天数 (Frost)、犯罪率 (Murder)</p> <p>针对 states 数据框, 做 Murder 对 Population、Illiteracy、Income、Frost 的回归的置换检验, perm 参数为 Prob, 结果保存在 fit 中</p>
---	--

### 12.3.3 单因素方差分析和协方差分析

<pre>library(lmPerm) library(multcomp) fit &lt;- aovp(response~trt, data=cholesterol, perm="Prob")</pre>	<p>已知:</p> <p>multcomp 包的数据集 cholesterol 展示了各种疗法 (trt) 对降低胆固醇的疗效 (response)</p> <p>采用对比做单因素方差分析的置换检验, perm 参数为 Prob, 结果保存在 fit 中</p>
--	--

<pre>library(lmPerm) fit &lt;- aovp(weight ~ gesttime + dose, data=litter, perm="Prob")</pre>	<p>已知:</p> <p>数据集 litter 提供了研究控制妊娠期时间 (gesttime) 相同时, 观测四种药物剂量 (dose) 对鼠仔体重 (weight) 的影响</p> <p>对此做单因素协方差分析的置换检验, 结果保存在 fit 中</p>
---	---

### 12.3.4 双因素方差分析

<pre>library(lmPerm) fit &lt;- aovp(len~supp*dose, data=ToothGrowth, perm="Prob")</pre>	<p>已知:</p> <p>ToothGrowth 数据集中研究了两种喂食方法 (supp) 和三种剂量 (dose) 对豚鼠牙齿生长 (len) 的影响</p>
---	---



	对此做双因素方差分析的置换检验，perm 参数为 Prob
--	-------------------------------

## 12.4 置换检验点评

非正态数据（如分布偏倚很大）、存在离群点、样本很小或无法做参数检验	置换检验真正发挥功用的地方是处理_____、_____、_____或_____等情况。
初始样本对感兴趣的总体情况代表性很差	如果_____，即使是置换检验也无法提高推断效果
获取置信区间 估计测量精度	置换方法对于_____和_____是比较困难的，但这正是自助法大显神通的地方

## 12.5 自助法

从初始样本重复随机替换抽样，生成一个或一系列待检验统计量的经验分布	所谓自助法，即_____
置信区间	无需假设一个特定的理论分布，便可生成统计量的_____，并能检验统计假设

## 12.6 boot 包中的自助法

	一般来说，自助法有三个主要步骤：
能返回待研究统计量值的 一个数值 一个向量	（1）写一个能返回_____函数； 如果只有单个统计量（如中位数），函数应该返回_____； 如果有一列统计量（如一系列回归系数），函数应该返回_____
boot()	（2）为生成 R 中自助法所需的有效统计量重复数，使用_____函数对上面所写的函数进行处理
boot.ci()	（3）使用_____函数获取第（2）步生成的统计量的置信区间

boot() bootobject <- boot(data=, statistic=, R=, ...)	主要的自助法函数是_____，它的格式为_____
--	---------------------------

向量、矩阵、数据框	boot()函数的 data 参数可以是_____
生成 k 个统计量以供自举的函数（k=1 时对单个统计量进行自助抽样） indices	boot()函数的 statistic 参数是为了_____； 函数需包括_____参数，以便 boot()函数用它从每个重复中选择实例
自助抽样的次数	boot()函数的 R 参数表示_____
其他对生成待研究统计量有用的参数，可在函数中传输	boot()函数的... 参数是表示_____

<b>t0</b> (从原始数据得到的 <b>k</b> 个统计量的观测值) <b>t</b> (一个 $R \times k$ 矩阵, 每行即 <b>k</b> 个统计量的自助重复值)	若 <b>boot()</b> 的结果储存在 <b>bootobject</b> 中, 则 <b>bootobject</b> 含有的元素为 _____
<b>bootobject\$t0</b> 和 <b>bootobject\$t</b>	可以用 _____ 和 _____ 来获取上述元素

<b>print()</b> 和 <b>plot()</b> <b>boot.ci()</b>	一旦生成了自助样本, 可通过 _____ 和 _____ 函数来检查结果; 如果结果看起来还算合理, 使用 _____ 函数获取统计量的置信区间
--	---

<b>boot.ci(bootobject, conf=, type=)</b>	<b>boot.ci()</b> 函数的语法格式为 _____
预期的置信区间 (默认: <b>conf=0.95</b> )	<b>boot.ci()</b> 函数的 <b>conf</b> 参数表示 _____
返回的置信区间类型 可能值为 <b>norm</b> 、 <b>basic</b> 、 <b>stud</b> 、 <b>perc</b> 、 <b>bca</b> 和 <b>all</b> (默认: <b>type="all"</b> )	<b>boot.ci()</b> 函数的 <b>type</b> 参数表示 _____ <b>type</b> 的可能值有 _____

### 12.6.1 对单个统计量使用自助法

	针对 <b>mtcars</b> 数据集, 做每加仑行驶的英里数 ( <b>mpg</b> ) 对车重 ( <b>wt</b> ) 和发动机排量 ( <b>disp</b> ) 的回归, 用自助法获得 95% 的 <b>R</b> 平方值的置信区间:
<pre>rsq &lt;- function(formula, data, indices) {   d &lt;- data[indices, ]   fit &lt;- lm(formula, data=d)   return(summary(fit)\$r.square) }</pre>	写一个获得 <b>R</b> 平方值的函数, 命名为 <b>rsq</b> <sup>139</sup>
<pre>results &lt;- boot(data=mtcars, statistic=rsq, R=1000, formula=mpg~wt+disp)</pre>	对 <b>R</b> 平方值做 1000 次抽样, 结果保存在 <b>result</b> 中
<pre>print(results)</pre>	展示自助法抽样结果
<pre>plot(results)</pre>	绘制自助法抽样结果
<pre>boot.ci(results, type=c("perc", "bca"))</pre>	获得 95% 的置信区间

### 12.6.2 多个统计量的自助法

	针对 <b>mtcars</b> 数据集, 做每加仑行驶的英里数 ( <b>mpg</b> ) 对车重 ( <b>wt</b> ) 和发动机排量 ( <b>disp</b> ) 的回归, 用自助法获得三个回归系数 (截距项、车重和发动机排量) 95% 的置信区间:
<pre>bs &lt;- function(formula, data, indices) {</pre>	写一个返回回归系数向量的函数, 命名为 <b>bs</b>

<sup>139</sup> 在这一步中设定的参数 **formula** 和 **data** 的具体内容要在下一步中的 **boot()** 函数中给出。

<pre>d &lt;- data[indices, ] fit &lt;- lm(formula, data=d) return(coef(fit)) }</pre>	
<pre>results &lt;- boot(data=mtcars, statistic=b s, R=1000, formula=mpg~wt+di sp)</pre>	对回归系数做 1000 次自助抽样，结果保存在 results 中
<pre>print(results)</pre>	展示自助抽样结果
<pre>bootobject\$ t</pre> 截距项 车重 发动机排量	绘制多个统计量自助抽样时，添加一个索引参数，指明 plot()和 boot.ci()函数所分析_____的列； 在本例中索引 1 指_____, 索引 2 指_____, 索引 3 指_____
<pre>plot(results, index=2)</pre>	绘制车重的回归系数的自助抽样结果
<pre>boot.ci(results, type="bca", index=2)</pre>	获得车重的回归系数的自助抽样结果
<pre>boot.ci(results, type="bca", index=3)</pre>	获得发动机排量回归系数的 95%置信区间

有些人认为只要样本能够较好地代表总体，初始样本大小为 20~30 即可得到足够好的结果	自助法的初始样本需要多大？
1000 次重复在大部分情况下都可满足要求	自助法中应该重复多少次？

## 第 13 章 广义线性模型

### 13.1 广义线性模型和 glm()函数

非正态因变量的分析	广义线性模型扩展了线性模型的框架，它包含了_____
正态分布 做任何分布的假设	在标准线性模型中，你可以假设 Y 呈_____，但你并没有对预测变量 $X_j$ _____
$g(\mu_Y) = \beta_0 + \sum_{j=1}^p \beta_j X_j$ 条件均值的函数（称为连接函数）	广义线性模型拟合的形式为_____，其中 $g(\mu_Y)$ 是_____（称作_____）
指数分布族中的一种分布即可	你可放松 Y 为正态分布的假设，改为 Y 服从_____
最大似然估计的多次迭代	设定好连接函数和概率分布后，便可以通过_____推导出各参数值

#### 13.1.1 glm()函数

glm() 函数 glm(formula, family=family(link=function), data=)	R 中可通过_____拟合广义线性模型 该函数的基本语法格式为_____
---	---

	下列概率分布（family）的相应默认的连接函数（function）是：
link = "logit"	binomial
link = "identity"	gaussian
link = "inverse"	gamma
link = "1/mu^2"	inverse.gaussian
link = "log"	poisson
link = "identity", variance = "constant"	quasi
link = "logit"	quasibinomial
link = "log"	quasipoisson

	已知： 响应变量为 Y 三个预测变量为 X1、X2、X3； 数据框为 mydata
二值响应变量（0 和 1）	Logistic 回归适用于_____
glm(Y~X1+X2+X3, family=binomial(link="logit"),	拟合 Logistic 回归的命令为_____

data=mydata)	
在给定时间内响应变量为事件发生数目的情形	泊松回归适用于_____
glm(Y~X1+X2+X3, family=poisson(link="log"), data=mydata)	拟合泊松回归的命令为_____
glm(Y~X1+X2+X3, family=gaussian(link="identity"), data=mydata)	标准线性模型也是广义线性模型的一个特例；如果令连接函数 $g(\mu_Y) = \mu_Y$ 或恒等函数，并设定概率分布为正态（高斯）分布，那么标准线性模型等同于_____
极大似然估计	广义线性模型参数估计的推导依据的是_____，而非最小二乘法

### 13.1.2 连用的函数

	与 glm()连用的函数有：
summary()	展示拟合模型的细节
coefficients()、coef()	列出拟合模型的参数（截距项和斜率）
confint()	给出模型参数的置信区间（默认为 95%）
residuals()	列出拟合模型的残差值
anova()	生成两个拟合模型的方差分析表
plot()	生成评价拟合模型的诊断图
predict()	用拟合模型对新数据集进行预测
deviance()	拟合模型的偏差
df.residual()	拟合模型的残差自由度

### 13.1.3 模型拟合和回归诊断

	已知： model 为 glm()函数返回的对象
plot(predict(model, type="response"), residuals(model, type="deviance")) <sup>140</sup>	绘制初始响应变量的预测值与残差的图形
plot(hatvalues(model))	绘制帽子值的诊断图
plot(rstudent(model))	绘制学生化残差值的诊断图
plot(cooks.distance(model))	绘制 Cook 距离统计量的诊断图

<sup>140</sup> 根据 <https://zhidao.baidu.com/question/201007864664151765.html>, predict 中的参数 type="response"用于选择预测之后的输出结果，此参数能用在 binomial 数据，也就是响应变量是二分值的时候，这个参数选成 type="response", 表示输出结果预测响应变量为 1 的概率。另外，residuals 中的参数 type="deviance"表示获取残差的方式为 deviance residuals, 其公式可参照 <https://www.rdocumentation.org/packages/binomTools/versions/1.0-1/topics/Residuals> 和 [https://r-forge.r-project.org/scm/viewvc.php/\\*checkout\\*/pkg/BinomTools/inst/ResidualsGLM.pdf?revision=6&root=binomtools&pathrev=6](https://r-forge.r-project.org/scm/viewvc.php/*checkout*/pkg/BinomTools/inst/ResidualsGLM.pdf?revision=6&root=binomtools&pathrev=6) 中的说明。

<code>library(car)</code> <code>influencePlot(model)</code>	创建一个综合性的诊断图，横轴代表杠杆值，纵轴代表学生化残差，绘制的符号大小与 Cook 距离大小成正比 <sup>141</sup>
--	--

有许多值时 只有有限个值时（比如 <b>Logistic</b> 回归）	当响应变量_____时，诊断图非常有用； 当响应变量_____时（比如_____回归），诊断图的功效就会降低很多
---	---

## 13.2 Logistic 回归

二值型结果	当通过一系列连续型和/或类别型预测变量来预测_____变量时， <b>Logistic</b> 回归是一个非常有用的工具
-------	---

	已知：  AER 包中的数据集 <b>Affairs</b> 包含了 9 个变量：一年来婚外情频率（ <b>affairs</b> ）、性别（ <b>gender</b> ）、年龄（ <b>age</b> ）、婚龄（ <b>yearsmarried</b> ）、是否有小孩（ <b>children</b> ）、宗教信仰程度（ <b>religiousness</b> ）、学历（ <b>education</b> ）、职业（ <b>occupation</b> ）、婚姻的自我评分（ <b>rating</b> ）  做以上变量对婚外情频率的 <b>Logistic</b> 回归：
<code>data(Affairs, package="AER")</code>	加载数据集 <b>Affairs</b>
<code>summary(Affairs)</code>	对 <b>Affairs</b> 中各变量做描述性统计
<code>table(Affairs\$affairs)</code>	作 <b>affairs</b> 变量的频次表
<code>Affairs\$ynaffair[Affairs\$affairs &gt; 0] &lt;- 1</code>  <code>Affairs\$ynaffair[Affairs\$affairs == 0] &lt;- 0</code>  <code>Affairs\$ynaffair &lt;- factor(Affairs\$ynaffair, levels=c(0, 1), labels=c("No", "Yes"))</code>	把 <b>affairs</b> 转变成二值型因子 <b>ynaffair</b> ，有婚外私通的为 1，没有的为 0，并分别标注为 <b>Yes</b> 和 <b>No</b>
<code>table(Affairs\$ynaffair)</code>	作 <b>ynaffair</b> 的频次表
<code>fit.full &lt;- glm(ynaffair ~ gender + age + yearsmarried + children + religiousness + education + occupation + rating, data=Affairs, family=binomial())</code>	将 <b>ynaffair</b> （二值因子）作为结果变量做 <b>Logistic</b> 回归，结果保存在 <b>fit.full</b> 中

<sup>141</sup> 以上四幅图像没有在原书中展示，<http://blog.csdn.net/skyonefly/article/details/52040261> 为我们提供了这些图的示例。

<code>summary(fit.full)</code>	展示 fit.full 的结果 <sup>142</sup>
<code>fit.reduced &lt;- glm(ynaffair ~ age + yearsmarried + religiousness + rating, data=Affairs, family=binomial())</code>	在上述回归中 p 值显著的变量为 age、yearsmarried、religiousness、rating，以这些变量为自变量再做一次 Logistic 回归，结果保存在 fit.reduced 中
<code>summary(fit.reduced)</code>	展示 fit.reduced 的回归结果
<code>anova()</code> 卡方检验	新模型的每个回归系数都非常显著： $p < 0.05$ 。由于两个模型嵌套（fit.reduced 是 fit.full 的子集），使用_____对他们进行比较；对于广义线性回归，可以用_____
<code>anova(fit.reduced, fit.full, test="Chi sq")</code>	使用上述函数比较两次回归
表明四个预测变量的新模型与九个完整预测变量的模型拟合程度一样好	结果卡方值不显著，表明_____

### 13.2.1 解释模型参数

<code>coef(fit.reduced)</code>	查看 fit.reduced 得到的回归系数
$Y=1$ 的对数优势比 (log)  一单位预测变量的变化可引起的响应变量对数优势比的变化  指数化	在 Logistic 回归中，因变量是_____； 回归系数的含义是当其他预测变量不变时，_____； 由于对数优势比解释性差，可对回归系数进行_____ <sup>143</sup>
<code>exp(coef(fit.reduced))</code>	对 fit.reduced 的结果进行指数化

<code>confint()</code>	可使用_____函数获取系数的置信区间
<code>exp(confint(fit.reduced))</code>	可使用_____在优势比尺度上得到系数 95%的置信区间
$\exp(\beta_j)^n$	对于二值型 Logistic 回归，某预测变量 n 单位的变化引起的较高值上优势比的变化为_____，它反映的信息可能更为重要

### 13.2.2 评价预测变量对结果概率的影响

<code>testdata\$prob &lt;- predict(fit.reduced, newdata=testdata, type="response")<sup>144</sup></code>	已知：  <code>testdata &lt;- data.frame(rating=c(1, 2, 3, 4, 5),</code>
---	--

<sup>142</sup> gender 只有男、女两个值，如果同时纳入回归会造成完全共线性，R 在这里自动取了 gender 为 male。同理，children 选取了 yes。

<sup>143</sup> 因变量阳性结果的概率 P 与自变量 X 的关系通常不是直线关系，而是呈现曲线关系；而自变量 X 与 P 和 (1-P) 比值的对数呈线性关系，因此 Logistic 回归模型 P 与 X 线性函数表达式为： $\ln(p/(1-p)) = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$ 。P 为事件发生的概率，1-P 为事件不发生的概率。其中优势比 (odds) 就是指的  $p/(1-p)$ 。当我们对此式进行指数化后， $\ln(p/(1-p))$  转化为  $p/(1-p)$ ，更便于对各变量的系数进行解释。结合本例，指数化前的回归系数是 1.931 (Intercept)，-0.035 (age)，0.101 (yearsmarried)，-0.329 (religiousness)，-0.461 (rating)。由于 age, religiousness, rating 的系数符号都为负，表明他们与婚外情的对数优势比是呈现负相关关系的，只有 yearsmarried 与婚外情的对数优势比呈现正相关关系。对应的，指数化回归系数后，我们看到只有 yearsmarried 的系数是大于 1 的，age, religiousness, rating 的系数都小于 1。因此，在保持其他变量不变的前提下，只有 yearsmarried 可以使得婚外情取 1 的概率变大。

<sup>144</sup> 对 type="response" 的注释见 13.1.3 节。

	<pre>age=mean(Affairs\$age), yearsmarried=mean(Affairs\$yearsmarried), religiousness=mean(Affairs\$religiousness))</pre> <p>使用 <code>fit.reduced</code> 构建的模型和系数预测不同水平下 <code>rating</code> 对应婚外情概率的值（保持其他变量不便），结果保存到 <code>testdata\$prob</code> 中</p>
--	---

<pre>testdata\$prob &lt;- predict(fit.reduced, newdata=testdata, type="response")</pre>	<p>已知：</p> <pre>testdata &lt;- data.frame(rating=mean(Affairs\$rating), age=seq(17, 57, 10), yearsmarried=mean(Affairs\$yearsmarried), religiousness=mean(Affairs\$religiousness))</pre> <p>使用 <code>fit.reduced</code> 构建的模型和系数预测不同年龄（<code>age</code>）对应的婚外情概率的值（保持其他变量不变），结果保存到 <code>testdata\$prob</code> 中</p>
---	--

### 13.2.3 过度离势

类二项分布	当出现过度离势时，仍可使用 <code>glm()</code> 函数拟合 Logistic 回归，但此时需要将二项分布改为_____
残差偏差： 残差自由度： 比 1 大很多	<p>检测过度离势的一种方法是比较二项分布模型的_____与_____<sup>145</sup>；</p> <p>如果_____便可认为存在过度离势</p>

"bi nomial " "quasi bi nomial "	对过度离势进行卡方检验，第一次拟合使用 <code>family=_____</code> ，其结果记为 <code>fit</code> ；第二次拟合使用 <code>family=_____</code> ，返回对象记为 <code>fit.od</code>
<code>pchi sq(summary(fit.od)\$dispersion * fit\$df.residual, fit\$df.residual, lower = F)</code>	卡方检验为_____
不存在过度离势	若得到的 <code>p</code> 值不显著，可认为_____

### 13.2.4 扩展

<code>robust</code> 包中的 <code>glmRob()</code> 函数	当 Logistic 回归模型数据出现离群点和强影响点时，_____包的_____函数可以做稳健 Logistic 回归
--	--

<sup>145</sup> 残差偏差和残差自由度可以在 `summary(fit.reduced)` 的 Residual deviance:615.36 on 596 degrees of freedom 找到。



<code>ml logit</code> 包中的 <code>ml logit()</code> 函数	若响应变量包含两个以上的无序类别（比如，已婚/寡居/离婚），便可使用_____包的_____函数可以做多元 Logistic 回归
<code>rms</code> 包中的 <code>lrm()</code> 函数	若响应变量是一组有序类别（比如，信用风险为差/良/好），便可使用_____包的_____函数拟合序数 Logistic 回归。

## 13.3 泊松回归

通过一系列连续型和/或类别型预测变量来预测计数型结果变量	当_____时，泊松回归是一个非常有用的工具。
------------------------------	-------------------------

	<p>已知：</p> <p><code>robust</code> 包中的 <code>data=breslow.dat</code> 包含了癫痫治疗相关的数据。我们取相应变量为 <code>sumY</code>（治疗开始后八周内癫痫发病数）、预测变量为治疗方案（<code>Trt</code>（<code>progabide</code> 或 <code>placebo</code>））、年龄（<code>Age</code>）、基础癫痫发病数（<code>Base</code>）。已知这几个变量在 <code>data=breslow.dat</code> 中分别是第 10、第 8、第 7、第 6 个变量</p> <p>拟合 <code>sumY</code> 对 <code>Base</code>、<code>Age</code>、<code>Trt</code> 的泊松回归：</p>
<code>data(breslow.dat, package="robust")</code>	加载 <code>robust</code> 包中的数据集 <code>data=breslow.dat</code>
<code>names(breslow.dat)</code>	展示 <code>data=breslow.dat</code> 的各变量名
<code>summary(breslow.dat[c(6, 7, 8, 10)])</code>	总结各变量
<pre>opar &lt;- par(no.readonly=TRUE) par(mfrow=c(1, 2)) attach(breslow.dat) hist(sumY, breaks=20) boxplot(sumY ~ Trt) par(opar)</pre>	<p>绘制 <code>sumY</code> 的直方图和箱线图</p> <ol style="list-style-type: none"> <li>1) 生成一个可以修改的当前图形的参数列表</li> <li>2) 创建 1 行 2 列的图形矩阵</li> <li>3) 绑定 <code>breslow.dat</code> 数据集</li> <li>4) 绘制 <code>sumY</code> 的直方图，共切分成 20 个组</li> <li>5) 绘制不同 <code>Trt</code> 对应的 <code>sumY</code> 的箱线图</li> <li>6) 还原图形设置</li> </ol>
方差异质性	与标准最小二乘回归不同，泊松回归并不关注_____
<code>fit &lt;- glm(sumY ~ Base + Age + Trt, data=breslow.dat, family=poisson())</code>	拟合 <code>sumY</code> 对 <code>Base</code> 、 <code>Age</code> 、 <code>Trt</code> 的泊松回归，保存在 <code>fit</code> 中
<code>summary(fit)</code>	查看回归结果

### 13.3.1 解释模型参数

<code>coef()</code>	使用_____函数可获取模型系数
条件均值的对数形式 $\log_e(\lambda)$	在泊松回归中，因变量以_____来建模
癫痫发病数的对数均值	若年龄的回归参数为 0.0227，表明保持其他预测变量不变，年龄增加一岁，_____将相应增加 0.0227 <sup>146</sup>

<code>exp(coef(fit))</code>	通常在因变量的初始尺度（癫痫发病数，而非发病数的对数）上解释回归系数比较容易； 为此用_____命令来指数化系数
期望的癫痫发病数	经指数化系数后，Age 的系数为 1.023，表明保持其他变量不变，年龄增加一岁，_____将乘以 1.023
从安慰剂组到治疗组 乘以 0.858 14.2% <sup>147</sup>	Trtprogabide 为 0.858，表明一单位 Trt 的变化（即_____），期望的癫痫发病数将_____，也就是说，保持基础癫痫发病数和年龄不变，服药组相对于安慰剂组癫痫发病数降低了_____
成倍 线性	与 Logistic 回归中的指数化参数相似，泊松模型中的指数化参数对响应变量的影响都是_____增加的，而不是_____相加

### 13.3.2 过度离势

响应变量观测的方差比依据泊松分布预测的方差大	当_____时，泊松回归可能发生过度离势
标准误和置信区间 显著性检验	如果存在过度离势，那么可能会得到很小的_____和_____，并且_____也过于宽松
残差偏差与残差自由度的比例远远大于 1	与 Logistic 回归类似，如果_____，那么表明存在过度离势
<code>deviance(fit)/df.residual(fit)</code>	对于 sumY 对 Base、Age、Trt 的泊松回归的结果 fit，可以用命令_____来检验

<code>library(qcc)</code> <code>qcc.overdispersion.test(breslow.dat\$sumY, type="poisson")</code>	使用 qcc 包检验泊松模型过度离势的函数检验 data=breslow.dat 数据集中 sumY 的过度离势
--	--

<code>fit.od &lt;- glm(sumY ~ Base + Age + Trt, data=breslow.dat, family=quasipoisson())</code>	使用类泊松方法拟合 sumY 对 Base、Age、Trt 的回归，结果保存在 fit.od 中
---	--

### 13.3.3 扩展

<sup>146</sup> 原文此处为 0.03，应该是一处笔误。

<sup>147</sup> 原文此处为 20%，应该是一处笔误。

<p>一个比率 (<math>\frac{\lambda}{time}</math>)</p> <p>记录每个观测的时间长度的变量 (<code>time</code>)</p>	<p>在允许时间段变化的泊松回归模型中，结果变量为_____；</p> <p>分析比率，必须包含一个_____</p>
<p><code>fit &lt;- glm(sumY ~ Base + Age + Trt, data=breslow.dat, offset=log(time), family=poisson)</code></p>	<p>纳入时间变量 <code>time</code>，使用泊松回归拟合 <code>sumY</code> 对 <code>Base</code>、<code>Age</code>、<code>Trt</code> 的回归，结果保存在 <code>fit</code> 中</p>
<p>零膨胀的泊松回归</p>	<p>在一个数据集中，0 计数的数目比用泊松模型预测的多时可用_____</p>
<p><code>pscl</code> 包中的 <code>zeroinfl()</code> 函数</p>	<p>_____包的_____函数可以做零膨胀的泊松回归</p>
<p><code>robust</code> 包中的 <code>glmRob()</code> 函数</p> <p>稳健的</p>	<p>当存在离群点和强影响点时，可用_____包的_____函数拟合_____的广义线性模型</p>

## 第 14 章 主成分分析和因子分析

一组很少的不相关变量 主成分	主成分分析（PCA）是一种数据降维技巧，它可将大量相关变量转化为_____，这些无关变量称为_____
线性组合 最大化各主成分所解释的方差	主成分是观测变量的_____，形成线性组合的权重是通过_____来获得

一组变量的潜在结构	因子分析（EFA）是一系列用来发现_____的方法
线性组合 结构基础或“原因”	主成分（PC1 和 PC2）是观测变量（X1 到 X5）的_____； 因子（F1 和 F2）被当作观测变量的_____，而不是它们的线性组合

### 14.1 R 中的主成分和因子分析

	psych 包中有用的因子分析函数：
principal ()	含多种可选的方差旋转方法的主成分分析
fa()	可用主轴、最小残差、加权最小平方或最大似然法估计的因子分析
fa.parallel ()	含平行分析的碎石图
factor.plot ()	绘制因子分析或主成分分析的结果
fa.diagram()	绘制因子分析或主成分的载荷矩阵
scree()	因子分析和主成分分析的碎石图

### 14.2 主成分分析

#### 14.2.1 判断主成分的个数

psych 包的 fa.parallel () 函数	利用_____包的_____函数，可以同时三种特征值判别准则进行评价
----------------------------	------------------------------------

library(psych) fa.parallel (USJudgeRatings[, - 1], fa="pc", n.iter=100, show.legend=FALSE)	利用上述函数绘制 USJudgeRatings 数据集（去掉第一个变量）基于观测特征值的碎石检验，根据 100 个随机数据矩阵推导出来的特征值均值，以及大于 1 的特征值准则 <sup>148</sup>
---	--

#### 14.2.2 提取主成分

<sup>148</sup> 解读图 14-2：（1）相对于  $y=1$  的实心直线上方的点应该保留。（2）表示随机矩阵推导的特征值均值的虚线上方的点应该保留。（3）观察由直线和 x 符号组成的折线，在图形曲折程度变化最大之处之上的点应该保留。

<b>principal ()</b> 函数 <b>principal (r, nfactors=, rotate=, scores=)</b>	_____函数可以根据原始数据矩阵或者相关系数矩阵做主成分分析，其格式为_____
<b>r</b> 是相关系数矩阵或原始数据矩阵	解释 principal()函数的 r 参数
<b>nfactors</b> 设定主成分数（默认为 1）	解释 principal()函数的 nfactors 参数
<b>rotate</b> 指定旋转的方法（默认最大方差旋转（ <b>varimax</b> ），见 14.2.3 节） <sup>149</sup>	解释 principal()函数的 rotate 参数
<b>scores</b> 设定是否需要计算主成分得分（默认不需要）	解释 principal()函数的 scores 参数

<b>pc &lt;- principal (USJudgeRatings[, -1], nfactors=1)</b>	对 USJudgeRatings 数据集（去掉第一个变量）提取 1 个主成分，结果保存在 pc 中
--	---

<b>pc</b>	展示 pc; 解释 pc 中的以下指标:
<b>PC1</b> 栏包含了成分载荷，指观测变量与主成分的相关系数。如果提取不止一个主成分，那么还将会有一栏 <b>PC2</b> 、 <b>PC3</b> 等栏	<b>PC1</b> 包含了_____
<b>h2</b> 栏指成分公因子方差，即主成分对每个变量的方差解释度	<b>h2</b> 栏指_____
<b>u2</b> 栏指成分唯一性，即方差无法被主成分解释的比例（ $1 - h2$ ）	<b>u2</b> 栏指_____
<b>SS loadings</b> 行包含了与主成分相关联的特征值，指的是与特定主成分相关联的标准化后的方差值	<b>SS loadings</b> 行包含了_____
<b>Proportion Var</b> 行表示的是每个主成分对整个数据集的解释程度	<b>Proportion Var</b> 行表示_____

	已知:  Harman23.cor 数据集的 cov 是 8 个身体测量指标（变量）的相关系数  用较少的变量替换这些变量:
<b>library(psych)</b> <b>fa.parallel (Harman23.cor\$cov, n.obs=302, fa="pc", n.iter=100, show.legend=FALSE)</b>	判断要提取的主成分数量
<b>pc &lt;- principal (Harman23.cor\$cov, nfactors=2, rotate="none")</b>	提取 2 个主成分，不进行主成分旋转，结果保存在 pc 中

<sup>149</sup> 这个数由 14.2.1 节的碎石检验而定。

	解释 pc 中的以下指标:
第二个主成分与各变量间的相关系数	PC2
PC1 和 PC2 分别解释的整个数据集方差的百分比	Proportion Var
PC1 和 PC2 总共解释的整个数据集方差的百分比	PC2 对应的 Cumulative Var

### 14.2.3 主成分旋转

去噪	旋转尽可能地对成分_____
正交旋转 斜交旋转	旋转方法有两种: 使选择的成分保持不相关 (_____), 和让它们变得相关 (_____)
方差极大旋转 即载荷阵每列只有少数几个很大的载荷, 其他都是很小的载荷	最流行的正交旋转是_____, 它试图对载荷阵的列进行去噪, 使得每个成分只由一组有限的变量来解释 (即_____)

<code>rc &lt;- principal (Harman23.cor\$cov, nfactors=2, rotate="varimax")</code>	对 Harman23.cor\$cov 提取 2 个主成分, 并做方差极大旋转, 结果保存在 rc 中
RC	在 rc 展示的结果中, 列名称都从 PC 变成了_____, 以表示成分被旋转

### 14.2.4 获取主成分得分

<code>principal ()</code> 函数	利用_____函数, 可以获得每个调查对象在该主成分上的得分
<code>score=TRUE</code> <code>scores</code>	当数据是原始数据时, 可以使用参数_____直接获得主成分得分; 当使用该参数时, 主成分得分存储在 <code>principal()</code> 函数返回对象的_____元素中

<code>pc &lt;- principal (USJudgeRatings[, -1], nfactors=1, score=TRUE)</code>	对 USJudgeRatings[, -1] 提取 1 个主成分并获得主成分得分, 结果保存在 pc 中
--	--

主成分得分 主成分得分的系数	当主成分分析基于相关系数矩阵时, 原始数据便不可用了, 也不可能获取每个观测的_____, 但是你可以得到用来计算_____
-------------------	--

<code>rc &lt;- principal (Harman23.cor\$cov, nfactors=2, rotate="varimax")</code>	对 Harman23.cor\$cov 提取 2 个主成分, 做方差极大旋转, 结果保存在 rc 中
---	--

round(unclass(rc\$weights), 2) <sup>150</sup>	获得主成分得分系数		
$PC1 = 0.28*height + 0.30*arm.span + 0.30*forearm + 0.29*lower.leg - 0.06*weight - 0.08*bitro.diameter - 0.10*chest.girth - 0.04*chest.width$ $PC2 = -0.05*height - 0.08*arm.span - 0.09*forearm - 0.06*lower.leg + 0.33*weight + 0.32*bitro.diameter + 0.34*chest.girth + 0.27*chest.width$	已知：		
		RC1	RC1
	height	0.28	-0.05
	arm.span	0.30	-0.08
	forearm	0.30	-0.09
	lower.leg	0.28	-0.06
	weight	-0.06	0.33
	bitro.diameter	-0.08	0.32
	chest.girth	-0.10	0.34
	chest.width	-0.04	0.27
主成分得分公式为_____			

## 14.3 探索性因子分析

无法观测	因子分析的目标是通过发掘隐藏在数据下的一组较少的、更为基本的_____的变量，来解释一组可观测变量的相关性
------	---

	<p>已知:</p> <p>ability.cov\$cov 提供了 6 个变量的协方差矩阵</p> <p>将其转化为相关系数矩阵:</p>
<code>options(digits=2)</code>	设置默认 2 位小数
<code>covariances &lt;- ability.cov\$cov</code>	令 covariances 为 ability.cov\$cov
<code>correlations &lt;- cov2cor(covariances)</code>	将 covariances 转化为相关系数矩阵 correlations

### 14.3.1 判断需提取的公共因子数

<code>fa.parallel(correlations, n.obs=112, fa="both", n.iter=100)</code>	可以用_____命令绘图判断需对 correlations 提取的因子数(同时展示主成分和公共因子分析的结果)
大于 0，而不是 1	对于 EFA，Kaiser-Harris 准则的特征值数大于_____，而不是_____

### 14.3.2 提取公共因子

<p>fa() 函数</p> <p><code>fa(r, nfactors=, n.obs=, rotate=, scores=, fm=)</code></p>	可以使用_____提取公因子，其格式为_____
	解释 fa() 中的以下参数

<sup>150</sup> unclass 将 rc\$weights 从文本类型变成数字类别 (unclass() 可以去掉对象的类别)，round() 函数进行了保留 2 位小数的处理。

<b>r</b> 是相关系数矩阵或者原始数据矩阵	<b>r</b>
<b>nfactors</b> 设定提取的因子数（默认为 1）	<b>nfactors</b>
<b>n.obs</b> 是观测数（输入相关系数矩阵时需要填写）	<b>n.obs</b>
<b>rotate</b> 设定旋转的方法（默认互变异数最小法）	<b>rotate</b>
<b>scores</b> 设定是否计算因子得分（默认不计算）	<b>scores</b>
<b>fm</b> 设定因子化方法（默认极小残差法）	<b>fm</b>

包括最大似然法（ <b>ml</b> ） 主轴迭代法（ <b>pa</b> ） 加权最小二乘法（ <b>wls</b> ） 广义加权最小二乘法（ <b>gls</b> ） 最小残差法（ <b>minres</b> ）	与 PCA 不同，提取公共因子的方法很多，包括最大似然法（_____）、 主轴迭代法（_____）、加权最小二乘法（_____）、广义加权最 小二乘法（_____）和最小残差法（_____）
主轴迭代法	有时候最大似然法不会收敛，此时使用_____效果会很好

<b>fa &lt;- fa(correlations, nfactors=2, rotate="none", fm="pa")</b>	使用主轴迭代因子法提取 <b>correlations</b> 的 2 个因子，不做旋转，结 果保存在 <b>fa</b> 中
--	--

### 14.3.3 因子旋转

<b>fa.varimax &lt;- fa(correlations, nfactors=2, rotate="varimax", fm="pa")</b>	采用主轴迭代因子法提取 <b>correlations</b> 的 2 个因子，做正交旋转， 结果保存在 <b>fa.varimax</b> 中
存在一个语言智力因子和非语言智力因子	将与 <b>reading</b> 和 <b>vocab</b> 相关系数较大的因子 <b>PA1</b> 称为语言因子，将 与 <b>picture</b> 、 <b>blocks</b> 、 <b>maze</b> 相关系数较大的因子 <b>PA2</b> 称为非语言因 子， <b>general</b> （普通智力测量）这一变量在两个因子载荷上较为平 均，这表明_____

两个因子不相关	使用正交旋转将人为地强制_____
斜交转轴法 <b>promax</b>	使用_____，如_____，可允许两个因子相关，且模型更符合 真实数据

<b>fa.promax &lt;- fa(correlations, nfactors=2, rotate="promax", fm="pa")</b>	采用主轴迭代因子法提取 <b>correlations</b> 的 2 个因子，做斜交旋转， 结果保存在 <b>fa.promax</b> 中
---	--

变量与因子的相关系数 因子结构矩阵、因子模式矩阵和因子关联矩阵	对于正交旋转，因子分析的重点在于因子结构矩阵（_____）， 而对于斜交旋转，因子分析会考虑三个矩阵：_____、_____、
------------------------------------	--



--	--

标准化的回归系数 <sup>151</sup>	在 <code>fa.promax</code> 中, <code>PA1</code> 和 <code>PA2</code> 是因子模式矩阵, 他们是_____而不是相关系数
-------------------------	--

$F = P * \Phi$ ( $F$ 是因子载荷阵, $P$ 为因子模式矩阵, $\Phi$ 为因子关联矩阵)	因子结构矩阵(因子载荷阵)没有被列出来, 但可以用公式_____获得(该公式见第一版第 311 页, 第二版第 310 页的 <code>fsm</code> 函数)
---	---

<code>factor.plot()</code> 或 <code>fa.diagram()</code>	使用_____或_____函数可以获得绘制正交或者斜交结果的图形
<code>factor.plot(fa.promax, labels=rownames(fa.promax\$loadings))</code>	使用 <code>factor.plot()</code> 绘制 <code>fa.promax</code> 的图像, 标签使用 <code>fa.promax\$loadings</code> 的行名称
<code>fa.diagram(fa.promax, simple=FALSE)</code>	使用 <code>fa.diagram()</code> 绘制 <code>fa.promax</code> 的图像
最大的载荷 因子间的相关系数 图形有多个因子时	在 <code>fa.diagram()</code> 中若使用 <code>simple=TRUE</code> , 那么仅显示每个因子下_____, 以及_____; 这类图形在_____时十分实用

#### 14.3.4 因子得分

<code>fa.promax=fa(correlations, nfactors=2, rotate="promax", fm="pa", scores=TRUE)</code>  <code>fa.promax\$weights</code>	获取 <code>correlations</code> 的因子得分, 采用主轴迭代因子法, 提取 2 个因子, 做斜交旋转, 结果保存在 <code>fa.promax</code> 中
---	--

#### 14.3.5 其他与 EFA 相关的包

### 14.4 其他潜变量模型

<sup>151</sup> Standardized Regression Coefficients, 假定回归方程为  $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + u$ ,  $\hat{\beta}_1$  和  $\hat{\beta}_2$  通过 OLS 获得, 则标准化的回归系数为  $\hat{\beta}_1 = \hat{\beta}_1 \cdot [sd(x_1)/sd(y)]$ 。它消除了变量所取单位的影响。

## 第 15 章 处理缺失数据的高级方法

### 15.1 处理缺失值的步骤

### 15.2 识别缺失值

NA NaN	R 使用_____（不可得）代表缺失值，_____（不是一个数）
Inf -Inf	符号_____和_____分别代表正无穷和负无穷

is.na() is.nan() is.infinite()	函数_____用来识别缺失值； 函数_____用来识别不可能值； 函数_____用来识别无穷值； 每个返回结果都是 TRUE 或者 FALSE
is.na()	当 $x \leftarrow NA$ 时，以上三个函数返回 TRUE 的是_____
is.na() 和 is.nan()	当 $x \leftarrow 0/0$ 时，以上三个函数返回 TRUE 的是_____
is.infinite()	当 $x \leftarrow 1/0$ 时，以上三个函数返回 TRUE 的是_____

complete.cases()	_____函数可用来识别矩阵或数据框中没有缺失值的行。若每行都包含完整的实例，则（每行）返回 TRUE 的逻辑向量；若某行有一个或多个缺失值，则该行返回 FALSE <sup>152</sup>
------------------	---

	已知： data(sleep, package="VIM")
sleep[complete.cases(sleep), ]	列出没有缺失值的行
sleep[!complete.cases(sleep), ]	列出有一个或多个缺失值的行

1 和 0 sum() 和 mean()	由于逻辑值 TRUE 和 FALSE 分别等于_____和_____，可用_____和_____函数来获取关于缺失数据的有用信息
sum(is.na(sleep\$Dream))	获取 sleep 数据集的 Dream 变量有几个缺失值
mean(is.na(sleep\$Dream))	获取 sleep 数据集的 Dream 变量的缺失值占比
mean(!complete.cases(sleep))	获取 sleep 数据集中有缺失值的观测的占比

<sup>152</sup> 此处为了更清晰的表述代码含义，修改了原文的表述。

NA 和 NaN 有效值	complete.cases()函数仅将_____和_____识别为缺失值, 无穷值 (Inf 和 -Inf) 被当作_____
-----------------	--

## 15.3 探索缺失值模式

### 15.3.1 列表显示缺失值

mi ce 包中的 md. pattern() 函数	_____包的_____函数可以生成 一个以矩阵或者数据框形式展示缺失模式的表格																																																																																																														
library(mi ce) data(sleep, package="VIM") md. pattern(sleep)	对 sleep 数据集做上述分析																																																																																																														
表中的 1 和 0 显示了缺失值模式：0 表示变量的列中有缺失值，1 则表示没有缺失值。第一行表述了“无缺失值”的模式（所有元素都为 1）。第二行表述了“除了 Span 之外无缺失值”的模式。第一列表示各缺失值模式的实例个数，最后一列表示各模式中有缺失值的变量的个数。此处可以看到，有 42 个实例没有缺失值，仅 2 个实例缺失了 Span。9 个实例同时缺失了 NonD 和 Dream 的值。数据集包含了总共 (42×0)+(2×1)+…+(1×3)=38 个缺失值。最后一行给出了每个变量中缺失值的数目。	若所得结果为： <table><tr><th></th><th>BodyWgt</th><th>BrainWgt</th><th>Pred</th><th>Exp</th><th>Danger</th><th>Sleep</th><th>Span</th><th>Gest</th><th>Dream</th><th>NonD</th></tr><tr><td>42</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>2</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>3</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr><tr><td>9</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>2</td></tr><tr><td>2</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>2</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>2</td></tr><tr><td>2</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>3</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>3</td></tr><tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>4</td><td>4</td><td>4</td><td>12</td><td>38</td></tr></table> 如何对上表进行解读？		BodyWgt	BrainWgt	Pred	Exp	Danger	Sleep	Span	Gest	Dream	NonD	42	1	1	1	1	1	1	1	1	1	0	2	1	1	1	1	1	1	0	1	1	1	3	1	1	1	1	1	1	1	0	1	1	9	1	1	1	1	1	1	1	1	0	2	2	1	1	1	1	1	0	1	1	1	2	1	1	1	1	1	1	1	0	0	1	2	2	1	1	1	1	1	0	1	1	0	3	1	1	1	1	1	1	1	0	1	0	3		0	0	0	0	0	4	4	4	12	38
	BodyWgt	BrainWgt	Pred	Exp	Danger	Sleep	Span	Gest	Dream	NonD																																																																																																					
42	1	1	1	1	1	1	1	1	1	0																																																																																																					
2	1	1	1	1	1	1	0	1	1	1																																																																																																					
3	1	1	1	1	1	1	1	0	1	1																																																																																																					
9	1	1	1	1	1	1	1	1	0	2																																																																																																					
2	1	1	1	1	1	0	1	1	1	2																																																																																																					
1	1	1	1	1	1	1	0	0	1	2																																																																																																					
2	1	1	1	1	1	0	1	1	0	3																																																																																																					
1	1	1	1	1	1	1	0	1	0	3																																																																																																					
	0	0	0	0	0	4	4	4	12	38																																																																																																					

### 15.3.2 图形探究缺失数据

aggr()	aggr()函数不仅绘制每个变量的缺失值数, 还绘制每个变量组合的缺失值数
library("VIM") aggr(sleep, prop=FALSE, numbers=TRUE)	使用上述函数分析 VIM 包的 sleep 数据集的数据缺失情况, 显示各缺失情况的计数统计, 保留数值标签
aggr(sleep, prop=TRUE, numbers=TRUE)	使用上述函数分析 VIM 包的 sleep 数据集的数据缺失情况, 显示各缺失情况的比例统计, 保留数值标签

matrixplot(sleep)	_____命令可生成 sleep 数据集每个实例数据的图形, 浅色表示值小, 深色表示值大, 红色表示缺失值 <sup>153</sup>
margi nplot() 函数	_____函数可生成一幅散点图, 在图形边界展示两个变量的缺失值信息

### 15.3.3 用相关性探索缺失值

<sup>153</sup> 原文提到该图可以进行交互, 在 RStudio 中这一功能失灵, 但是可以在 R 的基本软件中实现。想在 RStudio 中实现按 BodyWgt 排序可以用 matrixplot(sleep, sortby="BrainWgt")。另外, 原文的“无缺失值的睡眠变量 (Dream、NonD 和 Sleep)”有歧义, 应该是“睡眠变量 (Dream、NonD 和 Sleep) 中没有缺失的数值”。

	分析变量“缺失”与其他变量间的关系：
<code>x &lt;- as.data.frame(abs(is.na(sleep)))</code>	用指示变量替代数据集中的数据（1 表示缺失，0 表示存在）生成的数据框保存在 x 中
<code>y &lt;- x[which(apply(x, 2, sum) &gt; 0)]</code>	用标准差大于 0 为依据提取含缺失值但不全是缺失值 <sup>154</sup>
<code>cor(y)</code>	列出有缺失值的变量之间的相关系数
<code>cor(sleep, y, use="pairwise.complete.obs")</code>	展示含缺失值变量与其他可观测变量间的关系，采用成对删除

## 15.4 理解缺失数据的来由和影响

## 15.5 理性处理不完整数据

## 15.6 完整实例分析（行删除）

<code>newdata &lt;- mydata[complete.cases(mydata), ]</code> 或 <code>newdata &lt;- na.omit(mydata)</code>	已知 mydata 是含有缺失值的数据框或者矩阵形式的实例（行），可以用_____命令或者_____命令将所有包含缺失值的行都删除，然后将结果存储到 newdata 中
--	--

<code>options(digits=1)</code>  <code>cor(na.omit(sleep))</code> 或 <code>cor(sleep, use="complete.obs")</code>	使用行删除法删除 sleep 中含有缺失值的观测，并计算新数据集各变量间的相关系数，保留 1 位有效数字（使用两种代码得到相同的结果）
--	---

<code>fit &lt;- lm(Dream ~ Span + Gest, data=na.omit(sleep))</code>	使用行删除法删除 sleep 中含有缺失值的观测，针对新数据集做 Dream 对 Span 和 Gest 的回归，结果保存在 fit 中
---	--

## 15.7 多重插补

然后返回一个包含多个（默认为 5 个）完整数据集的  插补  函数可依次对每个完整数据集应用统计模型（如线性模型或广义线性模型）  整合为一组结果	使用 mice() 函数做多重插补。首先从一个包含缺失数据的数据框开始，然后返回_____；  每个完整数据集都是通过对原始数据框中的缺失数据进行_____而生成的；  然后，with() 函数可依次_____；  最后，pool() 函数将这些单独的分析结果_____
---	---

	基于 mice 包的分析过程可分为：
--	--------------------

<sup>154</sup> 全是缺失值或者全不是缺失值的情况下，标准差都为 0。

<code>library(mice)</code>	载入 mice 包
<code>imp &lt;- mice(mydata, m)</code>	令 <code>mydata</code> 为包含缺失值的矩阵或数据框, 获得包含 <code>m</code> 个插补数据集的列表对象, 命名为 <code>imp</code>
<code>fit &lt;- with(imp, analysis)</code>	对 <code>imp</code> 做应用统计分析 <code>analysis</code> , 结果保存在 <code>fit</code> 中
<code>pooled &lt;- pool(fit)</code>	获取 <code>m</code> 个单独统计分析平均结果的列表对象, 命名为 <code>pooled</code>
<code>summary(pooled)</code>	获取分析结果

	将多重插补应用到 <code>sleep</code> 数据集上:
<code>library(mice)</code>	载入 mice 包
<code>data(sleep, package="VIM")</code>	绑定 VIM 包的 <code>sleep</code> 数据集
<code>imp &lt;- mice(sleep, seed=1234)</code>	完成插补 (随机数种子为 1 2 3 4), 保存在 <code>imp</code> 中
<code>fit &lt;- with(imp, lm(Dream ~ Span + Gest))</code>	针对插补后的数据, 做 <code>Dream</code> 对 <code>Span</code> 和 <code>Gest</code> 的回归, 结果保存在 <code>fit</code> 中
<code>pooled &lt;- pool(fit)</code>	整合单独的统计分析结果到 <code>pooled</code> 中
<code>summary(pooled)</code>	展示分析结果
<code>imp</code>	查看 <code>imp</code> 对象的汇总信息
<code>imp\$imp\$Dream</code>	查看 <code>Dream</code> 变量上有缺失值的观测的实际插补值
<code>complete(imp, action=#)</code>	使用_____函数可以观察 <code>m</code> 个插补数据集中的任意一个
<code>dataset3 &lt;- complete(imp, action=3)</code>	查看 <code>imp</code> 中插补后的第三个数据集, 命名为 <code>dataset3</code>

## 15.8 处理缺失值的其他方法

### 15.8.1 成对删除

<code>cor(sleep, use="pairwise.complete.obs")</code>	获得 <code>sleep</code> 数据集变量间的相关系数, 并使用成对删除法
成对删除法 简单 (非随机插补)	作者不建议使用_____法和_____法

### 15.8.2 简单 (非随机) 插补

## 第 16 章 时间序列<sup>155</sup>

	来自 <b>stats</b> 包的时序分析函数：
<b>ts()</b>	生成时序对象
<b>start()</b>	返回时间序列的开始时间
<b>end()</b>	返回时间序列的结束时间
<b>frequency()</b>	返回时间序列中时间点的个数
<b>window()</b>	对时序对象取子集
<b>stl()</b>	用 <b>LOESS</b> 光滑将时序分解为季节项、趋势项和随机项
<b>monthplot()</b>	画出时序中的季节项
<b>HoltWinters()</b>	拟合指数平滑模型
<b>lag()</b>	返回取过指定滞后项后的时序
<b>arima()</b>	拟合 <b>ARIMA</b> 模型
<b>Box.test()</b>	进行 <b>Ljung-Box</b> 检验以判断模型的残差是否独立

	来自 <b>forecast</b> 包的时序分析函数：
<b>ma()</b>	拟合一个简单的移动平均模型
<b>seasonplot()</b>	生成季节图
<b>forecast()</b>	预测时序的未来值
<b>accuracy()</b>	返回时序的拟合优度量
<b>ets()</b>	拟合指数平滑模型，同时也可以自动选取最优模型
<b>Acf()</b>	估计自相关函数
<b>Pacf()</b>	估计偏自相关函数
<b>ndiffs()</b>	找到最优差分次数以移除序列中的趋势项
<b>auto.arima()</b>	自动选择 <b>ARIMA</b> 模型

	来自 <b>graphics</b> 包的时序分析函数：
<b>plot()</b>	画出时间序列的折线图

	来自 <b>base</b> 包的时序分析函数：
--	--------------------------

<sup>155</sup> 本章来自第二版的第 15 章。是第二版新增的内容。

<code>diff()</code>	返回取过滞后项和（或）差分后的序列
	来自 <code>tseries</code> 包的时序分析函数：
<code>adf.test()</code>	对序列做 ADF 检验以判断其是否平稳
<code>bds.test()</code>	进行 BDS 检验以判断序列中的随机变量是否服从独立同分布

## 16.1 在 R 中生成时序对象

观测值、起始时间、终止时间以及周期（如月、季度或年）的结构	在 R 中分析时间序列的前提是我们将分析对象转成 <u>时间序列对象</u> ，即 R 中一种包括____、____、____以及____的结构
-------------------------------	--

<code>ts()</code> 函数	一个数值型向量或数据框中的一列可通过____存储为时序对象
<code>myseries &lt;- ts(data, start=, end=, frequency=)</code>	上述函数的语法格式为____
所生成的时序对象	其中， <code>myseries</code> 是____
原始的包含观测值的数值型向量	<code>data</code> 是____
时序的起始时间和终止时间	<code>start</code> 参数和 <code>end</code> 参数（可选）给出____
每个单位时间所包含的观测值数量	<code>frequency</code> 为____
年度数据	<code>frequency=1</code> 对应____
月度数据	<code>frequency=12</code> 对应____
季度数据	<code>frequency=4</code> 对应____

	已知：  <code>sales &lt;- c(18, 33, 41, 7, 34, 35, 24, 25, 24, 21, 25, 20, 22, 31, 40, 29, 25, 21, 22, 54, 31, 25, 26, 35)</code>
<code>tsales &lt;- ts(sales, start=c(2003, 1), frequency=12)</code>	生成时序对象 <code>tsales</code> ，数据为 <code>sales</code> ，起始时间为 2003 年 1 月，月度数据
<code>plot(tsales)</code>	生成 <code>tsales</code> 的折线图
<code>start(tsales)</code>	获取 <code>tsales</code> 的开始时间
<code>end(tsales)</code>	获取 <code>tsales</code> 的结束时间
<code>frequency(tsales)</code>	获取 <code>tsales</code> 单位时间中观测的数量
<code>tsales.subset &lt;- window(tsales, start=c(2003, 5), end=c(2004, 6))</code>	获取 <code>tsales</code> 从 2003 年 5 月到 2004 年 6 月的子集，保存在 <code>tsales.subset</code> 中

## 16.2 时序的平滑化和季节性分解

### 16.2.1 通过简单移动平均进行平滑处理

每个数据点都可用这一点和其前后两个点的平均值来表示，这就是居中移动平均	什么是居中移动平均？ 它的数学表达式是什么？ $k=2q+1$ 是什么？
$S_t = (Y_{t-q} + \dots + Y_t + \dots + Y_{t+q}) / (2q + 1)$ <p><math>k=2q+1</math> 是每次用来平均的观测值的个数，一般我们会将其设为一个奇数</p>	
最后的 $(k-1)/2$ 个观测值	居中移动平均法的代价是每个时序集会损失_____

<code>ma()</code>	<code>forecast</code> 包的_____函数可以用来做时序数据的平滑处理
-------------------	---

	绘制 Nile 数据的简单移动平均：
<code>library(forecast)</code>	载入 <code>forecast</code> 包
<code>opar &lt;- par(no.readonly=TRUE)</code>	保存图形参数
<code>par(mfrow=c(2, 2))</code>	设定图形组合为 $2 \times 2$ 的矩阵
<code>ylim &lt;- c(min(Nile), max(Nile))</code>	令 <code>ylim</code> 为 Nile 最小值和最大值组成的向量
<code>plot(Nile)</code>	绘制 Nile 的折线图
<code>plot(ma(Nile, 3), ylim=ylim)</code>	绘制 $k$ 为 3 的 Nile 的移动平均图， $y$ 轴范围为 <code>ylim</code>
<code>plot(ma(Nile, 7), ylim=ylim)</code>	绘制 $k$ 为 7 的 Nile 的移动平均图， $y$ 轴范围为 <code>ylim</code>
<code>plot(ma(Nile, 15), ylim=ylim)</code>	绘制 $k$ 为 15 的 Nile 的移动平均图， $y$ 轴范围为 <code>ylim</code>
<code>par(opar)</code>	复原图形初始设置

### 16.2.2 季节性分解

趋势因子、季节性因子和随机因子	存在季节性因素的时间序列数据（如月度数据、季度数据等）可以被分解为_____、_____和_____
长期变化	趋势因子能捕捉到_____
一年内的周期性变化	季节性因子能捕捉到_____
那些不能被趋势或季节效应解释的变化	随机（误差）因子能捕捉到_____

相加模型 相乘模型	可以通过_____模型或_____模型来分解数据
相乘模型	如果波动是与趋势成正比的，即整体增长时波动越大；这种基于现有水平的放大（或者缩减）决定了_____模型更适合这类情况



LOESS 光滑 stl()	将时序分解为趋势项、季节项和随机项的常用方法是用_____做季节性分解。这可以通过 R 中_____函数实现
stl(ts, s.window=, t.window=)	上述函数的语法格式是_____
将要分解的时序	其中, ts 是_____
季节效应变化的速度	参数 s.window 控制_____
趋势项变化的速度 更快的变化速度	t.window 控制_____, 较小的值意味着_____
"periodic"	令 s.window=_____可使得季节效应在各年间都一样
相加模型 对数变化	stl()函数只能处理_____; 若遇到适用于相乘模型的序列, 可通过_____转换为相加模型

相乘模型	观察 AirPassengers 序列, 发现序列的波动随着整体水平的增长而增长, 即_____更适合这个序列
取对数	在用 str()季节性分解该序列之前, 需对原始序列_____, 使得序列方差稳定下来, 从而可以对其拟合一个可加性季节模型

	用 str()对 AirPassengers 序列做季节性分解:
plot(AirPassengers)	绘制 AirPassengers 原始序列的折线图
lAirPassengers <- log(AirPassengers)	对 AirPassengers 做对数变换, 保存在 lAirPassengers 中
plot(lAirPassengers)	绘制 lAirPassengers 的折线图
fit<-stl(lAirPassengers, s.window="period")	对 lAirPassengers 做季节性分解, 季节效应限定为每年都一样, 结果保存在 fit 中
plot(fit)	绘制 fit
fit\$time.series	获得每个观测值经对数变换后各分解项的值
exp(fit\$time.series)	获得每个观测值各分解项的值 <sup>156</sup>

monthplot(AirPassengers)	绘制 AirPassengers 每个月份组成的子序列 (连接所有年份 1 月的点, 连接所有年份 2 月的点, 以此类推)
library(forecast) seasonplot(AirPassengers)	绘制 lAirPassengers 以年份为子序列的季节图

<sup>156</sup> 因为设定了 s.window="period", 如果把 fit\$time.series 和 exp(fit\$time.series)展开, 就能发现每年各月的 seasonal 值是一样的。所以原文说“观察季节效应可发现, 7 月的乘客数增长了 24% (即乘子为 1.24), 而 11 月的乘客数减少了 20% (即乘子为 0.8)”, 我们只用观察 1949 年分解的数据就行。

## 16.3 指数预测模型

指数模型	_____是用来预测时序未来值的最常用模型
常数水平项和时间点 $i$ 处随机项的时间序列	单指数模型拟合的是只有_____和_____, 这时认为时间序列不存在趋势项和季节效应
水平项和趋势项	双指数模型 (也叫 Holt 指数平滑) 拟合的是有_____和_____ 的时序
水平项、趋势项以及季节效应	三指数模型 (也叫 Holt-Winters 指数平滑) 拟合的是有_____, _____以及_____的时序

forecast 包的 ets() 函数	_____包的_____函数可以拟合指数模型
ets(ts, model="ZZZ")	上述函数的语法格式为_____
第一个字母代表误差项 第二个字母代表趋势项 第三个字母则代表季节项	限定模型的字母"ZZZ"有三个; 第一个字母代表_____; 第二个字母代表_____; 第三个字母则代表_____
相加模型 (A) 相乘模型 (M) 无 (N) 自动选择 (Z)	可选的字母包括: 相加模型 (_____) 相乘模型 (_____) 无 (_____) 自动选择 (_____)
ets(ts, model="ANN") ses(ts)	用于拟合单指数模型的函数为_____或_____
ets(ts, model="AAN") holt(ts)	用于拟合双指数模型的函数为_____或_____
ets(ts, model="AAA") hw(ts)	用于拟合三指数模型的函数为_____或_____

### 16.3.1 单指数平滑

加权平均; 使得距离现在越远的观测值对平均数的影响越小	单指数平滑根据现有的时序值的_____对未来值做短期预测, 其中权数选择的宗旨是_____
$Y_t = \text{level} + \text{irregular}_t$	单指数平滑模型假定时序中的观测值可被表示为_____
$Y_{t+1} = c_0 Y_t + c_1 Y_{t-1} + c_2 Y_{t-2} + c_3 Y_{t-3} + \dots^{157}$ $c_i = \alpha(1 - \alpha)^i, i = 0, 1, 2, 3, \dots, 0 \leq \alpha \leq 1$	在时间点 $Y_{t+1}$ 的预测值 (一步向前预测) 可写作_____;

<sup>157</sup> 原文的公式存在错误。

	其中 $c_i$ 等于_____；
权重下降的速度； 近期观测值的权重越大； 历史观测值的权重越大	上式中， $\alpha$ 参数控制_____； $\alpha$ 越接近 1，则_____； $\alpha$ 越接近 0，则_____；

	针对 <code>nhtemp</code> 时序拟合单指数平滑：
<code>library(forecast)</code>	载入包
<code>fit &lt;- ets(nhtemp, model="ANN")</code>	拟合单指数平滑，结果保存在 <code>fit</code> 中
<code>forecast(fit, 1)</code>	预测时序未来 1 步的值
<code>plot(forecast(fit, 1))</code>	绘制 <code>nhtemp</code> 的折线图，以及单指数模型所得到的一步向前预测
<code>accuracy(fit)</code>	展示预测中最主流的几个准确性度量

不大	在预测准确性度量的几个指标中，平均误差（ME）和平均百分比误差（MPE）用处_____
比较不同的预测的准确性； 测量尺度中存在一个真实为零的点	平均绝对百分误差（MAPE）没有单位，因此可用来_____，但它同时也假定_____

### 16.3.2 Holt 指数平滑和 Holt-Winters 指数平滑

水平项和趋势项（斜率）	Holt 指数平滑可以对有_____和_____的时序进行拟合
$Y_t = \text{level} + \text{slope} * t + \text{irregular}_t$	时刻 $t$ 的观测值可表示为_____
水平项的指数型 斜率的指数型 $[0, 1]$ 越近的观测值的权重越大	平滑参数 $\alpha$ 控制_____， $\beta$ 控制_____； 两个参数的取值范围都是_____； 参数取值越大意味着_____

水平项、趋势项以及季节项 $Y_t = \text{level} + \text{slope} * t + s_t + \text{irregular}_t$	Holt-Winters 指数光滑可用来拟合有_____、_____、以及_____的时间序列； 此时模型可表示为_____
季节项的指数下降 $[0, 1]$ 越近的观测值的季节效应权重越大	其中， $s_t$ 代表_____； 除 $\alpha$ 和 $\beta$ 参数外， $\gamma$ 光滑参数控制_____； $\gamma$ 参数的取值范围是_____； $\gamma$ 值越大，意味着_____

	用 Holt-Winters 指数光滑来预测 AirPassengers 时序中接下来的五个值:
<code>library(forecast)</code>	载入包
<code>fit &lt;- ets(log(AirPassengers), model="AAA")</code>	将 AirPassengers 对数化, 使其满足可加模型, 用 Holt-Winters 拟合对数化后的序列, 结果保存在 fit 中
<code>fit</code>	展示拟合结果
<code>accuracy(fit)</code>	展示拟合准确性
<code>pred &lt;- forecast(fit, 5)</code>	预测 AirPassengers 时序接下来的五个值, 结果保存在 pred 中
<code>pred</code>	展示 pred
<code>plot(pred)</code>	绘制经对数化后的 AirPassengers 以及接下来五个预测值的折线图
<code>pred\$mean &lt;- exp(pred\$mean)</code>	将预测值指 <sup>158</sup> 数化, 保存在同名变量中
<code>pred\$lower &lt;- exp(pred\$lower)</code>	将 80%和 95%置信区间的下界值 <sup>159</sup> 指数化, 保存在同名变量中
<code>pred\$upper &lt;- exp(pred\$upper)</code>	将 80%和 95%置信区间的上界值 <sup>160</sup> 指数化, 保存在同名变量中
<code>p &lt;- cbind(pred\$mean, pred\$lower, pred\$upper)</code>	按列合并指数化后的变量, 保存在 p 中
<code>dimnames(p)[[2]] &lt;- c("mean", "Lo 80", "Lo 95", "Hi 80", "Hi 95")<sup>161</sup></code>	将 p 的列依次重命名为“mean”、“Lo80”、“Lo95”、“Hi80”、“Hi95”
<code>p</code>	展示重命名后的 p

### 16.3.3 ets()函数和自动预测

<code>ets(AirPassengers, model="MAM")</code>  <code>hw(AirPassengers, seasonal="multiplicative")</code>	假定趋势项可加, 季节项和误差项可乘, 对 AirPassengers 时间序列进行拟合 (采用两种方式)
---	---

<code>library(forecast)</code> <code>fit &lt;- ets(JohnsonJohnson)</code> <code>fit</code>	对 JohnsonJohnson 时间序列进行自动指数预测
<code>plot(forecast(fit), flty=2)</code>	绘制 JohnsonJohnson 的折线图以及未来八个季度 (默认) 的预测值, 其中预测值由虚线表示

## 16.4 ARIMA 预测模型

### 16.4.1 概念介绍

<sup>158</sup> 指 pred 中的 Point Forecast 列。

<sup>159</sup> 指 pred 中的 Lo 80 和 Lo 95 列。

<sup>160</sup> 指 pred 中的 Hi 80 和 Hi 95 列。

<sup>161</sup> dimnames(p)是包含了 p 的行名称和列名称的列表, 其中[[2]]是列名。

<code>lag(ts, k)</code>	时间序列可以通过_____函数变成 k 阶滞后
-------------------------	-------------------------

k 时期之前的观测值 ( $Y_{t-k}$ )	自相关度量时间序列中各个观测值之间的相关性 $AC_k$ 即一系列观测值 ( $Y_t$ ) 和_____之间的相关性
相关性 ( $AC_1, AC_2, \dots, AC_k$ )	_____构成的图即自相关函数图 (ACF 图)
<code>forecast</code> 包中的 <code>Acf()</code> 函数	_____包的_____函数能够绘制 ACF 图

	偏自相关即当序列 $Y_t$ 和 $Y_{t-k}$ 之间的所有值 ( $Y_{t-1}, Y_{t-2}, Y_{t-3}, \dots, Y_{t-k+1}$ ) 带来的效应都被移除后, 两个序列的相关性 <sup>162</sup>
<code>forecast</code> 包中的 <code>Pacf()</code> 函数	_____包的_____函数能够绘制 PACF 图

平稳性 (或可以被转换为平稳序列)	ARIMA 模型主要用于拟合具有_____的时间序列
时序图	平稳性一般可以通过_____直观判断
对数变换 Box-Cox 变换	如果方差不是常数, 则需要对数据做变换, 可用的方法包括_____和_____
差分	如果数据中存在趋势项, 则需要对其进行_____
<code>diff()</code> <code>diff(ts, differences=d)</code> 对 <code>ts</code> 的差分次数	可以通过_____函数对序列进行差分, 其语法格式为_____, 其中 <code>d</code> 表示_____
<code>forecast</code> 包的 <code>ndiffs()</code> 函数	_____包的_____函数能够帮我们找到最优的 <code>d</code> 值
<code>tseries</code> 包的 <code>adf.test()</code> 函数 序列满足平稳性	可以通过 ADF 统计检验量来验证平稳性假设; _____包的_____函数可以用来做 ADF 检验; 如果结果显著, 则认为_____

#### 16.4.2 ARMA 和 ARIMA 模型

它之前 <code>p</code> 个值的线性组合 $AR(p): Y_t = \mu + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_1 Y_{t-3} + \dots + \beta_p Y_{t-p} + \varepsilon_t$	在一个 <code>p</code> 阶自回归模型中, 序列中的每一个值都可以用_____来表示; 其表达式为_____
之前的 <code>q</code> 个残差的线性组合 $MA(q): Y_t = \mu - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \theta_1 \varepsilon_{t-3} - \dots - \theta_p \varepsilon_{t-p} - \varepsilon_t$	在一个 <code>q</code> 阶移动平均模型中, 时序中的每个值都可以用_____来表示; 其表达式为_____

<sup>162</sup> 这里的描述比较抽象, 更具体的描述为: 对以下 `k` 阶自回归方程进行 OLS 估计:  $Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_1 Y_{t-3} + \dots + \beta_k Y_{t-k} + \varepsilon_t$ , 则  $\hat{\beta}_k$  就是 `k` 阶样本偏自相关系数 `p` (详见陈强《高级计量经济学及 stata 应用》第 20.1 节)。

$Y_t = \mu + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \beta_3 Y_{t-3} + \dots + \beta_p Y_{t-p} + \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \theta_1 \varepsilon_{t-3} - \dots - \theta_p \varepsilon_{t-p} + \varepsilon_t$	将 $p$ 阶自回归和 $q$ 阶移动平均混合起来的方法即ARMA( $p, q$ ), 其表达式为_____
---	---

差分 观测值 残差	ARIMA( $p, d, q$ )模型意味着时序被_____了 $d$ 次, 且序列中的每个观测值都是用过去的 $p$ 个_____和 $q$ 个_____的线性组合表示的
-----------------	---

(1) 确保时序是平稳的 (2) 找到一个(或几个)合理的模型(即选定可能的 $p$ 值和 $q$ 值) (3) 拟合模型 (4) 从统计假设和预测准确性等角度评估模型 (5) 预测	建立 ARIMA 模型的五个步骤包括_____
---	-------------------------

	(1) 确保时序是平稳的:
<code>library(forecast)</code> <code>library(tseries)</code>	载入 forecast 和 tseries 包
<code>plot(Nile)</code>	绘制 Nile 的时序图(以便观察是否有某种趋势, 及方差是否稳定)
<code>ndiffs(Nile)</code>	找到最优差分次数以便移除序列中的趋势项
<code>dNile &lt;- diff(Nile)</code>	对 Nile 做一阶差分, 结果保存在 dNile 中
<code>plot(dNile)</code>	绘制 dNile 的时序图
<code>adf.test(dNile)</code>	对 dNile 做 ADF 检验

	(2) 找到一个(或几个)合理的模型(即选定可能的 $p$ 值和 $q$ 值):
<code>Acf(dNile)</code>	绘制 dNile 的 ACF 图
<code>Pacf(dNile)</code>	绘制 dNile 的 PACF 图
$q$	通过 ACF 确定 ARIMA 模型中的_____值
$p$	通过 PACF 确定 ARIMA 模型中的_____值 <sup>163</sup>

	(3) 拟合模型
<code>arima(ts, order=c(p, d, q))</code> <sup>164</sup>	可以用_____函数拟合一个 ARIMA 模型, 其表达式为_____

<sup>163</sup> “表 15-6 选择 ARIMA 模型的方法”中“逐渐减小到零”指的就是“拖尾”, 在读图观察时注重 2 点: (1) 随着滞后阶数增大, 相关值趋近于 0; (2) 趋近 0 的过程是缓慢、平缓的。“在  $p$  (或  $q$ ) 阶后减小到零”指的是“截尾”, 在读图观察时注重: (1) 存在这样一阶滞后, 这一阶滞后本身的相关值是比较大的, 而这一阶滞后后的那阶滞后的相关值呈现“断崖式”下降, 而且在此之后的相关值保持在一个较小的区间里。(以下为个人观点:) 在观察时注重整体趋势, 注重头几阶滞后的相关值变化, 对于个别异常值可以无视。如果有按一定周期出现的较大相关值, 要检查季节性。

<sup>164</sup> 原文此处 `order=c(q,d,q)` 有误。

<pre>library(forecast) fit &lt;- arima(Nile, order=c(0, 1, 1)) fit accuracy(fit)</pre>	<p>用 Nile 序列拟合 ARIMA(0,1,1)模型：</p> <p>载入包；</p> <p>拟合 ARIMA(0,1,1)结果保存在 fit 中；</p> <p>展示结果；</p> <p>展示拟合准确性</p>
<p>AIC 值</p> <p>AIC 值越小越好</p>	<p>如果还有其他备选模型，则可以通过比较_____来得到最合理的模型，比较的准则是_____</p>

	(4) 从统计假设和预测准确性等角度评估模型
<p>均值为 0 的正态</p> <p>零</p> <p>独立正态</p>	<p>一般来说，一个模型如果合适，那模型的残差应该满足_____分布，并且对于任意的滞后阶数，残差自相关系数都应该为_____；</p> <p>换句话说，模型的残差应该满足_____分布</p>
<pre>qqnorm(fit\$residuals) qqline(fit\$residuals)</pre> <p>落在图中的直线上</p>	<p>作 QQ 图分析 fit 残差的正态性；</p> <p>如果满足正态分布，则数据中的点会_____</p>
<pre>Box.test()</pre> <p>残差的自相关系数为零</p>	<p>_____函数可以检验残差的自相关系数是否都为零；</p> <p>如果模型没有通过显著性检验，即可认为_____</p>
<pre>Box.test(fit\$residuals, type="Ljung-Box")</pre>	<p>用上述函数检验 fit 残差的独立性</p>

	(5) 预测
<pre>forecast(fit, 3)</pre>	<p>对 Nile 做接下来 3 年的预测</p>
<pre>plot(forecast(fit, 3))</pre>	<p>绘制 Nile 的时间序列和未来 3 年的预测</p>

### 15.4.3 ARIMA 的自动预测

<p>forecast 包的 ets() 函数</p>	<p>_____包中的_____函数实现最优指数模型的自动选取</p>
<p>forecast 包的 auto.arima() 函数</p>	<p>_____包中的_____函数可以实现最优 ARIMA 模型的自动选取</p>

	对 sunspots 时间序列做 ARIMA 自动预测：
<pre>library(forecast)</pre>	<p>载入包</p>
<pre>fit &lt;- auto.arima(sunspots)</pre>	<p>对 sunspots 时序做 ARIMA 自动预测，结果保存在 fit 中</p>
<pre>fit</pre>	<p>展示自动预测模型和方法</p>
<pre>forecast(fit, 3)</pre>	<p>实现未来 3 期的预测</p>

`accuracy(fit)`

展示拟合的准确性

## 16.5 延伸阅读